Fast Facial Feature Tracking with Multi-Cue Particle Filter

Liyue Zhao, Jianhua Tao
National Laboratory of Pattern Recognition (NLPR),
Institute of Automation, Chinese Academy of Sciences
Email: {lyzhao, jhtao}@nlpr.ia.ac.cn

Abstract
The paper represents an effective and robust facial feature tracking approach based on the multi-cue particle filter. Both color and edge distributions are integrated into the filter to ensure the tracking accuracy. Sub-region color model is used to rapidly depict the spatial layout of each facial feature. Furthermore, the paper uses edge orientation histogram as a complementary feature to enhance the robustness of tracking results. In order to enhance the robustness of our work, we propose a point distribution model to constraint face configuration and avoid tracking fails during occlusion. An efficient updating algorithm is introduced to avoid tracking error accumulating problems. Experiments show that the method based on the multi-cue particle filter and the updating algorithm gives us an inspiring tracking result. Compared with other facial feature tracking approaches, the method has the good performance in long-time facial feature tracking with temporary occlusions.

Keywords: facial feature tracking, particle filter, template update

1 Introduction
Real-time facial feature tracking is a challenge work and has many applications in the field of human-computer interaction. Most applications, such as facial expression analysis, face animation and facial expression coding, require robust and accurate tracking results as an important data to be analysis. Until now, facial feature tracking still suffers from many problems such as the lost or the drifting of tracking targets during temporary occlusion and large facial motion. Many previous works of facial features tracking paid attention on detecting and tracking facial features based on deformable templates [17] and the AAM [3]. Such methods could obtain high performance of feature contour tracking, but the iterative nature of deformable templates and the AAM make it difficult to be implemented in real-time tracking. Bourel et al [2] apply an enhanced KLT tracker to track facial feature points. This approach solves the lost and drifting problems of KLT tracker during tracking facial feature points, but it will be inefficiency when it suffers from temporary occlusion.

Recently, some one has tried to use particle filters, which has been widely used in object tracking, for the facial feature tracking. For instance, Nummiaro et al [11] and Perez et al [14] present color-based particle filters to track human face. Compared with traditional model-based tracking methods, particle filter method achieves high tracking robust in some familiar tracking problems, such as temporary occlusion and large facial movements. Moreover the proper computational complexity of this approach conduces to its application in real-time tracking. Now many groups devoted to track facial features with particle filter. Hamlaoui et al [7] and Fleck et al [18] present an AAM-based condensation approach, which is proved to be very effective to improve the accuracy of facial feature tracking. On the other hand, Valstar and Pantic [13] propose a particle filtering with color-based observation model to track facial feature points.

As particle filters have shown a good performance in the facial tracking, it is also used in our work. To improve the efficiency and the robust of such model, in the paper, we propose a multi-cue based observation model in the particle filter. The observation model consists of a mixture model that incorporates information from both color and edge of the facial features. It enables us to search and track facial feature points quickly, despite the fact that the features often suffer from occlusions and large movements. To get more detailed tracking results, sub-region color models are used to represent facial features. However, it is difficult to track facial features with only color-based model. Edge features are treated as another important cue, which is capable of depicting facial feature texture. For the purpose of real-time tracking, we adopt edge orientation histogram (EOH) as observation model. Many previous works indicate that the EOH not only could be used in detecting and tracking the interest points on rigid objects [9], but also do perform well on non-rigid objects, such as human body [16] and hand [6].

Although color and edge features based particle filter could deal usual tracking situation, the most common
problems of facial feature tracking are temporary occlusion and tracking error accumulation. In order to solve such problems, we propose the point distribution model (PDM) to constraint tracking results and avoid tracking fails during occlusion. The most familiar PDMs are ASM [4] and AAM [3]. With such model the points drifting problem come to well solution during occlusion. In particular, we use a model update algorithm [10] to estimate the model update feasibility of observation models and ensure tracking results in a long time period. Final experiments show that the method based on the multi-cue particle filter and the updating algorithm gives us an inspiring tracking result which can be easily used for emotion recognition, or facial animation researches.

The remainder of the paper is organized as follows. In Section 2, we explain the particle filter algorithm and introduce our novel integration of multi-cue observation models which contain edge and color information. Section 3 describes the model update algorithm. In Section 4 we discuss the experiment results and draw conclusions in Section 5.

2 Multi-cue particle filter model

The particle filter is a Bayesian sequential importance sampling technique, which recursively approximates the posterior distribution using a finite set of weighted samples. It consists of essentially two steps: prediction and update. Given all available observations \( z_{t-1} = \{ z_1, \ldots, z_{t-1} \} \), the prediction stage uses the probabilistic system transition model \( p(x_t | x_{t-1}) \) to predict the posterior at time \( t \) as:

\[
p(x_t | z_{t-1}) = \int p(x_t | x_{t-1}) p(x_{t-1} | z_{t-1}) \, dx_{t-1}
\]

At time \( t \), the observation \( z_t \) is available, the state can be updated using Bayes’ rule

\[
p(x_t | z_{t}) = \frac{p(z_t | x_t) p(x_t | z_{t-1})}{p(z_t | z_{t-1})}
\]

where \( p(z_t | x_t) \) is described by observation models. Detailed description can be found in [8]. Observation models are crucial for measuring the likelihood of samples in object tracking and are the major difference among different particle filters. For instance, Nummiaro et al [11] and Perez et al [14] presented a color-based observation model in their particle filters for the object tracking by comparing color histograms among object samples. It is robust and efficient to track massive targets with the same color in most cases.

In our work, the observation model \( p(z_t | x_t) \) is defined as a combined local texture likelihood distribution for the facial features in each input image. And the point distribution model is adopt as a prior probabilistic distribution of the face configuration to obtain \( p(x_t | z_{t-1}) \).

2.1 Point distribution model (PDM)

Face configuration is an important cue which could be used to boost the particle filter. [15] performs a ASM model based particle filter to automatically localize facial feature points. The landmarks that defines 2D shape can be represented by a vector \( s = (s_1, s_2, \ldots, s_n) \) of length \( 2n \), where \( n \) is the number of landmarks. Given a set of manually labelled face images, the labeled face shapes can be aligned to the same scale and orientation. PCA is then applied to the aligned shape vectors. By taking the first \( k \) principal components, (which in our case \( k = 4 \)), face shape can be modelled as \( s = s_k + \sum_{i=1}^{n} \lambda_i s_i \), where \( s_k \) is the mean shape of the face, \( \lambda_i = (\lambda_1, \lambda_2, \ldots, \lambda_k) \) is the shape parameter vector, and \( s_i \) are the primary \( k \) vectors of shape. For a detailed introduction, the reader could refer to the paper of Cootes [4].

And the global 2D pose parameters are represented as \( p = (x, y, \alpha, \theta)^T \), where \( \alpha \) is scaling factor, \( \theta \) is the angle of rotation, and \( x, y \) is the position of the face in the image.

2.2 Sub-region color model

As we know, facial features normally have similar color distributions with adjacent regions. It’s hard for us to simply use color histograms to measure the similarity of detailed facial features. In order to employing spatial information which makes the features different to the background, we define a sub-region color model which is similar to the pyramid approach and could better depict the region texture. Furthermore, parameters of the model can also be computed efficiently.

Our approach tracks the square region around one facial feature point instead of tracks one point itself. The sub-region color model separated one square region into \( n \) sub-square regions \( R_{1}, \ldots, R_{n} \). Each region \( R_{i} \) could be depicted by a mean color vector \((r_i, g_i, b_i)\), where \( r_i, g_i, b_i \) represent the average color value of red, green and blue in the \( i \)-th pixel region. Color vector of the pixels within each sub-region \( R_{i} \) could be represented as \((r_i, g_i, b_i) = \sum_{\text{pixels}} (r, g, b) / |R_i| \).

The notation \(|R_i|\) measures the total number of pixels in \( R_i \). The mean color vector of the first frame can be initialized as \((r_1', g_1', b_1')\) in the each region \( R_{i} \).
In our work, $k' = (r_i', g_i', h_i')$ in the first frame is used as the reference color model which might be updated (see Section 3), and $k_i = (r_i, g_i, h_i)$ is the candidate color model in frame $t$. We use Euclidean distance to measure the similarity of $k'$ and $k_i$.

$$
\rho(k', k_i) = \left[ \sum_{i=1}^{n} (r_i' - r_i)^2 + (g_i' - g_i)^2 + (h_i' - h_i)^2 \right]^{1/2}
$$

The likelihood distribution is then given by

$$
p_{\text{color}}(z_i | x_i) \propto e^{-d^2(p|z_i)/2\sigma^2}
$$

### 2.3 Edge orientation histogram

As color features in the search window is easy to be confused among adjacent regions, edge features became an alternative to measure the similarity of samples. But traditionally, edge features are only used in gesture recognition [5] and other aspects of object detection and recognition [6]. Yang [17] employs similar edge orientation histogram (EOH) to evaluate observation likelihood of particle filtering. The advantage of using EOH to depict the search windows of facial feature point is that it could better express the edge or corner properties than color histogram.

In our work, for each frame $t$, we calculate the image gradients $g_x$, $g_y$ along the $x$ and $y$ directions. Here, we denote the position of one pixel by $x = (x,y)$, and the bin index in frame $t$ by $b_t(x) = \{1, \ldots, n\}$. The orientation value $R(x)$ and gradient value $G(x)$ could be represent as $R(x) = \arctan \left( \frac{g_y(x)}{g_x(x)} \right)$ and $G(x) = \sqrt{g_x^2(x) + g_y^2(x)}$. The orientation value is defined between $0^\circ$ to $360^\circ$. We integrate the orientation value into an edge orientation histogram with $N$ bins. The edge distribution $p_t = \{p_t^{(k)}\}_{k=1}^K$ at location $y$ is calculated as:

$$
p_t^{(k)} = K \sum_{i=1}^{I} \delta_b(i, b_t(x)) \delta[\delta(b_t(x) - u)]
$$

where $I$ is the number of pixels in the region, $\delta$ is the Kronecker delta function [11], $K$ is the normalization constant to ensure $\sum_{k=1}^{K} p_t^{(k)} = 1$. $\delta(z)$ represents the weight of the pixel value at $x(x,y)$ and in our work we set $\delta(z) = 1$.

In the first frame, the reference edge orientation histogram is defined as $p^\ast$. The similarity of two histograms is measured by the Bhattacharyya distance [11],

$$
d(p^\ast, p_t) = \left[ 1 - \sum_{k=1}^{K} \sqrt{p_t^{(k)} p_t^{(k)}} \right]^{1/2}
$$

the likelihood distribution is finally defined as:

$$
p_{\text{edge}}(z_i | x_i) \propto e^{-d^2(p|z_i)/2\sigma^2}
$$

### 2.4 Multi-cue particle filtering

The $i$th sample $s_i^{(0)}$ could be represented as $s_i^{(0)} = (x_i^{(0)}, \pi_i^{(0)})$ where $x_i^{(0)}$ is the state vector that has defined in Section 2.1 as $x_i = (p, \lambda)$. As defined in Section 2, the state of the sample $x_i$ is separated into two parts: the feature 2D pose $p_i = (x,y,\alpha,\theta)$ and the shape parameter vector $\lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)$.

By using the dynamic model, the sample set is propagated as:

$$
x_i = A x_{i-1} + w_{i-1}
$$

where $A$ defines the deterministic component of the model $w_{i-1}$ and is a multivariate Gaussian random variable.

To combine both the sub-region color model and edge orientation histogram, we use hybrid weights to integrate them and to update the priori distribution calculated by the particle filter. For each sample $i$, the sample weight could be represented as:

$$
\pi_i^{(0)} = p_i^{(0)}(x_i | s_{i-1}) = \alpha p_{\text{edge}}(z_i | x_i) + (1-\alpha) p_{\text{color}}(z_i | x_i)
$$

The hybrid weight $\alpha$ can be set dynamically without affecting the convergence of the particle filter. When $\alpha = 0$, the algorithm degenerates to a color-based particle filter. By increasing $\alpha$ we place more importance on the edge orientation distributions. We can employ different value of $\alpha$ when we intend to track different facial features which have respective properties.

Function (9) designates that the likelihood distribution model of each facial feature point is independent. Then the weight could be handled separately to make the computing of each particle more effective, we replace the observation model as:

$$
\pi_i^{(0)} = p^{(0)}(x_i | z_{i-1}) = \prod_{t=1}^{T} p^{(0)}(x_i | z_{i-1}^{(t)})
$$

where $p^{(0)}(x_i | z_{i-1}^{(t)})$ denote observation model of the $k$th facial feature point.

The proposal of the weight $\pi$ is to down-weight the occlusion pixels. Assume that $\pi_i^{(0)}$ denotes the weights of the $k$th point in the $i$th sample. When points suffer occlusion during tracking process, we define the point as bad points. The value of weight $\pi_i^{(0)}$ will become very small, and insure that such bad.
points have less significance to the point distribution model. Then the state vector $x_i$ could be determined mainly by the contributions of good points. This characteristic of our method could assure the particle filter tracking robustly during occlusion. Finally, the mean state of the point distribution is estimated at each time step by:

$$E(x) = \sum_{i=1}^{N} \pi_i^{(k)}$$  \hspace{1cm} (11)

3 Model update during the tracking

Most of previous particle filters have no updating procedure [14], although some simple updating strategies [11] have been tried. They assume that the appearance of the tracking object is immutable throughout the entire image sequences. This assumption is not suitable for facial features which are changed irregularly during the period of facial expression. Sometimes, the facial feature tracking is also influenced by illumination conditions or the camera shifting. So the simple update strategy mentioned in [11] still causes large error accumulation during the tracking. Although the tracking error in each frame is tiny, the error accumulation is considerable to make the tracking point “drift” away from the target point after a few frames. In order to alleviate those influences we adopt an efficient algorithm [10] to determine the rationality of observation model update.

The main idea of our model update method is to use the prior model in the first frame to correct the drift of the current model in frame $t$. The whole course of our model update method is shown in Fig.1. We take one facial feature point at location $x$ for example. In the first frame, we denote the initial target model $M_t(x)$ with the square region pixels around position $x$. In frame $t$, the estimate model 2D pose parameter $q_t = [x,y,\theta,a]$ can be acquired by the tracker in Section 2. Then we attempt to match the current model $I_t(W(x,q_t))$ with the first model $M_t(x)$. The new pose parameter $q'_t$ is defined as:

$$q'_t = \text{argmin}_{q \in \mathbb{R}^4} \sum_{i=1}^{N} (I_t(W(x,q)) - M_t(x))^2$$  \hspace{1cm} (12)

We denote $\tau$ as a small threshold that enforces the requirement that the result of the second gradient descent does not diverge too far from the result of the first. If it does, there must be a problem and so we act conservatively by not updating the template in that step. If the resulting discrepancy $\|q'_t - q_i\|$ is below $\tau$, it means the tracking results should be accepted and the color reference model $k'_t$ and edge model $p'_t$, which defined in Section 2, is updated by the latest model $k'_t$ and $p'_t$. While the resulting discrepancy exceed $\tau$, it means that there must be a problem and so we conservatively by not updating both the color and edge models in that step.

4 Experiments

The proposed particle filter based tracker has been implemented in C++ on a 2.6GHz Pentium4 PC with 1G memory. We have applied the proposed method to a variety of image sequences and the image size of the sequence is $360 \times 240$. In our experiments we select 14 facial feature points (see Fig.2). The experiments indicate that our method performs robustly to temporary occlusion and large facial movements.

Figure 2: The selection of facial feature points.

4.1 Compare proposed method with color-based particle filter tracker

The first experiment is implemented to prove the superior of our multi-cue particle filter. So we compare the multi-cue particle filter, which include both edge and color information, with the color histogram based particle filter. Both two trackers are executed to track an image sequence with more than 100 frames.

Both trackers perform reliably during tracking nostrils for the whole sequence. Parts of tracking results are illustrated in Fig.3. Fig.3 (a) shows tracking results of the traditional color based tracker. Some of tracking points are easy to drift away when the face appearance changes due to facial expression. Illuminate variation is another problem which
influences the tracking accurate. As the illuminate changed, tracking points are drifting away. Fig.3 (b) indicates the tracking results implemented by our multi-cue based particle filter. The results reveal that points tracked by our multi-cue tracker are much approach to the real facial feature points. And our tracker can robustly handle the non-rigid motion and the external disturbance due to large facial expressions and big illumination variations.

Figure 3: Tracking results. Frames 20, 40, 60, 80, 100 are shown. Upper frames are the results of the color-based only particle filter. Lower frames are the results of the multi-cue based particle filter.

Figure 4: Tracking results. Frames 30, 33, 36, 39, 42 are shown. Upper frames are the results of the enhanced KLT tracker. Lower frames are the results of the multi-cue based particle filter.

4.2 Compare proposed method with enhanced KLT tracker

The second experiment is to demonstrate the robustness of our approach due to temporary occlusion, which is regard as the main improvement of our tracking strategy. We compare our tracker with the enhanced KLT tracker [2] when the facial features suffer from temporary occlusion or the face is particle obstructed by other objects. Some parts of tracking results are illustrated in Fig.4.

As seen in Fig.4, when facial feature points tracked by KLT tracker, it will drift away through the occlusion. The drifting of points is disastrous because the lost of tracking points is unrecovered. In the frame 36, the KLT tracker loses some of tracking points due to the temporary occlusion. Although in the frame 42, the face is completely visible and cluttered facial features reappear again, drift features are impossible to be recovered. But it can be easily handled with our approach. Although tracking points got by our approach will also loss their tracking targets, our tracking strategies reduce the sample weight of such bad points and cuts down the influence of these occlusion points to the whole model. Moreover, the update method is able to judge those points as bad points and records the latest good texture template of them. When the target features reoccur, our method could retrieve proper tracking targets and resume the tracking process. Whenever the occlusion time is transitory, the point losing or drifting can be easily corrected by our approach. The high sampling rate is not required in our work, which ensures the real time computing. For instance, in Fig.4, only 40 samples are used for each facial feature point.
5 Conclusions

In the paper, we propose a multi-cue based particle filter to track facial feature points. In our approach, we apply both edge and color information to describe the tracking regions. The combination of edge and color features performs better to measure the facial features. Using of orientation histograms and sub-region color models instead of traditional histograms speeds up the computational process. Such multi-cue observation models pay much attention on the spatial layout information which intensifies the robustness of feature tracking. The primitive observation model makes the efficient tracking computation. In order to enhance the robustness, we propose a PDM to constraint face configuration. And an efficient model update algorithm is used to make the updating determination more propriety. The framework succeeds in solving several problems such as temporary occlusion and large movement errors in the tracking. Our tracker performs efficiently to achieve real-time facial feature points tracking. Experiments indicate that our method realizes more robust and efficient tracking results than other comparative facial feature trackers. In the future work, we intend to make use of our approach to combine with a fast appearance model [1] to improve the tracking performance.

6 Acknowledgements

The work was supported by the National Natural Science Foundation of China (No. 60575032) and the 863 Program (No. 2006AA01Z138).

References


