Online Updating Appearance Generative Mixture Model for Meanshift Tracking

Jilin Tu[1], Hai Tao[2], and Thomas Huang[1]

[1] Elec. and Comp. Engr. Dept.
Univ. of Illinois at Urbana and Champaign
Urbana, IL 61801
{jilintu, huang}@ifp.uiuc.edu
[2] Elec. Engr. Dept.
Univ. of Calif. at Santa Cruz
Santa Cruz, CA 12345
tao@soe.ucsc.edu

Abstract. This paper proposes an appearance generative mixture model based on key frames for meanshift tracking. Meanshift tracking algorithm tracks object by maximizing the similarity between the histogram in tracking window and a static histogram acquired at the beginning of tracking. The tracking therefore may fail if the appearance of the object varies substantially. Assume the key appearances of the object can be acquired before tracking, the manifold of the object appearance can be approximated by some piece-wise linear combination of these key appearances in histogram space. The generative process can be described by a bayesian graphical model. Online EM algorithm is then derived to estimate the model parameters and to update the appearance histogram. The updating histogram would improve meanshift tracking accuracy and reliability, and the model parameters infer the state of the object with respect to the key appearances. We applied this approach to track human head motion and to infer the head pose simultaneously in videos. Experiments verify that, our online histogram generative updating algorithm constrained by key appearance histograms avoids the drifting problem often encountered in tracking with online updating, that the enhanced meanshift algorithm is capable of tracking object of varying appearances more robustly and accurately, and that our tracking algorithm can infer the state of the object(e.g. pose) simultaneously as a bonus.

1 Introduction

Visual tracking of object in complex environments is currently one of the most challenging and intensively studied tasks in machine vision field. Various visual cues have been employed in tracking, such as motion flow, edge, color, depth, etc. As low level visual cues usually tend to be noisy, *a prior* knowledge of the object being tracked is usually applied as global constraints during the tracking. In [1] the appearance statistics of the object is modeled by an appearance eigenspace and a so-called Eigentracking technique is introduced. The success

of tracking is therefore largely dependent on the consistency between the actual object appearance and the *a prior* knowledge learnt off-line. This assumption however might be violated due to occlusion, or changing of illumination, etc.

In order to take the novelties into consideration during tracking, people proposed tracking algorithms with online model updating. [2] extended Eigen-tracking by online updating the object appearance PCA eigenspace using sequential *Karhunen-Loeve* algorithm. Noticing PCA eigenspace results from fitting subspace to data using L_2 norm, Ho[3] took a step further and suggested that fitting appearance subspace to data using L_∞ norm leads to subspace obtained by Gramm-Schmitt orthogonalization. The resulting algorithm incorporates observation novelties into subspace representation in a timely manner, and is able to track objects subject to pose changes, occlusions, and illumination variations, etc. Along the other direction, Jepson [4] proposed to model the appearance of an object as a mixture of stable image structure, outliers, and two frame information obtained from optical flow. An online EM algorithm is employed to infer the model parameters. The inferred stable image structure is adapted to model slow appearance variations of the object, such as variations caused by pose change, and illumination changes. Short time disturbances, such as occlusions, are modeled as outlier processes. While tracking with online learning has the advantage of handling occlusions and appearance variations, they all suffer from drifting problem more or less. The appearance model with online updating tends to drift away from the actual appearance of the object as the tracking error accumulates after tracking of very long period.

Comaniciu[5] proposed a meanshift tracking algorithm that tracks the object by comparing the similarity between histogram of the tracking window and a static histogram acquired before the tracking. Comparing to the other tracking techniques, this algorithm was well-known for real-time computation and robustness against partial occlusion. Afterwards people have proposed many extensions of this algorithm to accommodate different tracking scenarios based on different assumptions. Collins[6] first proposed to improve the ad-hoc kernel scale selection technique in mean-shift tracking algorithm by using scale space techniques. Zivkovic[7] reformulated the mean-shift process as a EM optimization process and the scale selection problem is solved as a variance estimation problem in a way similar to mean estimation. To avoid the distraction caused by background pixels in tracking window during mean-shift tracking, Porikli[8] proposed to weight the mean-shift kernel by foreground likelihood.

While all the extensions of mean-shift algorithms focuses on the adaption of kernel parameters, they all assume the histogram of the tracked object does not change much during the tracking. This assumption limited its application in scenario where the appearance of the object changes substantially. For example, the histogram of the frontal face of a person may be substantially different from that of the rear view of the person's head, therefore mean-shift tracker with histogram of the frontal face could become unstable when the person turns his face away from the camera. In [9], Birchfield attacked similar problem by using histogram intersection to blend both skin color and hair color when computing

histogram similarity. This idea however can not be applied directly in mean-shift algorithm due to different tracking mechanism.

In this paper, we propose to adapt the static histogram in meanshift tracking algorithm by modeling it as random variable generated by piecewise linear combination of some histogram pairs in a generative framework. The model parameters can be estimated using on-line Expectation Maximizition(EM) techniques. With the histogram updated online, the meanshift tracker is able to track object of vast varying appearances. In the mean time, the constraints of the key appearance histograms prevent the tracking from drifting. We applied our algorithms to human head tracking. The experiments indicate that our algorithms can achieve more robust and accurate tracking performance comparing to ordinary meanshift algorithm. In the mean time, the head poses are successfully inferred based on the generative model parameters inferred during the tracking.

We first brief meanshift tracking algorithm in Section 2. In Section 3, the framework of meanshift tracking with online histogram updating is introduced. Section 4 introduces our histogram generative model and online EM algorithm. Section 5 presents the experimental evaluation on human head motion tracking and pose estimation using meanshift tracking with/without our histogram updating technique. We summarize the benefits of histogram updating and discuss some future works in Section 6.

2 Meanshift tracking[5]

Suppose the appearance of the object is represented by normalized color histogram, denoted as $\mathbf{h}_1 = \{h_1(n)\}$, and the histogram of the tracking window centered at y be $\mathbf{h}_2(y) = \{h_2(y,n)\}$. The similarity between the two histograms can be represented by $\rho[\mathbf{h}_1, \mathbf{h}_2(y)] = \sum_n \sqrt{h_1(n)h_2(y,n)}$.

Denote a kernel centered at pixel p_i as $k(p_i)$, the Meanshift tracking algorithm can be summarized as follows:

1. Compute the histogram $\mathbf{h}_2(y_0)$ in the current frame, calculate $\rho_0 = \rho[\mathbf{h}_1, \mathbf{h}_2(y_0)] = \sum_n \sqrt{h_1(n)h_2(y_0,n)}$.
2. Compute likelihood ratio β_i between the current frame and the previous frame at each pixel in the tracking window : $\beta_i = \sum_n^N \delta[I(p_i) - n]\sqrt{\frac{h_1(n)}{h_2(y,n)}}$, i=1,..,R.
3. Compute the new location y_1 by meanshift $y_1 = \frac{\sum_i^R p_i \beta_i k(p_i)}{\sum_i^R \beta_i k(p_i)}$ and compute $\rho_1 = \rho[\mathbf{h}_1, \mathbf{h}_2(y_1)]$.
4. Quit with failure if $|\rho_1| < \epsilon_0$, quit with success if $|\rho_1 - \rho_0| < \epsilon_1$, else $y_0 = y_1$, goto 1.

3 Meanshift tracking with online appearance updating

As the template histogram $\mathbf{h}_1 = \{h_1(n)\}$ is kept static, the performance of meanshift tracking algorithm would become unpredictable in scenario where the appearance of the object has been undergoing huge variations.

A solution to this problem is to do online histogram updating. As we mentioned at the beginning of the paper, tracking with online model updating without constraints results in drifting problem. We therefore would rather constrain the online updating process by some key appearances acquired before the tracking. The key appearances can be acquired manually from some representative frames in the video. Or they can be acquired automatically. As tracking with online learning usually provides good performance for short clips without drifting problem, the tracked appearances in the tracking window can be clustered into key frames and be used by our algorithm for tracking video of very long period. Therefore our algorithm is an effective complement to the current available tracking tools.

The flowchart for meanshift tracking with histogram updating is illustrated in Figure 1. At frame t, meanshift tracking is carried out with an approximated histogram constrained on the manifold defined by key appearance histograms given the histogram observed in the tracking window of frame $t-1$. The approximated histogram is then updated based on the histogram observation in the updated tracking window of frame t. This procedure may iterate several times till the center of the tracking window converges. The question is now how to generate a histogram that approximates the observed histogram subject to the manifold constraints imposed by the key appearance histograms. We propose two bayesian inference approaches to attack this problem.

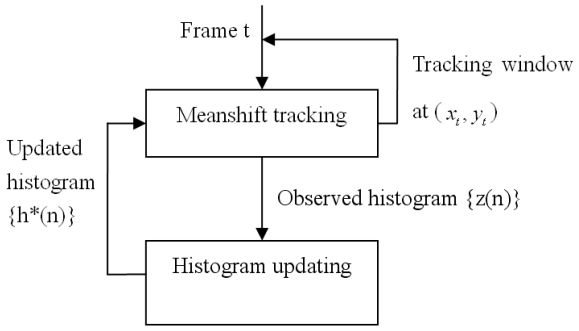

Fig. 1. The flowchart for meanshift tracking with histogram updating

4 Generating histogram from piece-wise linear combination of key appearance histogram pairs

Suppose K key appearances of the object can be acquired before the tracking. Denote their histograms as $\{h_1(n)\}$, $\{h_2(n)\}$,..., $\{h_K(n)\}$. And suppose the histogram of the object being tracked at current frame $\{z^*(n)\}$ can be piece-wise linearly approximated by some pairs of the key appearance histograms. The

formulation is thus as follows:

$$z^*(n) = \sum_{t=1}^{M} \{w_t h_{L(t)}(n) + (1 - w_t) h_{R(t)}(n)\}[m = t] \tag{1}$$

where [.] is a boolean operator, e.g. $[m = t] = 1$ if $m = t$, otherwise $[m = t] = 0$, m is a discrete hidden variable, $w_t \in [0\,1], t = 1, ..., M$ is the model parameter. $L(t), R(t) \in [1, ..., K]$ specifies the pairs of key appearance samples and defines the configuration of the appearance manifold that is piece-wise linearly approximated. The $\{L(t), R(t) : t = 1, ..., M\}$ pairs are specified by user according to domain knowledge. In the simple case where every pair of key appearance samples are considered, we have $M = K(K-1)/2$.

The bayesian generative model is illustrated in Figure 2.

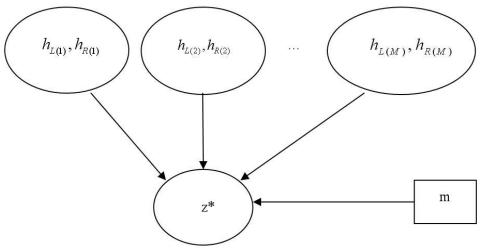

Fig. 2. The generative model for piece-wise linearly approximation of key appearances

Assuming gaussian distribution for simplicity, the joint distribution of the observation $z(n)$ at histogram bin n and the hidden variable m can be modeled as $P(z(n), m) = p(m)p(z(n)|m)$, where

$$p(m) = \frac{1}{M},\ for\ m = 1, ..., M$$
$$p(z(n)|m) = \prod_{t=1}^{M} [G(z(n); w_t h_{L(t)}(n) + (1 - w_t) h_{R(t)}(n), \Psi)]^{[t=m]} \tag{2}$$

where $G(; \mu, \Psi)$ denotes Gaussian distribution with mean μ and covariance Ψ.

We can conveniently obtain the *a posterior* probability of m given observation z,

$$p(m|z) = \frac{p(z, m)}{p(z)} = \frac{p(z, m)}{\sum_{t=1}^{K} p(z, t)} = \frac{p(z|m)}{\sum_{t=1}^{K} p(z|t)} \tag{3}$$

The expectation of log likelihood of the observation of histogram $\{z(n)\}$ is

$$E[LL(\{z(n)\}|m, \mathbf{w})] = \sum_{n} \sum_{m=1}^{M} p(m|z(n)) \log p(z(n), m)$$

$$\sim \sum_n \sum_{m=1}^{M} p(m|z(n)) \log G(z(n); w_m h_{L(m)} + (1-w_m) h_{R(m)}, \Psi)$$

.

Let $\frac{\partial E[LL]}{\partial w_m} = 0, m = 1, ..., M$, the following updating rule is obtained:

$$\hat{w}_m = \frac{\sum_n [z(n) - h_{R(m)}(n)][h_{L(m)}(n) - h_{R(m)}(n)] p(m|z(n))}{\sum_n [h_{L(m)}(n) - h_{R(m)}(n)]^2 p(m|z(n))}$$

Intuitively, we can tell this updating rule computes a probability weighted similarity measure between $\{z(n)\}$ and $h_{R(m)}(n)$, $h_{L(m)}(n)$.

If we further consider the past histogram observations under an exponential envelope located at the current time u, $C_u(k) = \alpha e^{-(u-k)/\tau}$, for $k \leq u$. $\alpha = 1 - e^{-\tau}$ so that $\sum_{k=-\infty}^{u} C_u(k) = 1$. The expectation of log likelihood of the observation of histogram $\{z_l(n) : l = -\infty, ..., u\}$ becomes

$$E[LL(\{z_l(n)\}|\{m_l, \mathbf{w}_l\}, l = -\infty ... u)] = \sum_{l=u}^{-\infty} C_u(l) E[LL(\{z_l(n)\}|m_l, \mathbf{w}_l)]$$

With the assumption that the histogram of the object does not change very quickly, we have the approximation $p(m_u = t|z_l(n)) \sim p(m_l = t|z_l(n)), t = 1, ..., M$ if time l and u are close enough. Taking the derivative of expectation of log likelihood, we obtain the updating rules

$$\begin{aligned}
D^1_{t,u} &= \alpha \sum_n [z_u(n) - h_{R(t)}(n)][h_{L(t)}(n) - h_{R(t)}(n)] p(m_u = t|z_u(n)) + (1-\alpha) D^1_{u-1} \\
D^2_{t,u} &= \alpha \sum_n [h_{L(t)}(n) - h_{R(t)}(n)]^2 p(m_u = t|z_u(n)) + (1-\alpha) D^2_{u-1} \\
\hat{w}_{t_u} &= \frac{D^1_{t,u}}{D^2_{t,u}}
\end{aligned} \tag{4}$$

Therefore given histogram $\{z_u(n)\}$ as observation and $\{\hat{w}_{t_{u-1}}\}$ as initialization of the model parameters $\{\hat{w}_t\}$ at frame u, the model parameters can be inferred as follows:

E-Step Compute $p(m|z_u(n))$ using Eq. 3 with $p(z(n)|m)$ defined in Eq. 2.
M-Step Compute $\hat{w}_t, t = 1, ..., M$ using Eq. 4,

Finally, the approximated histogram given current histogram observation $\{z(n)\}$is

$$h^*(n) = E[z^*(n)|z(n)] = \sum_{t=1}^{M} \{\hat{w}_t h_{L(t)}(n) + (1 - \hat{w}_t) h_{R(t)}(n)\} p(m = t|z(n))$$

Loosely speaking, $\{h^*(n)\}$ can be understood as the point closest to the histogram observation on the manifold approximated by the key frame histograms

in a probabilistic sense. We then use $\{h^*(n)\}$ as the color histogram template for meanshift tracking.

Suppose the histogram bin size is of $D \times D \times D$, and M pairs of key appearance histograms are specified, the computation complexity is asymptotically $O(MD^3)$ per iteration.

5 Experiments

One frequently encountered application scenario in human machine interaction is to track a person's head and to detect the person's head pose. The detection of the person's frontal face in particular can trigger some other face analyzing tools to reveal the person's identity, facial expression, eye gaze, lip movement, etc.

We find our algorithm a perfect application to this scenario as the head pose could be inferred directly according to the online updated histogram generative model parameters. For evaluation purpose, a video sequence is shot in which the subject moves his head around with different head poses starting with frontal view pose. The background contains a lot of shading, the color of which resembles the hair color, thus could be distraction of meanshift tracker. The frame size of the video is of 180 by 120. Because human head motion is relatively slow, the video is down-sampled to 4 frames/second.

For convenience of notation, the algorithms we are going to evaluate are indexed as follows:

MS_STATIC Meanshift algorithm with static histogram
MS_UPDATE Meanshift algorithm with histogram updating

We first applied algorithm **MS_STATIC** to the video. The histogram is computed in RGB color space with bin size $10 \times 10 \times 10$. The histogram bin size remains the same for the rest of the experiment. Similar to CAMShift in OpenCV[10], the window size is automatically adapted according to the 2-nd order moment of the object likelihood image. Some frames of the tracking result are shown in the first column of Figure 3. As template histogram is static and can not exactly characterize the appearance of the object in motion, the tracking window lags behind the head motion. The last 3 frames show that the shading in the background resembles the hair color and distracts the tracking window after the subject turns his head sideways.

To apply the meanshift algorithm with histogram updating, we acquired the human head appearances of frontal view, side view, and rear view before the tracking. Denote their histograms as $\{h_1(n)\}$, $\{h_2(n)\}$, and $\{h_3(n)\}$ respectively. We assumed that the histogram of the human head appearance at arbitrary pose can be approximated by either the linear combination of frontal view and side view histograms, or that of side view and rear view. The piece-wise linearly approximation model is thus formulated as

$$\begin{aligned} z(n) = &\{w_1 h_1(n) + (1 - w_1) h_2(n)\}[m = 1] \\ &+\{w_2 h_3(n) + (1 - w_2) h_2(n)\}[m = 2] \end{aligned} \tag{5}$$

We let $\alpha = 0.2$ so that the past 5-10 frames can be taken into consideration during on-line EM updating, and we empirically specified $\Psi = 0.1$. The key frames of the tracking result are shown in the second row of Figure 3. Comparing to the result of **MS_STATIC** in the first row, the new histogram updating mechanism enabled the meanshift tracker to track the head very closely when the head is turning away from the camera.

Fig. 3. Results for meanshift tracking with/without histogram updating. (a) **MS_STATIC**; (b)**MS_UPDATE**.

After histogram normalization, the approximation error between the observed histogram and the updated histogram is 0.164. Therefore the histogram updated with piece-wise linear combination constraint approximated the observed histogram in tracking windows pretty accurately.

As we collected appearance histogram for three key head poses(frontal, side, and rear views), we wish to infer these head poses through the estimated histogram generative model parameters. using the rule as follows taking Eq. 5 into consideration:

1. If majority vote of hidden variable m is frontal-side view combination, and $w_1 > T$, predict the head pose is frontal view.
2. If majority vote of hidden variable m is rear-side view combination, and $w_2 > T$, predict the head pose is rear view.
3. Otherwise, predict the head pose is side view.

The threshold T is set to 0.5 by default, but user may adjust it in practice.

Figure 4 compares the pose estimation accuracy against ground truth during the video. The ground truth is labeled by visual inspection. We can tell our algorithm was able to make correct estimation despite background clutters and illumination variations, except the estimation result is in general lagging behind the ground truth.

We also notice abnormality at frame 140 where the estimation predicted the ground truth when the subject is turning from rear view to side view. This is actually caused by tracking inaccuracy. The tracking at this frame is somewhat distracted by background clusters, and give inaccurate pose estimation which happens to be the pose which the subject is about to turn to.

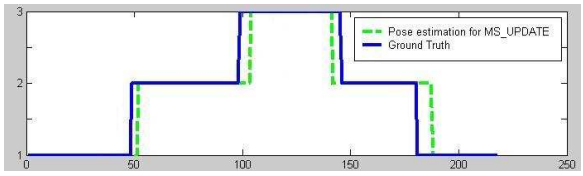

Fig. 4. Comparison of the pose estimation against ground truth for the whole video sequence. Frontal view–1; Side view–2; Rear view–3

Finally, we applied our algorithm to some video sequences provided by Birchfield[11], some key frames for tracking one of the video are shown in Figure 5. The video contains a lot of head movements and pose changes. The background contains a lot of clutters, and some clutters has color components resembles skin color. As the head moves, the shading on the face also varies. In the middle of the video, the subject waves yellow folders and hands in front of his face. Therefore it is a very challenging video for tracking and pose estimation. Our algorithm is able to track the whole sequence, and reaches pose recognition accuracy 77% after comparing to ground truth. Comparing to Birchfield's tracking result provided by [11], our algorithm is less likely to be distracted by background clutters and motion dynamics, and can provide head pose estimation as a bonus.

Fig. 5. More results for tracking and pose estimation with algorithm**MS_UPDATE**

6 Summary

In this paper, we proposed a generative mixture model and online EM updating algorithm for histogram updating. Experiment showed that, our model enabled meanshift tracking to achieve more robust tracking performance than that with static histogram. Based on the estimated model parameter, the object state(head poses) could be easily inferred.

Comparing to meanshift tracking with static histogram, meanshift tracking with histogram updating yields more robust and accurate tracking performance. Comparing to the past online learning techniques for visual tracking, our online EM algorithm with key appearance constraints avoids the notorious drifting problem. With the inferred model parameters, the object states(e.g. head pose) can be inferred as bonus.

Taking all these benefits into consideration, acquisition of more than one key appearances for the object, the only overhead added to the tracking algorithm, become worthwhile. Therefore our proposed online histogram updating technique for meanshift tracking is indeed an effective complement to the current tracking techniques. Besides, our proposed histogram generative model with its corresponding online EM updating algorithm is not confined by meanshift algorithm. It can be considered as an general object appearance model that can provide likelihood measure in other bayesian tracking frameworks.

References

1. Black, M., Jepson, A.: Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. Intenational Journal of Computer Vision **26** (1998) 63–84
2. Ross, D., Lim, J., Yang, M.H.: Probabilistic visual tracking with incremental subspace update. In: ECCV. Volume 2. (2004) 470–482
3. Ho, J., Lee, K.C., Yang, M.H., Kriegman, D.: Visual tracking using learned linear subspace. In: IEEE CVPR. (2004)
4. Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: Robust on-line appearance models for visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 1296–1311
5. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: IEEE Conf. Computer Vision and Pattern Recognition (CVPR'00). Volume 2., Hilton Head Island, South Carolina (2000) 142–149
6. Collins, R.: Mean-shift blob tracking through scale space. In: Computer Vision and Pattern Recognition (CVPR'03), IEEE (2003)
7. Zivkovic, Z., Krose, B.: An em-like algorithm for color-histogram-based object tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 1. (2004) 798–803
8. Porikli, F.: Human body tracking by adaptive background models and mean-shift analysis. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance. (2003)
9. Birchfield, S.: Elliptical head tracking using intensity gradients and color histograms. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (1998) 232–237
10. G.R., B.: Computer vision face tracking as a component of a perceptual user interface. In: IEEE workshop on Application of Computer Vision, Princeton (1998) 214–219
11. Birchfield, S.: (http://www.ces.clemson.edu/ stb/research/headtracker/)