ROBUST HAND TRACKING WITH POSTURE RECOGNITION VIA ONLINE LEARNING

Huasong Huang[1], Yulong Zhou[2], Pengjin Chen[2], Runwei Ding[2]

[1] Shenzhen National Engineering Laboratory of Digital Television Co.,Ltd.

[2] Engineering Lab on Intelligent Perception for Internet of Things(ELIP), Shenzhen Graduate School of PekingUniversity, Shenzhen, 518055 CHINA.

hshuang@neldtv.org, 1101213442@sz.pku.edu.cn, pengjinchen@126.com, dingrunwei@pkusz.edu.cn

Abstract: Robust hand tracking is a great challenge due to human hand's small size and drastic appearance changes. Recently, machine learning especially online learning methods have shown their promising ability in object tracking. This paper successfully achieved hand tracking under a lately popular online learning framework named Tracking-Learning-Detection (TLD) by a win-win thought that hand tracking and posture recognition can benefit from each other. Firstly, the object model is extended in order to import posture recognition which is done without extra recognition algorithms. In turn, the introducing of hand postures enhance hand tracking since the tracker is adaptive to different hand shapes. At last, skin color is sufficiently applied in every module (tracking, learning and detection) of TLD further improving the speed and accuracy of tracking. Experiments show that the proposed method works well on hand tracking with the additional ability to recognize some given hand postures under various difficulties.

Keywords: Hand tracking; TLD; Online learning;

1 Introduction

With the advancement in computer applications, it's attractive and feasible to set users free from keyboard and mouse. Naturally, human hand is the first choice to interact with machines. Generally, the machine needs to recognize human hand postures or gestures as a certain command in human computer interaction (HRI). Before the recognition, the hand should be detected and tracked first. The whole application process is shown in Fig.1 [1]. Its major difficulty stems from the fact that human hand is an articulated object with complicated shapes and appearance changes, which makes many excellent methods for tracking rigid objects not easily applicable.

Hand tracking methods can be classified into model-based and appearance-based ones [2, 3]. The task for Model-based methods is to construct the mapping from model configuration space to image feature space, transited into searching in a high dimensional space. Full or partial Degrees-of-Freedom hand appearance model can be built to track hand articulations with a high

978-1-4799-4719-5 /14/$31.00 ©2014 IEEE

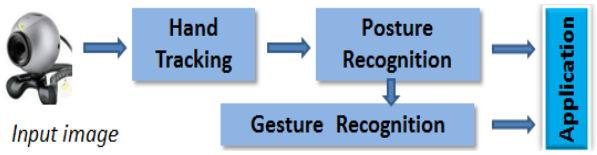

Fig.1 Human hand in HCI application process.

accuracy [4, 5]. Appearance based methods seek to build the mapping from image feature space to the hand appearance space. Skin color feature is often chosen for its simplicity [6] and contour features have been used in particle filter based methods [7]. Maximally Stable Extremal Regions is a new stable region feature also been used for hand tracking [8]. They are robust to appearance change to some degree, but cannot recover from full occlusion or tracking failures, when hand detection should take over. The detectors are usually classifiers trained offline with laborious and inefficient training process [9, 10]. An online learning framework would be more desirable, and should be feasible, as demonstrated by researches in rigid object tracking [11, 12].

To our knowledge, little work has been done to apply online learning methods to track articulated objects. Recently, Z. Kalal et al [14] have built an online learning framework named Tracking Learning Detection (TLD) which was promising in tracking rigid objects. In [12], Kalal successfully extends TLD to long-term face tracking in unconstrained environment. However, TLD fails when it comes to non-rigid objects like human hand with dramatic appearance changes. In this paper, we try to achieve robust hand tracking under this online learning framework. The reason we choose TLD framework is that it's robust against background clutter, camera moving, partial occlusion, hand rotation, out-of-plane rotation, and can recover from out-of-view motion or tracking failures while tracking rigid objects, besides, it's real-time. To overcome it's weakness in tracking articulated objects, we come up with a win-win idea that hand tracking and posture recognition can benefit from each other. That is, by importing hand postures, TLD can track hands well and posture recognition is also achieved as a plus without any additional recognition algorithm. Also, skin color is very useful in tracking human hands. In our method, not only in detection but also in every module of TLD does skin color be applied to better the tracking performance.

The rest of the paper is organized as follows. Section 2 states the TLD framework and its analysis in hand tracking. The proposed method is shown in section 3. Section 4 exhibits experimental results on both hand tracking and posture recognition. Conclusions are made in section 5.

2 Analysis of the TLD framework in hand tracking context

2.1 Tracking-Learning-Detection Framework

The TLD framework combines tracking, online learning and detection based on monitoring tracking errors. The online learning of an object is achieved by building an adaptive object appearance model which is a collection of object and background patches under "P-N experts" [15].

A so-called "Median Flow tracker" based on Lucas-Kanade optical flow method is used as a base tracker. A grid of 10×10 points are selected as feature points that are to be tracked. To evaluate the validation of the tracked feature point, a forward-backward error [16] method is proposed for tracking failure detection.

A three-stage cascaded classifier is applied for detection. The first stage is a patch variance filter to reject patches with lower variance. Patches passing through the variance filter are evaluated by a fern classifier [17, 18]. At last, a nearest neighbor (NN) classifier determines which patch is most likely to be the object.

The object model in TLD is an adaptive model with online learnt samples:

$$M = \{p_1^+,\ p_2^+, \cdots p_m^+, p_1^-,\ p_2^-, \cdots p_n^-\} \tag{1}$$

where p^+ and p^- represent the object and background patches respectively.

In learning process, a "P-expert" may generate a positive example by exploiting the temporal structure in image sequences, meanwhile, a "N-expert" may generate several negative examples by exploiting the spatial structure in image sequences.

2.2 Analysis of TLD in Hand Tracking

TLD learns a hand appearance in the first frame as the first positive example in object model which will influence the following tracking deeply. For example, if the hand in first frame is a fist, then TLD works well and learns different views of the fist as soon as the hand keeps its posture as a fist. However, when the fist suddenly changes to a scissor, or a palm reappears after the fist is out of view, TLD cannot recognize the new hand posture and fails. So, if TLD knows that the fist, scissor, claw or other hand appearances are just different postures of human hand, then it will perform good hand tracking. Additionally, in hand tracking and posture recognition algorithms, skin color is mostly used owe to its stable character in cluster environment. In original TLD, patch containing only partial hand is sometimes learnt due to the drift problem which would never happen when skin color is implemented as a restriction. Also, skin color is helpful to speed up tracking and enhance the precision of TLD. Details of the importing of hand posture and integrating skin color will be elaborated in the next section.

3 Extended TLD for hand tracking

3.1 An Extension of the Object Model

Hand posture is brought in mainly through extending the object model of TLD. Considering that the original object model only contains positive and negative examples, we divide positive examples into K subsets. Each subset represents a kind of hand posture like fist, scissor, claw and so on. So, the object model becomes what we call hand appearance model:

$$M = \{P^{(1)},\ P^{(2)}, \cdots P^{(K)}, p_1^-,\ p_2^-, \cdots p_n^-\} \tag{2}$$

where

$$P^{(k)} = \{p_1^{(k)}, p_2^{(k)}, \cdots, p_{m_k}^{(k)}\},\ k = 1, 2, \cdots, K \tag{3}$$

is a collection of the learnt appearances of the k^{th} hand posture. p_i^- still represents the background patch. There are K hand postures in total. It degenerates to the original TLD when *K*=1.

Given an arbitrary patch *p* and hand appearance model *M*, the similarity measurements become:

1) Similarity with the k^{th} hand posture nearest neighbor:

$$S^{(k)}(p, M) = max_{q\in P^{(k)}} S(p, q).$$

2) Similarity with the positive nearest neighbor:

$$S^+(p, M) = \max\{S^{(k)}(p, M) : k = 1, 2, \cdots, K\}.$$

3) Similarity with the negative nearest neighbor:

$$S^-(p, M) = max_{p_i^-\in M} S(p, p_i^-).$$

4) Similarity with the k^{th} hand posture nearest neighbor considering η (ex.50%) earliest patches of the k^{th} hand posture:

$$S_\eta^{(k)}(p, M) = max_{p_i^{(k)}\in P^{(k)} \wedge i<\eta m_k} S(p, p_i^{(k)}).$$

5) Similarity with the positive nearest neighbor considering η earliest patches of each hand posture:

$$S_\eta^+(p, M) = \max\{S_\eta^{(k)}(p, M) : k = 1, 2, \cdots, K\}.$$

6) Relative similarity with the positive nearest neighbor:

$$S^r = \frac{S^+}{S^+ + S^-}.$$

7) Conservative similarity with the positive nearest neighor:

$$S^c = \frac{S_\eta^+}{S_\eta^- + S^-}.$$

Alogrithm 1 Posture Recognition

1:**INPUT:** hand appearance model *M*, Test Patch *p*
2:**OUTPUT:** *I:* Identification of hand pose or non-hand
3:Calculate Conservative Similarity S^C, record the subscript *i* where $S^c = S(p, p_i)$;
4:**if** $S^c < threshold\_nn$ **then**
5: I = 0(meaning non-hand); Return;
6:**end if**
7:**for** k = 1 : K **do**
8: **if** i ∈ $[a_k, b_k)$ **then**
9: I = k (meaning the k^{th} posture); Return;
11: **end if**
12:**end for**

here, in 1), $S(p, q)$ is the similarity of two image patches, generally the normalized cross correlation (NCC) is used to compute the similarity. S^c is calculated to measure the similarity of a patch *p* with the model *M*.

The algorithm needs a marking function to indicate the start and end position (a_k and b_k) for examples of the k^{th} hand posture in the hand appearance model. When a patch measured by NN is the nearest neighbor of the i^{th} example in the hand appearance model, the most probable posture of the measured patch is returned according to the marking function. When an example of a hand posture is learnt to update the classifier and hand appearance model, the marking function is also need to be updated. The posture recognition based on marking function is illustrated in algorithm 1. The indicators $b_k = a_{k+1}, k = 1 \cdots K - 1$.

3.2 Skin Color for Tracking, Learning and Detection

Skin color information is widely applied in hand tracking and posture recognition. Traditionally, it's used for detecting and segmenting human hand. Here, a skin color map for each frame is obtained first based on the back projection [19] of hue histogram. Then, the skin color map is employed in every module of TLD: tracking, learning, detection.

Skin color map for tracking. Notice that a positive patch containing a hand also contains background since human hand is of distinct and irregular shape. Thus, points to be tracked may belong to background other than the hand. What's worse, these negative points could be tracked well since the background is stable. As a result, the forward-backward error cannot kick out these points which would influence the result of tracking. With skin color map, a point is easily kicked out before being tracked if the value of the point on skin color map is less than a given threshold.

Skin color map for learning. Online generation of positive and negative examples is an essential module of TLD. In the original TLD, only temporal and spatial constraints are employed to correct the labeling of the patches. This will decrease the learning efficiency in hand tracking context on account that patches with half hand and half

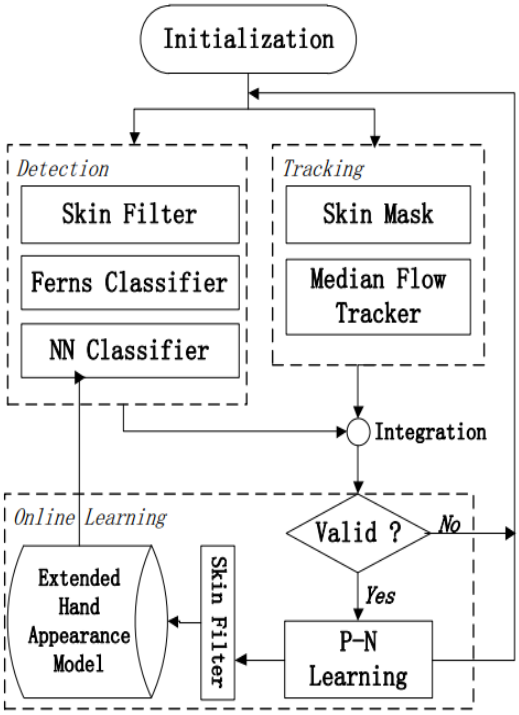

Fig.2 The overview of the proposed method

background are sometimes learnt as a positive example. This problem can be solved using skin color map. Patches labeled as positive examples with skin color pixels less than a threshold are relabeled back into negative examples.

Skin color map for detection. Since most existing methods use skin color cue to locate the hand roughly. Our method regards the skin color map as a first-stage filter replacing the variance filter, since the variance filter is suitable for tracking textured objects but not appropriate for hands with less texture. Only if the value of the patch's skin color pixels exceeds a threshold, can the patch be passed on to the second stage, otherwise, it is discarded.

3.3 An Overview Of The Proposed Method

An overview of the proposed method is summarized in Fig.2. Under TLD framework, the proposed hand tracking method consists of three modules: tracking, learning and detection. In initialization stage, TLD is told about the kind of hand postures, which is done by initializing the extended hand appearance model. In every frame, the detector and tracker work in parallel and their outputs are integrated for a further decision. In detection, if an image patch passes all the three cascaded detectors, then it's regarded as a hand area. In tracking, the points located within the hand area are tracked and the new hand patch is computed. After integration, if a hand posture is detected, the posture sample is learnt online by the "P-N" experts [15] to decide whether to be

added into the corresponding k^{th} posture model $P^{(k)}$. The extended hand appearance model is used for the NN classifier.

Table I Evaluation on hand tracking performance with different difficulties, the best performance are labeled in bold.

Sequences		Seq.1	Seq.2	Seq.3	Seq.4	Seq.5
Main Difficulties		Partial Occlusion	Face Distraction	Fast Motion	Out-of-view Motion	Background Clutter
Total Number of Frames		1901	1757	2210	1869	2300
Proposed Method	Recall	**91%**	**89%**	72%	**92%**	**100%**
	Precision	**100%**	**91%**	80%	**90%**	**87%**
Hough Track	Recall	74%	83%	77%	72%	85%
	Precision	61%	83%	82%	76%	78%
CAMShift	Recall	84%	61%	**84%**	80%	86%
	Precision	79%	60%	**85%**	83%	87%
Original TLD	Recall	22%	21%	30%	25%	22%
	Precision	24%	18%	26%	23%	21%

4 Experiments and discussions

4.1 Experimental Preliminary

To evaluate the extended TLD method for hand tracking, 5 real-scene RGB sequences with resolution of 640× 480 pixels were built. All the sequences are under the difficulties of moving camera, scale change and changing postures, meanwhile each sequence suffers from some of the difficulties of occlusion, face distraction, fast motion, out-of-view motion and background clutter. In human computer interaction, four common postures which are fist, palm, claw and scissor may represent the instructions "Confirm", "Move", "Drag" and "Done" respectively. Thus in these sequences, the four common hand postures are tested and labeled 1, 2, 3, 4 correspondingly, as shown in Fig.3. The tracking performance is measured with recall (R) and precision (P):

$$R = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (4)$$

$$P = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (5)$$

where true and false positives are correctly and wrongly tracked hands respectively, and false negatives are tracking failure reports when the hand is in the view. The tracker is evaluated by the overlap [14] of the tracked hand patch and the ground truth (generally overlap > 60% indicates tracking successfully). In initialization, only a small data set of the given hand postures are trained.

4.2 Evaluation on Hand Tracking

The proposed method for hand tracking is compared with the original TLD, CAMShift and HoughTrack on our sequences [13, 19]. Table I exhibits the tracking performances of these methods. Some precisions in the tests are 100 percent because there is no false positive since the hand area is restricted with skin color strictly and the extended model distinguishes hand from face well. Recall is 100 percent because the hand is always in the scene and tracking failure never happens. Most of the false negatives are caused by motion blur or reappearing after out-of-view. With skin color constraint, false positives are mainly caused by the face distractions when the hand suddenly changes its posture over the face.

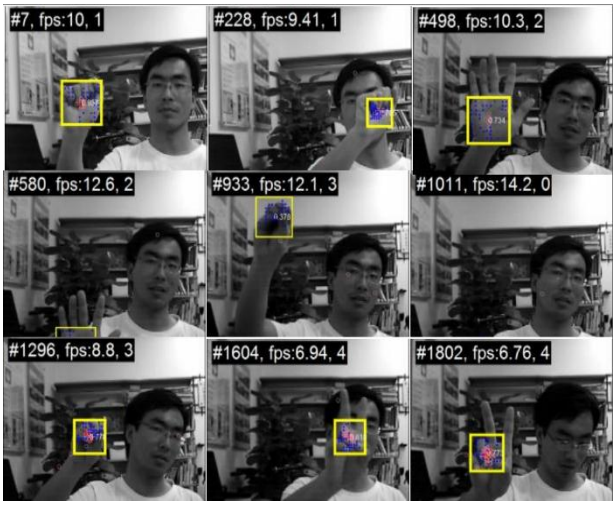

Fig.4 Tracking results of Sequence 1.

It can be concluded from Table I that the proposed method performs very well in hand tracking. Notice that both the recalls and precisions are very low in the original TLD because of the changing hand postures. Original TLD can only recognize one hand posture, thus it fails when posture changes. In fast motion sequence, the proposed method performs less well and CAMShift performs the best. CAMShift performs well even when there's motion blur, because the feature for CAMShift is thorough skin color, as a result, it performs badly when there's face distraction.

Fig.4 shows some output frames from sequence 1. It's concluded that the extended TLD for hand tracking is real time at a speed of about 10 frames per second and can do long-term tracking for at least 4 minutes. The result exhibits that the proposed method tracks the hand

accurately under posture changes, background clutter, face distraction (frame 228, 1604), partial occlusion (frame 580) and simultaneously recognizes hand posture as a plus. Due to the skin color, the feature points

Table II Different interpolation time of different methods

Hand posture		fist	palm	claw	scissor	total
Seq.1	Frames	463	402	439	507	1811
	Rate	96%	82%	88%	80%	86%
Seq.2	Frames	392	421	413	477	1703
	Rate	91%	77%	85%	71%	80%
Seq.3	Frames	602	485	496	533	2116
	Rate	83%	75%	70%	62%	73%
Seq.4	Frames	385	411	379	351	1526
	Rate	92%	68%	82%	83%	81%
Seq.5	Frames	523	498	569	527	2117
	Rate	94%	79%	86%	82%	85%

represented by blue points locate exactly on the hand region and thus drift problem seldom happens.

4.3 Performance on Posture Recognition

Although the goal of this paper is hand tracking, hand posture recognition is also achieved in our method, thus the performance on posture recognition is also evaluated. The recognition rates of the four hand postures on the five sequences are shown in Table II. Notice that the sum of the total frames of each posture is less than the number of total frames in each sequence because it's not taken into account when a hand posture is transiting to a new posture and in some frames the hand is out of view. It's calculated the mean recognition rate of all the sequences is 80.88%. There are two reasons for recognition failure. One is tracking failure and the other is recognizing one posture as another. The fist is both tracked and recognized well for its rigidity and clear texture. The recognition of palm is the worst mainly because some palms are tracked failed for the least texture on the palm. The claw or the scissor is sometimes recognized as other posture. In the experiment, we found that the recognition rate decrease as time going. This happens as the hand appearance model expands and the differences among these hand postures are not large enough. Despite of recognition failure sometimes, hand tracking performs well.

5 Conclusions

This paper successfully extends an online learning framework called tracking-learning-detection to hand tracking with recognition of given postures. Skin color is greatly exploited in every module of TLD improving the performance of hand tracking. The proposed method is robust against background clutter, camera moving, full-occlusion and out-of-view motion, which can rarely be well solved by the state-of-the-art methods. At the same time, the proposed method is real-time and can do the long-term tracking. More work would be done in the future. It is an challenge to learn hand postures online. Tracking double or more hands is a need for application. Trackers suit for less textured features like counter based tracker could be alternatives to the median-flow tracker in hand tracking.

Acknowledgements

This work is supported by National Natural Science Foundation of China (NSFC, No.60875050, 60675025), National High Technology Research and Development Program of China (863 Program, No.2006AA04Z247, 2012AA011705), Scientific and Technical Innovation Commission of Shenzhen Municipality (No.JCYJ20120614152234873, CXC201104210010A, JCYJ20130331144631730,No.JCYJ20130331144716089).

References

[1] P. Garg, N. Aggarwal and S. Sofat, "Vision Based Hand Gesture Recognition", World Academy of Science, Engineering and Technology, pp. 972-977, 2009.

[2] F. Mahmoudi and M. Parviz, "Visual Hand Tracking Algorithms", Proceedings of the conference on Geometric Modeling and Imaging: New Trends, pp.228-232, 2006.

[3] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review", Computer Vision and Image Understanding, Vol. 108, NO. 1-2, pp.52-73, Oct 2007.

[4] J. M. Rehg, T. and Kanade, "Digiteyes: Vision-based hand tracking for human-computer interaction", Proceedings of the workshop on Motion of Non-Rigid and Articulated Bodies, pp. 16-22, 1994.

[5] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla, "Model-based hand tracking using a hierarchical Bayesian filter", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, NO. 9, pp. 1372-1384, 2006.

[6] G. R. Bradski, "Computer Vision Face Tracking For Use in a Perceptual User Interface", Intel Technology Journal, NO. Q2, 1998.

[7] M. Donoser and H. Bischof, "Real time appearance based hand tracking", in Proceedings of the International Conference on Pattern Recognition, pp. 1-4, 2008.

[8] M. Isard, A. Blake, "CONDENSATION - conditional density propagation for visual tracking", International Journal of Computer Vision, Vol. 29, pp. 5-28, 1998.

[9] Eng-Jon Ong and R. Bowden, "A boosted classifier tree for hand shape detection", Proceedings of International Conference on Automatic Face and Gesture Recognition, pp. 889-894, May 2004.

[10] C. Tomasi, S.Petrov and A. Sastry, "3D tracking = classification + interpolation", Proceedings of International Conference on Computer Vision, Vol. 2, pp. 1441-1448, Oct 2003.

[11] S. Avidan, "Ensemble tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, NO. 2, pp. 261-271, Feb 2007.

[12] Z. Kalal, K. Mikolajczyk and J. Mates, "Face-TLD: Tracking-Learning-Detection Applied to Faces", IEEE Conference on Image Processing, pp. 3789-3792, 2010.

[13] Hong Liu, Wenhuan Cui and Runwei Ding, "Robust Hand Tracking with Hough Forest and Multi-cue Flocks of Features", International Symposium on Visual Computing, pp. 458-467, July 2012.

[14] Z. Kalal, K. Mikolajczyk and J. Mates, "Tracking-Learning-Detection", IEEE Transactions on

Pattern Analysis and Machine Intelligence, Vol. 6, NO. 1, pp. 1409-1422, Jan 2011.

[15] Z. Kalal, J. Mates and K. Mikolajczyk, "P-N Learning: Bootsrapping Binary Classifiers by Structural Constraints", IEEE Conference on Computer Vision and Pattern Recognition, pp. 49-56, Jun 2010.

[16] Z. Kalal, K. Mikolajczyk and J. Mates, "Forward-Backwoard Error: Automatic Detection of Tracking Failures", International Conference on Pattern Recognition, pp. 23-26, 2010.

[17] M. Ozuysal, M. Calonder, V. Lepetit and P. Fua, "Fast Keypoint Recognition Using Random Ferns", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, NO.3, pp. 448-461, Mar 2010.

[18] M. Ozuysal, P. Fua and V. Lepetit, "Fast keypoint recognition in ten lines of code", IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8, 2007.

[19] D. Comaniciu, V. Ramesh, P. Meer, "Real-time tracking of non-rigid objects using mean shift" IEEE Conference on Computer Vision and Pattern Recognition, vol.2, pp. 142-149, 2000.