REAL-TIME MULTI-CANDIDATES FUSION BASED HEAD TRACKING ON KINECT DEPTH SEQUENCE

Zhiting Yang[1], Yang Yang[1,], Yun-Xia Liu[2]*
[1]School of Information Science and Engineering, Shandong University, Jinan, China
[2]School of Control Science and Engineering, Shandong University, Jinan, China
yyang@sdu.edu.cn*

ABSTRACT

Considering depth images are robust to illumination variations with complex backgrounds, the paper developed a real-time head tracking system with one Kinect camera. Distance transform is applied to pre-processed depth frames to further reduce the effect of appearance deformation. A multi-candidates fusion strategy is proposed for template updating that assures head representation robustness. Two-stage template matching is adopted for computational efficiency in the searching procedure. In addition, an early termination criterion for template updating is presented to reliably improve the tracking accuracy. Abundant experimental results on our depth database demonstrate that the proposed method performs favorably against state-of-the-art methods in terms of robustness, accuracy, and efficiency.

Index Terms—Kinect, Head tracking, Multi-candidates fusion, Two-stage searching, Early termination

1. INTRODUCTION

Head detection is an active research area that has seen steadily improving performance in the last decades, with improving tracker complexity. It is considered as a canonical case of object tracking, which is one of the most fundamental problems in computer vision with numerous applications [1].

Many effective representation schemes have been proposed for robust object tracking on color images. For instance, the tracking-learning-detection (TLD) approach [2], the compressive tracker [3] and the generic object tracking approach [4]. However, performance of these trackers drops dramatically in complex backgrounds with illumination deformation and appearance change.

In contrast to color channels, the depth information demonstrates robust invariance to the existence of casting shadows and dynamic illuminations [5]. This provides an alternative to robust tracking. A contour cue based tracking method was proposed in [6] for real-time camera poses tracking. Reference [7] reports good results on hand tracking based on a single depth camera. Optical flow, color and depth data are involved simultaneously in the multiple cues combination (TMC) tracker in [8]. However, there are rare works about head tracking on depth images reported to the best of our knowledge.

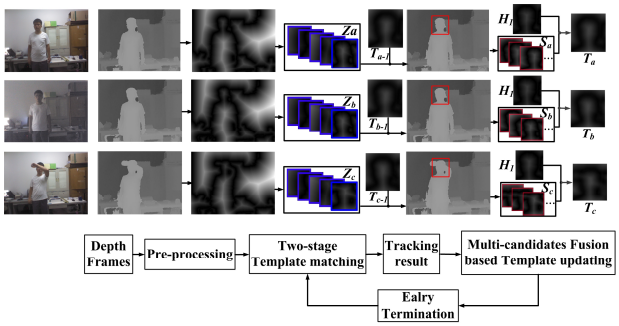

Figure 1. *Diagram of the proposed method under different illumination and deformation.*

In this paper, a real-time multi-candidates fusion based head tracking method is proposed (See Fig.1). In a typical head tracking scenario, a bounding box containing the head H_1 is manually initialized in the first frame. Certain pre-processing manipulations are necessary to enhance the quality of Kinect captured frames where distance transform in depth domain is adopted for better robustness. A correlation based two-stage template matching strategy with different sampling ratio is proposed for improved computation efficiency in the searching procedure. In particular, an early-termination template updating criterion is adopted which provides possibility for improve tracking accuracy to prevent abrupt and fast object motion. Finally, a multi-candidates fusion based template updating algorithm is proposed to integrate both information provided by ground truth H_1 and current frame. It serves as the matching template for tracking position estimation in the next frame.

The proposed method leads to an effective tracker that estimates the human head based on Kinect depth data only. With carefully designed parameters, it works computational efficiently in real-time manner. Extensive experimental results carried out on depth database show its effectiveness when compared with other stage-of-the-arts.

2. THE PROPOSED METHOD

In this paper, we aim at developing a head tracking method that is adaptive to significant appearance change without

being prone to drift on depth image sequences. To achieve this, we adopt a template matching based tracking paradigm, where patch based correlation is adopted as the similarity metric. Key techniques are explained in detail in the following sub sections.

2.1. Pre-processing

Depth frames collected by Kinect often suffer from degradation by several noise sources (e.g. environment noise, equipment noise, etc.) and occlusion [9]. In the proposed method, denoising is achieved by mean filtering for better smoothness and image quality of depth frames. A series of morphological operations including complement, erosion and dilation are adopted to fill in occlusion regions. Reader may refer to [10] for more implementation details.

To further reduce the effect of appearance deformation, we adopt distance transform to solve the problem of low level of discrimination of depth images and enrich detail information. Binary edge maps are first calculated, and then based on which the grayscale distance transformed images are obtained, indicating the distances between the pixel and nearest pixels lie on an edge. Euclidean distance is adopted to ensure invariance against deformation. As depicted in Fig.1, depth frames after pre-processing demonstrates robust invariance to complex background, illumination change and appearance deformation that are fatal factors in color features based tracking.

2.2. Candidate seeds generation

A bunch of depth image patches are selected as ***candidate seeds*** for template updating within a circular area centered at $c(H_i)$, where H_i denotes the depth patch of tracking head in the i-th frame, and $c(\bullet)$ denotes their center location.

Let Δc denote the number of pixels that the head location changes and the tracking result drifts between adjacent images at most. It is a key parameter that influences the tracking effectiveness and efficiency in a complementary manner. Our previous work [11] reveals that the number of pixels corresponding to each of 10 centimeters (decimeter, dm) under different distances conditions α was modeled as

$$\alpha = \frac{5.11\times10^4}{d_s - 56.5}, \tag{1}$$

where $d_s = (255 - D_i)\times16$ represents he physical distance d_s between head and camera, and the depth data D_i can be extracted easily on depth images.

With reasonable assumption that a common person's average walking pace is 1m/s (10 dm/s) and the Kinect camera captures 30 images per second, we have

$$\Delta c = \frac{10}{30}\alpha = \frac{1.7\times10^4}{d_s - 56.5}. \tag{2}$$

Taking into account abruptly rapid motion of the tracking object (is more than 1m/s), we also allow candidate seeds to be selected out of the radius Δc. The set of candidate seeds $CS_i = (X_{iN} \cup Y_{iU})$ can be randomly selected as

$$\begin{cases} X_{iN} = \left\{ X_{ij} \middle| \left| c(X_{ij}) - c(H_i) \right| \le \Delta c, j = [1,\cdots,N] \right\} \\ Y_{iU} = \left\{ Y_{it} \middle| \beta \le \left| c(X_{ij}) - c(H_i) \right| \le \varphi, t = [1,\cdots,U] \right\} \end{cases}, \tag{3}$$

where β and φ are inner and outer radius ($\Delta c \le \beta \le \varphi$) of the ring area, N and U are constant parameters that denote the cardinality of X_{iN} and Y_{iU}. We can tune the parameters freely to control the contribution of nearer and farther candidate seeds, also according to the characteristics of the depth sequences being considered.

2.3. Multi-candidates fusion based template updating

Given sets of candidate seeds being generated, it is common practice [3] [12] to treat nearer ones X_{iN} as positive samples and further ones Y_{iU} as negative samples. More and more complicated training classifiers (e.g. the compressive classifier in [12]) are designed to learn from features extracted from these samples. The classifier is updated frame by frame for object detection.

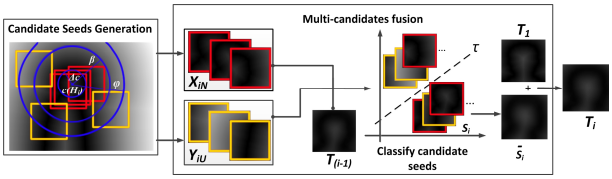

Figure 2. *Flowchart of multi-candidates fusion based template updating.*

However, the rationality behind the default settings of positive and negative samples according to the distance lacks sufficient evidence. Nearer patches at candidate seed positions may belong to the background, while those further ones may contain information that are related to a human head. These can be obviously observed in Fig.2. Besides, drifts of tracking objects worsen these mis-matches between real and appointed positive and negative samples, which could be fatal to success tracking.

We propose to classify the candidate seeds according to their "semantic" properties, that a head template is updated instead of the classifier. Patches extracted from candidate seed positions that are not related to the object to be tracked should be left out during template updating. Only those from related seed positions are utilized as ***candidates*** for fusion to generate the updated template for the next frame.

The proposed candidate determination algorithm can be formulated as following. For each candidate seed $c_{ip} \in CS_i = (X_{iN} \cup Y_{iU})$, $p = [1, \ldots, N+U]$, we calculate their correlation with the template of last frame T_{i-1},

$$corr_{ip} = \frac{\sum_m \sum_n \left(\left(c_{ipmn} - \overline{c_{ip}} \right) \left(T_{i-1mn} - \overline{T_{i-1}} \right) \right)}{\sqrt{\left(\sum_m \sum_n \left(c_{ipmn} - \overline{c_{ip}} \right)^2 \right) \left(\sum_m \sum_n \left(T_{i-1mn} - \overline{T_{i-1}} \right)^2 \right)}}, T_0 = H_1 \tag{4}$$

where m and n are the row and column indexes of image patches, $\overline{c_{ip}}$ and $\overline{T_{i-1}}$ are means of c_{ip} and T_{i-1} ,respectively. Then we get the set of candidates for the i-th frame as

$$S_i = \left\{ c_{ip} \,\middle|\, corr_{ip} \geq \tau \right\} \tag{5}$$

where τ is a threshold parameter. Then the multi-candidate fusion based template T_i at the i-th frame is derived by

$$T_i = \rho_1 \times \overline{S_i} + \rho_2 \times H_1 \tag{6}$$

where $\overline{S_i}$ is the mean of candidate sets, ρ_1, ρ_2 are weighting coefficients that satisfies $\rho_1 + \rho_2 = 1$. Template T_i is used for matching in the *(i+1)*-th frame to give the tracking result. Discussions on how to set the parameters τ and ρ_1 and how these settings influence the tracking results will be presented in detail in section 3.1.

2.4. Two-stage template matching strategy

The template matching module accounts for most computation cost of the whole algorithm. Huge computation burden would be insufferable if exhaustive search [13] is adopted. A two-stage template matching strategy is proposed to address the contradiction between tracking effectiveness and computation efficiency. See Fig.3 for the flowchart.

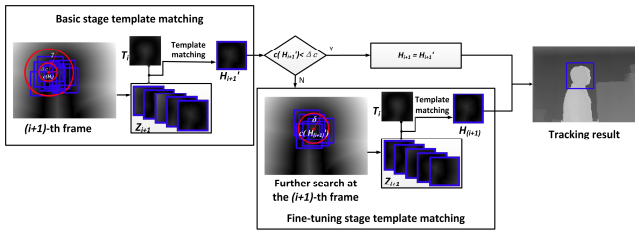

Figure 3. Flowchart of the two-stage template matching.

Different sampling ratios Δs are employed for nearer and further test samples, i.e.

$$\Delta s = \begin{cases} 1, & \text{if } \left| c(Z_{(i+1)k}) - c(H_i) \right| \leq \Delta c \\ 1/\delta, & \text{if } \Delta c \leq \left| c(Z_{(i+1)k}) - c(H_i) \right| \leq \gamma \end{cases}, \tag{7}$$

where γ denotes the search range and $Z_{(i+1)k}$ represents the k-th test sample in the *(i+1)*-th frame, $k = [1, \cdots, V]$ is the index for test samples. It has been reduced from $\pi\gamma^2$ to $\pi\Delta c^2 + \pi\delta^2 + \pi((\gamma - \Delta c)/\delta)^2$ or $\pi\Delta c^2 + \pi((\gamma - \Delta c)/\delta)^2$ as compared with the exhaustive search method.

According to this sampling rule, all test samples Z_{i+1} could be extracted and correlation between T_i

$$corr_{(i+1)k} = \frac{\sum_m \sum_n \left(\left(Z_{(i+1)k_{mn}} - \overline{Z_{(i+1)k}} \right) \left(T_{i_{mn}} - \overline{T_i} \right) \right)}{\sqrt{\left(\sum_m \sum_n \left(Z_{(i+1)k_{mn}} - \overline{Z_{(i+1)k}} \right)^2 \right) \left(\sum_m \sum_n \left(T_{i_{mn}} - \overline{T_i} \right)^2 \right)}} \tag{8}$$

could be computed. A *temporary* head location $c(H_{i+1}')$ that maximize the correlation coefficients could be obtained:

$$c(H_{i+1}') = \arg\max_k corr_{(i+1)k} . \tag{9}$$

We also record the maximum correlation coefficient as $corr_{max}$. This forms the ***basic stage*** of template matching.

A second ***fine-tuning stage*** is designed to enhance the tracking accuracy. We carry out exhaustive search centered at $c(H_{i+1}')$ within radius δ to obtain the more accurate head location $c(H_{i+1})$, and $corr_{max}$ should also be updated. Note that the second fine-tuning stage is optional and only take place when $|c(H_{i+1}') - c(H_i)| \geq \Delta c$. However, we find this quite effective in practical tracking applications.

2.5. Early termination criterion for template updating

Drift is an annoying phenomenon that usually occurs when the object being tracked undergoes significant appearance changes (e.g. drastic self-occlusion, abrupt motion, etc). Wrongly updating the template will lead to severe tracking degradation or failure for subsequent frames thus should be avoided even at the cost of low tracking accuracy.

We find $corr_{max}$ as an effective indicator for potential drift. Considering the prominent head detection performance of the correlation based matching, the threshold parameter τ in section 2.3 could be re-utilized as the criterion whether the updating template process should be interrupted, namely early termination criterion. In case $corr_{max} < \tau$ in the *(i+1)*-th frame, we can conclude that T_{i+1} is not similar to T_i. To prevent tracking drift, we stop template updating by setting $T_{i+1} = T_i$ and expanding γ to γ^* $(\gamma^* > \gamma)$ at the *(i+2)*-th frame. The early-termination criterion prevents the template from getting worse, thus the performance of the tracking system is improved.

3. EXPERIMENTS

We carry out tracking experiments on a self-built database [14] by our laboratory as most benchmark datasets [15] contain only color information. Six depth sequences by Microsoft Kinect 1.0 are captured on six people: Yang, Ma, Li, Zhang, Dong and Kong. There are several challenging situations including deformations, random motion, complex background and self-occlusion, while sequence Dong is obtained from continuously shifting the camera. These all adds to difficulties of success tracking.

3.1 Parameter settings of the proposed method

It is good property to be robust to parameter settings. In this subsection, we discuss the settings of τ and ρ_1 in an experimental manner (See Fig.4.). The success rate is employed as the evaluation measure for tracking effectiveness that is calculated on the basis of the PASCAL VOC challenge [16].

As can be observed in Fig.4 (a) that the success rate reached the highest peak at about 0.99 when $\tau = 0.6$, which is a quite promising result. Note that the slight increase in tracking success rate when $\tau > 0.75$ may be explained that early termination is always executed and the template is

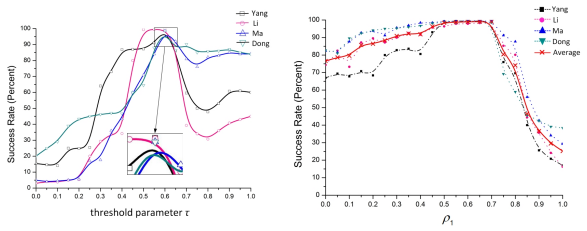

Figure 4. *Success rate curves with different* τ *and* ρ_1 .

seldom updated. In the presented method, τ is fixed to 0.6 for all the rest of experiments. In Fig.4 (b), the red curve indicates the averaged tracking success rate with respect to different ρ_1 settings. In case $\rho_1 = 0$ which means H_1 is adopted as the template for all frames and T_i is not updated at all, a relative success rate of 76% could be achieved. This verifies the effectiveness of the correlation based template matching method. On the other end, higher ρ_1 will result in severe drift and dramatic decline in success rate due to the critical lack of prior head information. The curve reaches the highest peak when ρ_1 is set to 0.5 ~ 0.7, which experimentally supported our multi-candidate fusion strategy to integrate both prior and data-derived information. In our experiments, ρ_1 is fixed to 0.7.

The rest of the parameters are set as following. The search radius Δc, β and φ for template updating are 10, 100 and 150 respectively. The cardinality of sets X_{iN}, Y_{iU} (N, U) are all set to 20. The searching radius γ is 50 and δ is 5 pixels in template matching module. A larger $\gamma^* = 70$ will substitute γ for the next frame if early termination occurs. All parameters of our method are fixed to all sequences.

3.2 Head tracking results

We compare the proposed tracking method with six state-of-the-art algorithms: TLD tracker [2], Semi-supervised tracker (SemiB) [17], Sparsity-based collaborative model (SCM) tracker [18], Compressive tracker (CT) [19], Incremental visual tracker (IVT) [20] and fast compressive tracker (FCT) [3]. Results are shown in Table 1, all based on implementations provided by the authors.

As clearly depicted in Table 1 that the proposed method yields the best tracking result. An average 99% success rate is achieved, which demonstrates its robustness. On the other hand, some tracker that works well in color channels [3, 19, 20] loses their superiority (below 50%) when directly applied to depth sequences.

For subjective assessment, Figure 5 shows part of tracking results of the proposed method under the varying circumstance of motion, self-occlusion, deformations and camera motion from top to bottom. It is safely concluded that our method is highly robust and adaptable. However others are not suitable for depth image tracking.

Table 1. *Tracking success rate comparison with six state-of –the-art tracking methods.*

Sequence	Frame number	Proposed	TLD [2]	SemiB [17]	SCM [18]	CT [19]	IVT [20]	FCT [3]
Yang	500	**100**	92	84	29	30	56	7
Ma	455	**98**	81	48	37	21	19	17
Dong	422	**98**	90	39	72	47	47	31
Li	468	**99**	96	84	96	53	35	34
Zhang	431	**100**	87	91	74	72	72	52
Kong	429	**99**	82	94	21	53	24	15
Average		**99**	88	73	54	46	42	26

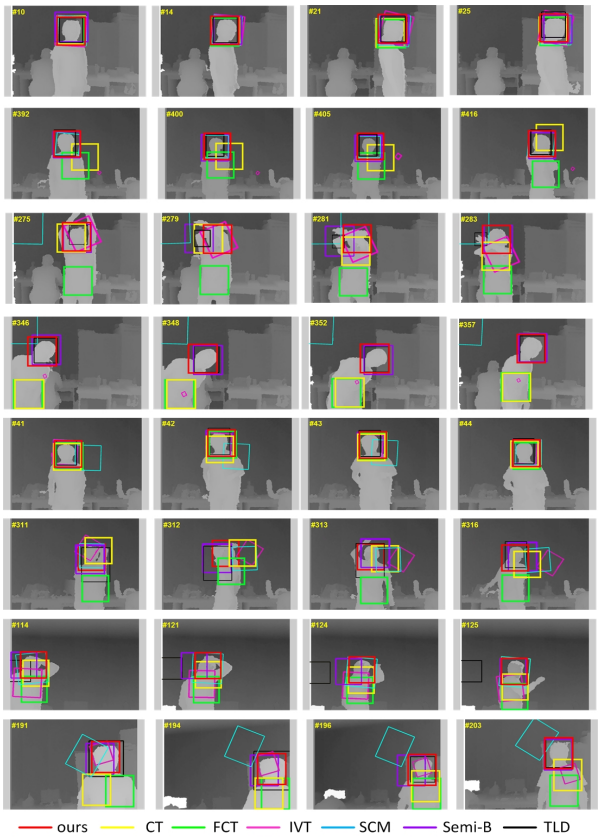

Figure 5. *Subjective comparison of tracking results.*

4. CONCLUSION

An effective real-time head tracking system based on Kinect depth sequences is proposed in this paper. The presented multi-candidates fusion strategy provides effective template updating. Two-stage template matching strategy and the early-termination criterion for template updating are adopt to further reduce the computational complexity and improve tracking accuracy, especially in drift handling. Robust head tracking is reported in experiments on our database.

5. ACKNOWLEDGEMENT

This paper is supported by the National Natural Science Foundation of China (Grant No. 61203269, 61305015, 61375084, 61401259 and 11474185).

6. REFERENCES

[1] C. Ma, X.K. Yang, C.Y. Zhang, and M.H. Yang, "Long-term Correlation Tracking," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5388-5396, 2015.

[2] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-Learning-Detection applied to faces," *IEEE International Conference on Image Processin*g, pp. 3789-3792, 2010.

[3] K.H. Zhang, L.Z. Zhang, and M.H. Yang, "Fast Compressive Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 2002-2015, 2014.

[4] P. Horst, M. Thomas, and B. Horst, "In Defense of Color-based Model-free Tracking," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2113-2120, 2015.

[5] M. Yang and S. Huang, "Appearance-Based Multimodal Human Tracking and Identification for Healthcare in the Digital Home, *Sensors*, pp. 14253-14277, 2014.

[6] Q.Y. Zhou, V. Koltum, and I. Labs, "Depth Camera Tracking with Contour Cues," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 632-638, 2015.

[7] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt, "Fast and Robust Hand Tracking Using Detection-Guided Optimization," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213-3221, 2015.

[8] Q. Wang, J. Fang, and Y. Yuan, "Multi-cue based tracking," *Neurocomputing*, pp. 227-236, 2013.

[9] K. Konolige and P. Mihelich, *http://wiki.ros.org/kinect_ calibration/technical*, Technical description of Kinect calibration, 2012.

[10] L. Xia, C. Chen and J. K. Aggarwal, "Human detection using depth information by Kinect," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 15-22, 2011.

[11] M. Li, Y. Yang and Y.X. Liu, "Robust 3D Human Tracking based on Kinect," *Chinese Control Conference,* 2015.

[12] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," *British Machine Vision Conference*, pp. 47-56, 2006.

[13] B. Babenko, M.H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1619-632, 2011.

[14] http://202.194.26.100/web2/yangyang/yzht/headtracking.htm.

[15] T. Liu, G. Wang, Q.X. Yang, "Real-time part-based visual tracking via adaptive correlation filters," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4902-4912, 2015.

[16] M. Everingham, L. Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object class (VOC) challenge," *International Journal of Computer Vision*, pp. 303-338, 2010.

[17] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," *European Conference on Computer Vision*, pp. 234–247, 2008.

[18] W. Zhong, H. Lu, and M.H. Yang, "Robust object tracking via sparsity-based collaborative model," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1838–1845, 2012.

[19] K.H. Zhang, L.Z. Zhang, and M.H. Yang, "Real-time compressive tracking," *European Conference on Computer Vision*, pp. 864–877, 2012.

[20] D. Ross, J. Lim, R. Lin, and M.H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, pp. 125–141, 2008.