FRAME-SUBSAMPLED, DRIFT-RESILIENT VIDEO OBJECT TRACKING

Xuan Wang[1], Yuhen Hu[1], Robert G. Radwin[2], John D. Lee[2]

[1]Dept of Electrical and Computer Engineering
[2]Dept of Industrial and Systems Engineering
University of Wisconsin – Madison
Madison, WI 53706

ABSTRACT

Performance-cost trade-offs in video object tracking tasks for long video sequences is investigated. A novel frame-subsampled, drift-resilient (FSDR) video object tracking algorithm is presented that would achieve desired tracking accuracy while dramatically reducing computing time by processing only sub-sampled video frames. A new pattern matching score metric is proposed to estimate the probability of drifting. A drift-recovery procedure is developed to enable the algorithm to recover from a drift situation and resume accurate tracking. Compared against state-of-the-art video object tracking algorithms, dramatic performance (accuracy) enhancement and cost (computing time) reduction are observed.

Index Terms **–** video object tracking, computing time, sub-sampling, drift-detection, drift-recovery

1. INTRODUCTION

Video object tracking (VOT) has received much attention due to a wide variety of applications including surveillance, non-invasive monitoring, visual data analytics [12]. Numerous algorithms have been proposed [1-6] and several large-scale benchmarks [7] and competitions [8-10] have also been proposed.

State-of-art video object tracking algorithms suffer from the drifting problem when the object being tracked gradually slips outside the tracking window as the tracking process continues. Existing robust video trackers [3, 4, 6] aim at reducing the probability of drifting. However, these trackers cannot detect onset of a drifting event, and would rely on manual correction to resume tracking.

Another important issue that has received little attention so far is the computation cost required for high performance video object tracking over very long video sequences. With the ever-increasing number of perpetual, surveillance and monitoring cameras installed on infrastructures, homes, and even human bodies, the amount of video data generated far exceeds all other type of data modalities combined [13]. Being a commonly used, first level video analytic tool, the task of video object tracking would consume tremendous computation power and computing time that is proportional to the number of frames in the video to be processed. For example, US Federal Highway Administration (FHA) has commissioned the Second Strategic Highway Research Program (SHRP 2) Naturalistic Driving Study (NDS) [11] to collect over two petabyte of video recordings of 3,400 drivers driving vehicles. Each video sequence lasts several hours. It would incur enormous cost (money and time) to manually supervise processing of each of these long video clips.

To address these two important issues in video object tracking, in this work, we propose an algorithm which leverages content-dependent prior information and accomplishes a frame-subsampled, drift resilient (FSDR) video object tracking algorithm.

A distinct feature of the FSDR algorithm is that it does not process a video sequence frame after frame continuously. Rather, it can perform video tracking on a video sequence sub-sampled at a rate of 1 per M (M> 1) consecutive frames. Since the number of frames that need to be processed is reduced by a factor of M, potentially, the overall processing speed may increase M times.

Another distinct feature of the FSDR algorithm is that we propose to use a Bayesian matching score to determine the likelihood of drifting. When the score falls below a threshold, it indicates drifting is likely occurring. Then, a drift-recovery procedure will be invoked to put the tracking "back-on-track". This procedure includes using known reliable templates or if applicable, generic object detection with the help of prior probability distribution of the object positions.

The paper is arranged as following: in section 2, related works are reviewed, and motivation is proposed. In section 3, prior information, subsampling and drifting issues are analyzed. In section 4, the FSDR algorithm is compared against state-of-art video object tracking algorithms. Discussion and conclusion are presented in section 5.

2. RELATED WORK

Recently, many video object tracking methods use an adaptive tracking-by-detection approach that formulate the task of visual object tracking a pattern classification problem. They also use online learning to update the object model [4]. Many of them made great and elaborate effort on model updating to avoid drifting. For example, the algorithm TLD [2] explicitly decomposes the long-term tracking task into three sub-tasks: tracking, learning, and detection. The detector corrects the tracker if necessary. And the learner estimates the detector's errors and learns from them to update the detector. Babenko et al [3] proposed to use multiple instance of the object template to reduce the template update error. In [6], a semi-supervised online boosting method is proposed to find a good model for updating. Self-paced learning [14] uses a self-paced curriculum-learning formalism to automatically select "right" frames for the classifier to learn the templates, so as to ensure the updated model covers the right appearance. Long-term correlation tracking [15] trains an online random fern classifier to re-detect object in case of tracking failure. STRUCK [4] uses kernelized structured output support vector machines to select and learn from targets online.

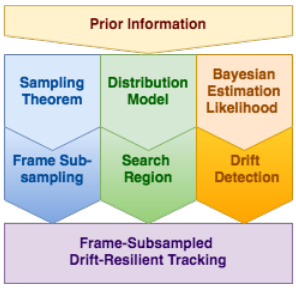

Figure 1 FSDR framework

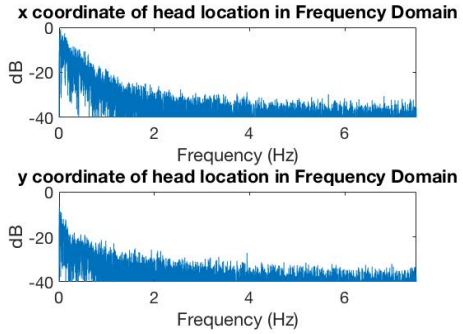

Figure 2 FFT of x and y coordinate of head location

These methods while trying to reduce the probability of drifting, do not explicitly detect the onset of drifting, nor provide any remedy while drift occurs. Thus, it is observed [7] that drifting is still an unavoidable problem in VOT. The drifting problem presents specific challenge when VOT is to be performed on very long video clips containing hundreds of thousands of frames. If drifting occurs early without being detected in time, remaining outcome would be erroneous, causing significant waste of time and money.

For processing large amount of video data, processing speed and execution time is also a key consideration. MOSSE [17] uses a new type of correlation filter, minimum output sum of squared error filter, and claims a high processing speed of hundreds of frames per second with robust performance. KCF [16] proposes an analytic model and diagonalizes the circulant matrix with Fourier transform to reduce both storage and computation. However, these correlation filter based trackers are more focused on mathematical model of filters to reduce computation load. Every frame in the video sequence is still processed despite significant temporal redundancy (correlation) among successive frames.

3. FRAME-SUBSAMPLED, DRIFT RESILIENT VIDEO OBJECT TRACKING

In the proposed FSDR VOT algorithm, an important assumption is that a set of training videos that exhibit similar characteristics of long, testing videos are available so that prior information may be exploited to facilitate faster, robust VOT. During the development, we will use the SHRP-II NDS driver state monitoring video as an example. These video clips are hour-long video clips taken from under the rear mirror in front of the driver. They record driver's head image during driving. The goal is to track the driver's head movement so that driver's state (distraction, sleeping, eating, phone call, etc.) may be detected. The major challenge is to achieve desired tracking performance while minimize the processing time.

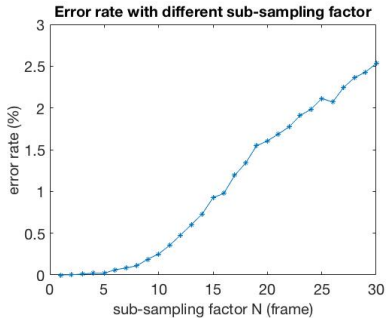

Figure 3 Error rate with respective to different subsampling factor

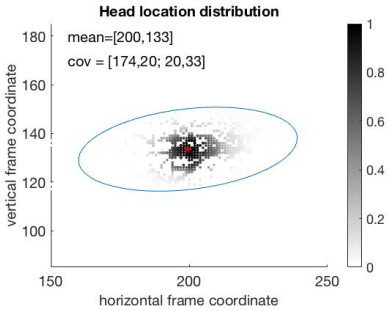

Figure 4 Head location distribution model with 3 (σ_x, σ_y) ellipse and mean value plotted

To achieve this goal, our proposed VOT tracking algorithm can be summarized in Fig. 1. It leverages strong prior information to facilitate three important improvements over existing VOT algorithms: frame sub-sampling, search region estimation and likelihood based drift detection.

3.1. Frame-subsampled rapid tracking

A key objective of VOT is to estimate the (x, y) position of the moving object as a function of time. If these time functions have band-widths (Nyquist rate) that are much smaller than (half of) the video frame rate, they may be reconstructed with sub-sampled sequence without incurring excessive tracking error. In Fig. 2, the spectra of manually annotated driver's head positions ($x(t)$ and $y(t)$) in the SHRP-II video HDEM25, which has a frame rate of 15 frames/second, are plotted. Clearly, most of energy concentrated within a band < 1 Hz. This implies that one may process a sub-sampled video sequence, say 1 in every N frames and then can reconstruct the entire track using interpolation.

An immediate benefit of frame-subsampling is saving of processing time by a factor of N. The concern of course, is whether this may lead to excessive tracking error. Assuming the tracking on the sub-sampled video sequence is done perfectly, the performance degradation due to interpolation is analyzed empirically. An interpolated tracking position is deemed as an error if it deviates from the annotated position by more than 20 pixels. In Fig. 3, the percentage error rate (# of error frames/# of interpolated frames) is plotted versus N, the sub-sampling ratio for $N = 1$ (no subsampling) to 30. Clearly the increase in tracking error due to interpolation is quite insignificant.

3.2. Prior distribution and head motion distribution

In many VOT applications, the trajectory of the object's movement may be confined into a specific region due to physical constraints

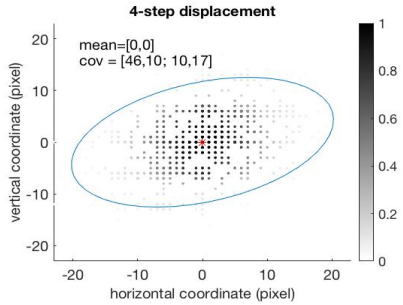

Figure 5 4-step head location displacement distribution

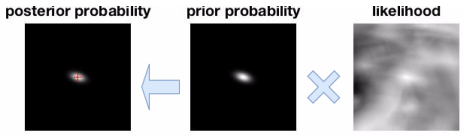

Figure 6 Probability score of each step of Bayesian estimation

or other prior knowledge. Such information can be exploited to enhance tracking performance. For example, in Fig. 4, the driver's annotated head positions in SHRP-II video HDEM25 are plotted and approximated by a normal distribution whose mean and ellipse corresponding to $3\sigma_x$, and $3\sigma_y$ are plotted. This prior distribution of head position will be very useful to help detecting whether drifting occurs during tracking. It also helps to resume tracking from a more reliable position once drifting is detected.

Next, we exploit the physical constraint the displacement of driver's head between neighboring frames must be limited. The empirical distribution of this displacement thus provides a good estimation of the N-step state transition probability $p(\mathbf{x}_k|\mathbf{x}_{k-N+1})$ where N is the frame sub-sampling factor. One example for $N = 4$ is depicted in Fig. 5. Again, this distribution is approximated by a bivariate normal distribution. The mean value of this distribution is at (0, 0), meaning the head's position remains unchanged.

3.3. Drift resilience

Drifting is the phenomenon that the estimated tracked location deviates from the ground truth and never returns. It is the most serious challenge of all video tracking algorithms and is often the results of a multitude of causes, including rotation, motion blur, background clutter, and occlusion. Previous approach for dealing with the drift problem is to develop so-called robust tracking algorithms that are less sensitive to these negative impacts.

In this work, we take a different approach: We develop a drift detection and associated drift recovery scheme to decide whether the current tracking estimate is the onset of a long run of drift. If so, drift recovery procedure will be invoked to discard the drifted position estimate and deduce a more reliable estimate using prior knowledge.

3.3.1. Drift Detection

Drift detection is formulated as a pattern classification problem using features derived from the tracking algorithm. Given the k^{th} frame image $\mathbf{z}_k$, a matching score S which is the posterior probability $p(\mathbf{x}_k|\mathbf{z}_{1:k}) \propto p(\mathbf{z}_k|\mathbf{x}_k)\cdot p(\mathbf{x}_k|\mathbf{x}_{k-N+1})$ will be evaluated over every pixel in the search range. In this work, we use cross-correlation of the latest template of the object and candidate template within the search region to estimate the likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$. However, other matching criterion that provides an estimate of $p(\mathbf{z}_k|\mathbf{x}_k)$ will be applicable as well. Fig 6 shows the posterior

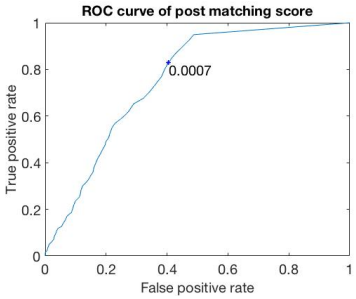

Figure 7 ROC curve for post matching score with the chosen threshold marked

matching score is proportional of the multiplication of the prior probability and the cross-correlation template matching score of the search region.

The position that maximizes $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ then will be designated as the Bayesian estimate of $\mathbf{x}_k$. Then $S(\mathbf{x}_k)$ will be used to determine if this estimate is likely the onset of a run of drift. We define drift as the situation when the estimated $\mathbf{x}_k$ deviates from the ground truth by more than 20 pixels. Then, we plot a receiver-operating curve (ROC) to determine a threshold h such that we determine drift is detected if $S(\mathbf{x}_k) \le h$. The ROC curve is plotted in Fig. 7. The threshold, which is closest to upper-left corner of the ROC, is chosen as the best threshold.

3.3.2. Drift Recovery

When a drift situation is detected, it implies the current template no longer contains the object being tracked. Several different drift recovery remedies may be applied: (a) Use known reliable templates which may include the template used to initialize the tracking, and templates that exhibit high matching scores $S(\mathbf{x})$ around $\mathbf{x}_{k-N+1}$ from previous frames. Matching scores will be evaluated for these candidates over the search area. (b) Use general object detection (such as a Viola-Jones (VJ) face detector [18]) to detect the presence and location of the object (face). Also incorporate the N-step state transition probability to focus on likely position of the object. In the experiments performed in this paper, drift recovery strategy (b) is used.

3.4. FSDR tracking scheme

A block diagram of the proposed FSDR tracking algorithm is shown in Fig. 8. It is initialized with a template detected using VJ face detector and the prior probability distribution of head positions p(x). Then a cross-correlation based tracking is applied to a sub-sampled video sequence with a sub-sampling factor N. The matching scores are closely monitored to detect onset of a drift. When a drift condition is detected, use the VJ face detector and p(x) to re-initiate the head-tracking.

When the FSDR tracking algorithm is to be applied to other types of VOT videos, the VJ head detector may not be applicable. Then the algorithm may be initialized with a manual specification of the object in the first frame. Then use this manually specified template to facilitate tracking. For drift recovery, a best-known template approach will be used.

4. EXPERIMENTAL RESULTS

Within the SHRP-II video, we used HDEM25 for training the FSDR algorithm including extracting prior probability, and N-step

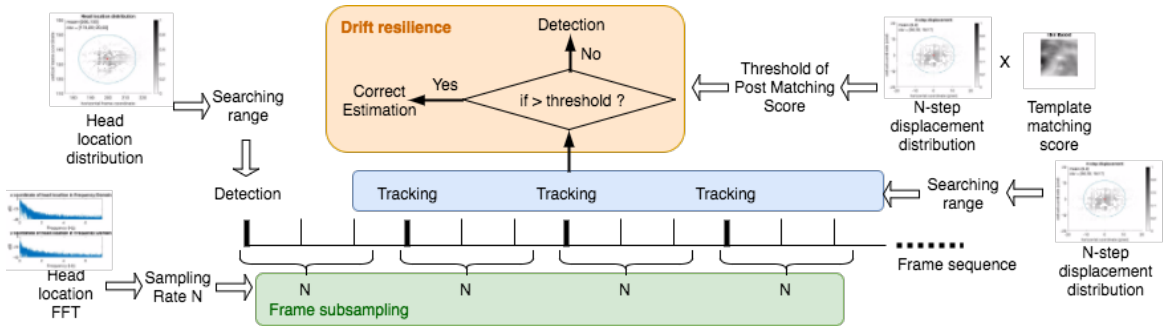

Figure 8 FSDR workflow

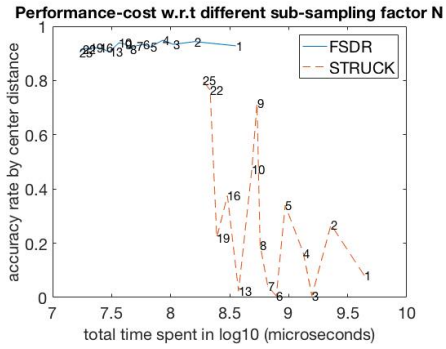

Figure 9 Performance-cost of FSDR and STRUCK

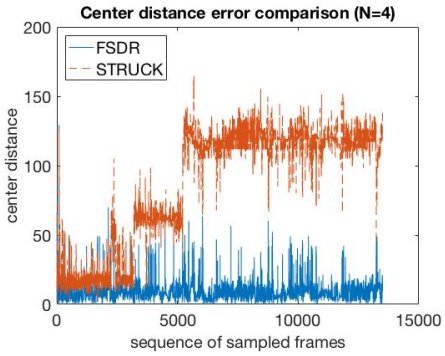

Figure 10 Center distance error comparison

state transition probability. We also use the training video sequences to derive the threshold for drift detection. We use a 60-minute (54000 frames @ 15frames/second) video, HDEM24, as the testing set. Drivers' head positions in all training data sets and the testing data sets are manually annotated. We define a tracking error as a deviation of more than 20 pixels between the tracked position and the ground truth position.

For comparison purpose, we also download the source code of a video tracker STRUCK [4], the champion of 2013 VOT challenge [8], and apply it to the HDEM24 video. The performance criterion is the percentage of frames in the testing video that are accurately tracked. The cost function is the total CPU time for execution of the tracking algorithm.

To investigate the effect of sub-sampling at different sub-sampling factor N, we tried $N = 1$ (no sub-sample) to 30. The result for FSDR tracker is summarized in Fig. 9. Two observations can be made: First, the computing time increases as N reduces. This is sort of expected. Secondly, the performance varies abruptly between adjacent values of sub-sampling factor N. But the performance gradually decreases as N increases.

For comparison, we overlay the performance-cost curve of STRUCK on the same curve of FSDR. It can be seen the curve for STRUCK varies much more wildly compared to that of FSDR. Overall, FSDR has much higher performance and much lower computing time compared to STRUCK.

A closer look at the experiment results reveals that these abrupt performance variation as shown in Fig. 10 is primarily due to drift occurring at different places of a video. If it occurs too early, the overall performance will be severely degraded.

5. CONCLUSION

In this paper we propose a frame-subsampled, drift-resilient tracker, which prevents drifting and achieves high tracking accuracy while dramatically reduces computing time on long video. The comparison with state-of-art trackers shows that this algorithm is efficient on large-scale long video data set. The adoption of prior information supports the sampled-tracking and drift-resilient scheme, which could greatly save time and ensure high accuracy. In future study, a drift-detection considering more classification criteria will be carried. Also, a drift-recovery step with more reliable and confident detectors will be studied. Besides, this frame-subsampled, drift-resilient scheme will be applied on more different types of trackers to explore the feasibility of this scheme.

6. ACKNOWLEDGEMENT

This material is based upon work supported by the Federal Highway Administration under contract number DTFH6114C00011.

7. REFERENCES

[1] Nam, Hyeonseob, and Bohyung Han. "Learning multi-domain convolutional neural networks for visual tracking." arXiv preprint arXiv:1510.07945 (2015).

[2] Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas. "Tracking-learning-detection." IEEE transactions on pattern analysis and machine intelligence34.7 (2012): 1409-1422.

[3] Babenko, Boris, Ming-Hsuan Yang, and Serge Belongie. "Robust object tracking with online multiple instance learning." IEEE Transactions on Pattern Analysis and Machine Intelligence 33.8 (2011): 1619-1632.

[4] Hare, Sam, Amir Saffari, and Philip HS Torr. "Struck: Structured output tracking with kernels." 2011 International Conference on Computer Vision. IEEE, 2011.

[5] Zhong, Wei, Huchuan Lu, and Ming-Hsuan Yang. "Robust object tracking via sparsity-based collaborative model." Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.

[6] Grabner, Helmut, Christian Leistner, and Horst Bischof. "Semi-supervised on-line boosting for robust tracking." European conference on computer vision. Springer Berlin Heidelberg, 2008.

[7] Wu, Yi, Jongwoo Lim, and Ming-Hsuan Yang. "Online object tracking: A benchmark." Proceedings of the IEEE conference on computer vision and pattern recognition. 2013.

[8] Kristan, Matej, et al. "The visual object tracking vot2013 challenge results."Proceedings of the IEEE International Conference on Computer Vision Workshops. 2013.

[9] LIRIS, France. "The Visual Object Tracking VOT2014 challenge results."

[10] Kristan, Matej, et al. "The visual object tracking vot2015 challenge results."Proceedings of the IEEE International Conference on Computer Vision Workshops. 2015.

[11] Kenneth L. Campbell, "The SHRP 2 naturalistic driving study," TR News 282, pp. 30-37, September-October 2012.

[12] Yilmaz, Alper, Omar Javed, and Mubarak Shah. "Object tracking: A survey."Acm computing surveys (CSUR) 38.4 (2006):

[13] Wang X, Hu Y H, Radwin R G, et al.
"Head tracking using video analytics." 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP). IEEE, 2015.

[14] Supancic, James S., and Deva Ramanan. "Self-paced learning for long-term tracking." Proceedings of the IEEE conference on computer vision and pattern recognition. 2013.

[15] Ma, Chao, et al. "Long-term correlation tracking." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[16] Wang, Naiyan, et al. "Understanding and diagnosing visual tracking systems." Proceedings of the IEEE International Conference on Computer Vision. 2015.

[17] Henriques, João F., et al. "High-speed tracking with kernelized correlation filters." IEEE Transactions on Pattern Analysis and Machine Intelligence 37.3 (2015): 583-596.

[18] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on.Vol. 1. IEEE, 2001.