Proceedings of the 2015
IEEE Conference on Robotics and Biomimetics
Zhuhai, China, December 6-9, 2015

Compressive Perceptual Hashing Tracking with Online Foreground Learning

Zheng Li[1†], Jian-Fei Yang[1†], Long Chen[1*] and Juan Zha[1]

Abstract— **This paper proposes a novel compressive sensing based perceptual hashing algorithm for visual tracking. Tracking object is represented by compressive perceptual hashing feature combined with patch-based appearance model. Besides, an updating foreground weight map is assigned for each object representation and the weight map is updated according to the accumulation of foreground pixel and distance between the foreground pixel and the center of the weight map. Based on the compressive perceptual hashing template and the weight map, our tracker searches the local region with the maximum response in an coarse-to-fine way. In addition, we introduce a visual attention knowledge that the object, namely foreground, should be always located in the center of the weight map, to handle the model drift problem. Extensive experiments demonstrate that the proposed tracking method achieves the state-of-the-art performance in challenging scenarios.**

I. Introduction

Visual tracking is one of the most key components for numerous robot applications, such as robot human interaction, robot navigation and autonomous driving, etc. Generative trackers and discriminative trackers are two main types of appearance-based trackers. The generative ones use a particular feature vector or subspace model to present the target object and search for region with the least reconstruction error from the target object. The discriminative trackers, namely tracking-by-detection, which treat the tracking problem as a local search detection problem is based on a binary classifier. Generally, the discriminative with prior data set could perform better but with external training cost.

Wu et al. [1] present a comprehensive evaluation of online trackers by 2013. Since then, several effective trackers have been proposed recently. Locality Sensitive Histograms Tracker (LSHT) [2] is a simple and real time tracking framework based on locality sensitive histogram method, which is robust to illumination changes. To address model drift problem, a multi-expert framework is chosen by [3], in which, an entorpy-regularized restoration scheme is utilized to correct undesirable effects of bad model updates for the base tracker. Nevertheless, object tracking is still a challenging problem under appearance changed situations caused by illumination changes, pose variation, occlusion and so on.

Object representation is the most crucial part of tracking problem. Numerous features and models have been chosen for object representation, such as Haar-like features [4], global integral histogram [5], locality sensitive histograms [2], sparse representation [6] and adaptive color attributes [7], etc. Unlike the strategy using complex appearance model to attain robustness of the object representation, we choose to represent object by a simple binary code constructed with the hashing technique for better efficiency of matching.

There are several researches on hashing-based tracking until 2015 [8], [9]. Fei et al. [9] proposed an object tracking approach using perceptual hashing algorithm (ahash, phash, dhash). However, pure perceptual hashing feature without patch strategy is unstable for various challenging scenarios. It is obviously a high-complexity and low-efficiency operation to extract features from all the patches with one-by-one way. We use a very sparse measurement matrix that asymptotically satisfies the restricted isometry property (RIP) in compressive sensing theory, thereby facilitating efficient projection from the image feature space to a low-dimensional compressive subspace.

As a dimensionality reduction manner, compressive sensing [10] has been introduced into visual tracking and achieves real time performance recently [11], [12]. In [12], a sparse measurement matrix is constructed to extract the efficient features from a multiscale image feature space. The tracking process is formulated as a binary classification by a naive Bayes classifier. Compressive tracker has good performance at some tracking situations in real time. But the performance is instable because of its random mapping process scheme.

Considering the complementary attributes of perceptual hashing based tracker and compressive sensing based tracker, we present a novel tracking mechanism by constructing compressive tracking framework with perceptual hashing patch-based appearance model. Additionally, we propose an online foreground learning method to address the drift problem.

The contributions of proposed tracking framework are as follows.

- In this paper, we propose a novel compressive perceptual hashing appearance model for robust and fast tracking.
- A novel online foreground learning method is proposed to handle the target drift problem, in which, every CPH template is also combined with an updating weight map for confidence evaluation.
- A visual attention knowledge, that the object should be always located in the center of the weight map, ie. the center of visual system, is first imported into the

[1] Z. Li, J. Yang, L. Chen and J. Zha are with Sun Yat-sen University, Zhuhai, Guangdong, P.R.China

[†] These two authors contributed equally to this work.

[*] Corresponding author. E-mail address: chenl46@mail.sysu.edu.cn (L.Chen)

tracking framework for model drift problem.

The rest of this paper is organized as follows. Section II describes three crucial components including discriminative compressive object representation in Section II-A, visual tracking framework in Section II-B and foreground learning in Section II-C. Experimental results and comparisons are shown in Section III. Section IV concludes the paper.

II. Compressive Perceptual Hashing Tracking

In this section, we present the proposed compressive perceptual hashing tracking algorithm as well as the weighted map updating by online foreground learning.

A. Discriminative Compressive Object Representation

1) Random projection and compressive sensing: To obatin complete object appearance representation, we are supposed to sample many multi-scale fragments covering different locations and sizes among the object, which can cause a high dimensional representation. Therefore, it is necessary to reduce the high-dimensional signal to lower-dimensional space without losing the significant information as well. To deal with it, we apply the recent significant concept of random projection [13] and compressive sensing [14]. In random projection, a random matrix $\mathbf{R} \in \mathbb{R}^{n \times m}$ whose rows have unit length projects data from the high-dimensional feature space $\mathbf{x} \in \mathbb{R}^m$ to a lower-dimensional space $\mathbf{v} \in \mathbb{R}^n$

$$\mathbf{v} = \mathbf{R}\mathbf{x} \tag{1}$$

where $n \ll m$. Each projection $\mathbf{v}$ is equivalent to a compressive measurement in the compressive sensing encoding stage. In our algorithm, $\mathbf{x}$ is composed of features of all fragments while $\mathbf{v}$ merely consists of certain fragments. The compressive sensing theory [14] affirms that it is possible to reconstruct the signal from a small number of random measurements if a signal is K-sparse. The sparse measurement matrix $\mathbf{R}$ preserves the salient information in any K-sparse signal when projection. Let $\mathbf{R} \in \mathbb{R}^{n \times m}$ be a random matrix with $\mathbf{R}(i,j) = r_{i,j}$ where

$$r_{i,j} = \sqrt{3} \times \begin{cases} +1 & \textit{with probability } \frac{1}{6} \\ 0 & \textit{with probability } \frac{2}{3} \\ -1 & \textit{with probability } \frac{1}{6} \end{cases} \tag{2}$$

which is generated in the initialization and restrict condition (RIP) [15], [16] for it is definitely necessary. The fragments at current are suitable for feature extraction at an excellent speed with little loss.

2) Patch-based appearance model: For each positive T_s^+ and negative sample T_s^- at a frame of time t, we adopt patch strategy based on multi-scale fragments to construct distinguishable representation. Each fragment is divided by patch strategy with the size of $l \times l$. The positions and sizes of fragments are specified in the initialization randomly, which are same for all samples in the frame sequence. And all patches are resized to 8×8 to extract perceptual hashing feature. By doing this, we further simplify the later stages of the procedure without losing too much of the structural information of the image, and also gain some measure of scale invariance.

3) Perceptual feature: Generally speaking, perceptual feature represents the characteristics of human vision. The research of cognitive psychology and human visual system demonstrate that human eyes are sensitive to illumination, color and texture information. Therefore, we construct our perceptual feature by means of intensity histogram and Discrete Cosine Transform (DCT) on behalf of illumination and contour profile. The perceptual representation of a sample consists of features of patches. Here comes the two kinds of perceptual features, intensity histogram and low-frequency energy spectrum.

The intensity histogram for a patch P is a 1D array, each of whose value is an integer representing the frequency of occurrence of particular intensity value. The corresponding histogram H_P is a B-dimensional vector defined as:

$$H_P(b) = \sum_{i=1}^{N} C(I_i, b), \quad b = 1, 2..., B \tag{3}$$

where N denotes the number of pixels and B is the total number of bins. Here we set B = 8. $C(I_i, b)$ is a binary function whose output is zero except when intensity value I_i belongs to bin b. Now that the local histogram H_P indicates grey information of a patch P.

The discrete cosine transform (DCT) has a strong "energy compaction" property and concentrates most of signals in a few low-frequency components of the DCT. The process of a DCT can be defined as:

$$d_m = \sum_{i=1}^{64} y_i \cos\left[\frac{\pi}{8} m \left(i + \frac{1}{2}\right)\right], m = 1, 2..., 64 \tag{4}$$

Based on Eq4, we can conclude that the low-frequency information is concentrated from the frequency spectrum. Owing that the rest of DCT coefficients prove to be useless, just eight values extracted from the upper left corner are considered as perceptual feature of profiles. Therefore, the 8-dimension vector D_P denotes the low-frequency feature of a patch P. The perceptual features of a patch P is the combination of them: $v_P = \begin{bmatrix} H_P & D_P \end{bmatrix}$. So far, the whole perceptual features matrix of a sample have been extracted:

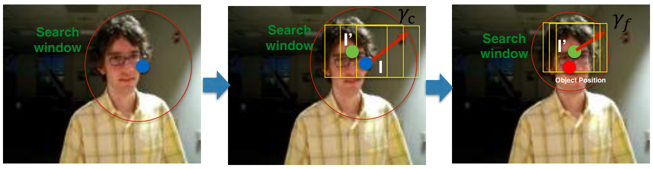
$$V = \begin{bmatrix} v_{P_1} & v_{P_2} & ...v_{P_k} \end{bmatrix} \tag{5}$$

where k denotes the total number of patches.

4) Binary matrix generation by locality-sensitive hashing: To compare different perceptual feature of images conveniently, the perceptual features matrix is necessarily processed by hashing method, one of which is locality-sensitive hashing (LSH), widely used in similarity search [17]. For each patch feature $v_P \in \mathbb{R}^{16}$, we construct m corresponding hashing functions. one of which is define as follows:

$$h_{P_i}(v_P) = sign(w_{P_i}^{\mathrm{T}} v_P + b) \tag{6}$$

where $w_{P_i} \in \mathbb{R}^{16}$ is generated randomly between $[-1, 1]$ satisfying *Gaussian distribution* and b is set to 0. Each patch can generate m binary codes through W_P of m hashing functions Here for all patches, we share the same W_P and just need a matrix multiplication operation to improve



Coarse to fine search Coarse search Fine search

Fig. 1: The principle of coarse-to-fine search.

efficiency, which can enhance computing efficiency a lot. Through LSH, we generate an unique perceptual image representation (PIR) for each sample and can further conduct confidence evaluation.

B. Visual Tracking with PIR

Now that the perceptual hashing feature is extracted, we build two main models to measure the difference and establish the learning strategy.

1) Appearance model: Given object O at frame $t-1$ and current frame t, tracking algorithm is required to locate the object O at t frame. In the former description, we clarify that O is presented by postive templates $\{T_s^+\}_{s=1}^{T^+}$ and background G is represented by negative templates $\{T_s^-\}_{s=1}^{T^-}$. Tracking task is to find the target location of a candidate C in t frame which is most similar to positive templates but most dissimilar to negative templates. Between candidates and templates, we define the discriminative similarity by as follow:

$$s(C,T_s) = \sum_{i=1}^{k} W_{P_i}^t (1 - \frac{1}{l^2} \left\| \mathbf{h}(P_i^C), \mathbf{h}(P_i^{T_s}) \right\|_H) \tag{7}$$

where l stands for the size of the patch and $W_{P_i}^t$ represents the foreground weight of i-th patch at t frame. The foreground weight, accounting for the weight of a patch in a candidate, will be introduced in Section II-C. Comprehensively, the confidence of j-th candidate C_j is defined by the average distance to all fragments as follows:

$$Con(C_j) = \delta(\frac{1}{T^+} \sum_{s=1}^{T^+} s(C,T_s^+) - \frac{1}{T^-} \sum_{s=1}^{T^-} s(C,T_s^-)) \tag{8}$$

where δ is a normalized coefficient, which lets the confidence constraint in $[-1,1]$. The proposed appearance model summarizes samples and object of the consecutive frames, which offers convenient and efficient conditions for tracking updating.

2) Tracking model: The task of tracking model is to update the object location at current frame in virtue of PIR and appearance model. Thus, we generate the search candidates by coarse-to-fine search and construct the tracking model by Bayesian framework.

Sample and coarse-to-fine search: As we know the ground truth I_1 at the first frame, we define that the positive sample $D^a = \{Z| \, \|I(Z) - I_1\| < \alpha\}$ and negative sample $D^{\beta,\xi} = \{Z|\xi < \|I(Z) - I_1\| < \beta\}$, where Z denotes the sample position and α, β, ξ denotes the sample distance threshold with $\alpha < \xi < \beta$ respectively. At the second frame, we choose the candidates as $D^{r^c} = \{Z| \, \|I(Z) - I_1\| < r^c, \Delta_c\}$ where r^c is the coarse radius and Δ_c is the coarse shifting step. Then we calculate the confidence of each candidates and select the maximal response one as the center of fine search circle $I_1{}'$. The fine search runs as $D^{r^f} = \{Z| \left\| I(Z) - I_1{}' \right\| < r^f, \Delta_f\}$ where r^f is the fine radius and Δ_f is the fine shifting step. After maximum confidence selection, we regard the results of fine search as the object position at second frame. From then on, the search is executed repeatedly as the same way in Fig 1.

At each frame, the estimate of the target $\hat{X}_t$ is defined by the MAP estimate over M samples:

$$\hat{X}_t = argmax_{Con^{(i)}_t} Con_t^{(i)}(C_i), \forall i \in [1,M] \tag{9}$$

Through the similarity measurement of candidates and fragments and probability selection, the most possible position of candidates is assured.

C. Foreground Learning for Object Tracking

1) Weighted map updating: In the tracking region of the current frame t, the foreground is extracted and recorded in weighted map W, all of whose elements, totally the same quantity of pixels, are initialized to 1 at the beginning. We are urged to model focus of attention that is motivated by the biological visual system which concentrates on certain image regions requiring detailed analysis [18], [19]. As presented in [20], Z.Zivkovic makes contributions on an improved adaptive Gaussian mixture model for background subtraction, which extracts the foreground under multivariate circumstances. With the simple method, we obtain the region of foreground and compute its pixel center as $F_t(x_f, y_f)$ at frame t. The weighted center K_t is updated by W^t. For each pixel $K_p(x,y)$ of a patch in the tracking region, we formulate its current foreground weight at frame t as follows:

$$w_{K_p}^t = \varepsilon e^{-\frac{\|K_p - K_t\|^2}{\sigma^2}} \tag{10}$$

which is a radial basis function(RBF) satisfying Gaussian distribution, where ε is the normalized coefficient and K_p denotes the position of a pixel in the tracking region, σ^2 stands for the object occupation in the tracking region. Bigger the σ is, the more attention field the tracking region embodies. Naturally, we compute the current foreground weight of the patch P_i at frame t by the mean of its region of weighted map:

$$\bar{W}_{P_i}^t = \frac{1}{l^2} \sum_{Kp \in P_i} w_{K_p}^t \tag{11}$$

where l is the size of patch. Then the continuous frame are relevant, so the updating is formulated as:

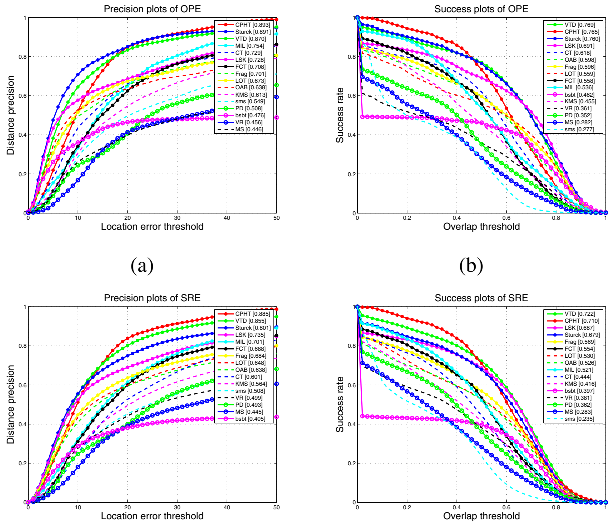
$$W_{P_i}^t = (1-\lambda) W_{P_i}^{t-1} + \lambda \bar{W}_{P_i}^t \tag{12}$$

where λ denotes the learning rate, $W_{P_i}^t$ and $W_{P_i}^{t-1}$ stands for accumulated foreground weight at frame t and t-1 respectively and the weighted map $W_{P_i}^t$ is used momentously in Equation 7. The learning rate balances the current and accumulated weight while the weighted map merits the different weight contribution of distinct patch to PIR.

2) Periodic center rectification: Besides, there is another usage of weighted map. Every τ frame, the deviation of object center may occur and we can eliminate the offset by defining a translation vector $\vec{Z_t}$ at current frame t:

$$\vec{Z_t} = \vec{K_t} - \vec{F_t} \tag{13}$$

It is obvious that the vector defined directs the foreground center to weighted learning center. It handles the problem of center shifting periodically.



(a) (b)

(c) (d)

Fig. 2: Distance precision and overlap success plots over 24 video sequences from benchmark using one-pass evaluation (OPE) and spatial robustness evaluation (SRE). The legend contains the area-under-the-curve score for each tracker. Our proposed tracking methods CPHT performs excellent against others.

III. Experiment

To evaluate the performance of the proposed tracking algorithm in various scenarios, we conduct experiments using 24 representative video sequences (boy, couple, david, david2, deer, dog1, doll, dudek, FaceOcc1, FaceOcc2, fish, fleetFace, Football, Football1, freeman1, freeman3, Ironman, jumping, Mhyang, shaking, Singer1, Soccer, Sylvester, Trellis) from the visual tracker benchmark [1] and compared it with other 15 state-of-the-art tracking methods. All the experiments are running on a PC with Intel i7 3770 CPU (3.4Ghz) and 8G memory. The video sequences from the benchmark contain various challenges such as illumination variation, background clutter, occlusion, abrupt target motion and rotation. The 15 evaluated trackers are structured output tracking with kernels (Struck) [21], visual tracking decomposition (VTD) [22], multiple instance learning tracker (MIL) [23], real-time compressive tracker (CT) [11], local sparse appearance model and k-selection (LSK) [24] method, fast compressive tracking (FCT) [12], fragment tracker (Frag) [5], locally orderless tracker (LOT) [25], online AdaBoost method (OAB) [26], kernel-based tracker (KMS) [27], mean-shift Blob tracking through scale space (SMS) [28], Beyond semi-supervised tracker (BSBT) [29], online selection of discriminative tracking features (VR) [30], peak difference tracker (PD) and the mean shift tracker (MS). All the tracking methods are evaluated by three metrics, (i) distance precision **(DP)**, which shows the percentage of frames whose estimated location is within the given threshold distance of the ground truth; (ii) overlap success rate **(OS)**, which is defined as the percentage of frames where the bounding box overlap surpasses a threshold; and (iii) center location error **(CLE)**, which indicates the average Euclidean distance between the ground-truth and the estimated center location. More details and results can be found in the supplement and the website: http://www.carlib.net/phtracking.html.

A. Experiment Setup

During the experiment, the search radius threshold for drawing positive samples is set to α = 4, generating 45 positive samples. The inner ξ and outer radius threshold β that generates negative samples are set to 8 and 30. The coarse search radius is initialized to r^c = 30 and the corresponding shifting step Δ_c = 4. The radius of fine search r^f = 10 and Δ_f is set to 1. The learning rate λ = 0.25 and the σ is set to 1.25. The period τ for foreground rectification is set to 5.

B. Overall Performance

We present the quantitative comparison results of distance precision (DP) at 25 pixels, overlap success rate (OS) at 0.5, center location errors (CLE) and tracking speed (FPS) in Table I. Among the trackers in the literature, our method achieves the best results with an average DP of 89.3% and the Struck achieves a little lower DP of 89.1%. Conclusively, our algorithm performs the favorable in other metrics with OS of 76.5% and CLE of 12.07 pixels. VTD also performs well with an average OS of 76.9%. While FCT, CT, MIL and MS achieves higher frame rate than others, our algorithm performs pretty good at 20.24 frames per second.

We also conduct the experiment in a conventional way of running trackers throughout test sequences with initialization from the ground truth position at the first frame, which reports the average distance precision under different distance threshold. In the Fig 2, we refer this to one-pass evaluation (**OPE**) and it is evident that our method performs excellent with distance precision as well as success rate. Moreover, to present the advantages of foreground learning, we conduct the spatial robustness evaluation (**SRE**) and our method outperforms others with overlap success rate under different overlap threshold where 12 different initial tracking boxes are set at the first frame.

C. Qualitative Evaluation

To evaluate the performance of our method in quality, we annotate the attributes of each sequence, and construct subsets with different dominant attributes including pose

TABLE I: Comparisons with advanced trackers on the 24 video sequences from benchmark. Our method performs favorably against other methods in distance precision (DP) at 25 pixels, overlap success rate (OS) at 0.5, center location error (CLE). The first and second highest values are highlighted by **red** and **blue** fonts.

	Ours	BSBT	Frag	KMS	LOT	LSK	MIL	MS	OAB	PD	SMS	Struck	VR	VTD	CT	FCT
DP (%)	**89.30**	47.60	70.10	61.30	67.30	72.80	75.40	44.60	63.80	50.80	54.90	**89.10**	45.60	87.00	72.90	70.80
OS (%)	**76.50**	46.20	59.60	45.50	55.90	69.10	53.60	28.20	59.80	35.20	27.70	76.00	36.10	**76.90**	61.80	55.80
CLE (pixel)	**12.07**	155.53	47.62	51.52	43.89	52.27	31.30	72.11	52.00	77.37	63.84	**16.49**	75.29	27.53	47.31	36.92
Speed (FPS)	20.24	7.00	6.30	3.16	0.70	5.50	38.10	31.08	22.40	10.91	19.20	20.20	12.26	5.70	**64.40**	**76.42**

and illumination change, occlusion, background clutters, Fast motion and motion blur, and in-plane and out-of-plane rotation. The comparison of these trackers proves the robustness of our proposed methods as follows.

Background clutters: The surrounding background of the tracking object in the *dudek* sequence changes in illumination and context(Fig. 3(a)). Beyond that, the face undergoes pose change and occlusion (♯210). The Struck, FCT, CT and CPHT algorithms perform well on this sequence. FCT performs well in these sequences as it extracts discriminative scale invariant features for the most correct positive sample (i.e., the target object) online with classifier update for foreground and background separation, so does CPHT. The background of the *Trellis* sequence is clutter and only the Struck, LSK and the proposed CPHT algorithm perform well on this sequence.

Fast motion and motion blur: The object in the *couple* sequence Fig 3(b) moves fast and undergoes an out-of-plane rotation. Only the CPHT and Frag algorithms perform well on this sequence. The images of the *deer* sequence are blurry due to fast motion of the deer. The Struck, FCT, OAB and CPHT algorithms work well in this sequence. As to this attributes, the proposed CPHT algorithm outperforms most of the other methods.

In-plane and out-of-plane rotation: The target object in the *david2* sequence (Fig 3(c)) undergoes the in-plane and out-plane rotation. The in-plane rotation in the *david2* sequence is big, so is the out-of-plane rotation in the *Football1* sequence. The MIL and MKS work well on the *david2* sequence, but fail in the *Football1* sequence, which contains objects undergoing in-plane rotation, out-of-plane rotation and abrupt motion. The VTD, LSK, Struck, FCT and CPHT method perform better than others.

Occlusion: The target object in the *FaceOcc1* sequence in Fig 3(d) undergoes part occlusion. The CPHT, BSBT, Frag, SMS, VR, LOT, OAB, MKS methods work well on this sequence. In the *FaceOcc2* sequence, the target undergoes pose variation and occlusion. Most advanced algorithms have good performance except MS, MKS, PD, VR, SMS. The OAB and MIL methods work well on this sequence as the most discriminative Haar-like features they used for object representation can handle pose variation and part occlusion effectively.

Scale and pose variations: For the *Soccer* in the Fig 3(e), the abrupt scale and pose variation leave a difficult problem. Only a few methods do not lose the target including CPHT, VTD, FCT and VR. In the sequence *Singer1*, tremendous scale variations happen because of the fast motion of video camera. The LOT, Frag, PD and MS lose the target but others perform fine.

Severe illumination changes: For the *Ironman* sequence shown in Fig 3(f), the appearance changes quickly due to illumination and pose variation when the background changes from fireworks (♯18) to a ray of light (♯36) and the direction of the face from left (♯18), to front (♯30 and ♯36) and to right (♯48). Only CPHT algorithm performs well on this sequence. Second-best performance comes from VTD and CT, due to its multiple observation models and compressive features respectively, nevertheless the two trackers are nearly lost from the target. In the *Shaking* sequence shown in Fig 3(f), the VTD, LSK, CPHT and Struck perform well on this sequence with lower tracking errors than other methods.

IV. Conclusion

In this paper, we propose a novel robust tracking algorithm with an appearance model based on compressive hashing feature that preserves the structure of original image space but with small size by means of sparse measurement matrix. In addition, we introduce a visual attention knowledge that the object, namely foreground, should be always located in the center of the weight map, to handle the model drift problem. Experimental results show that the proposed tracking approach performs favorably by comparing with lots of recent state-of-the-art algorithms.

V. Acknowledgment

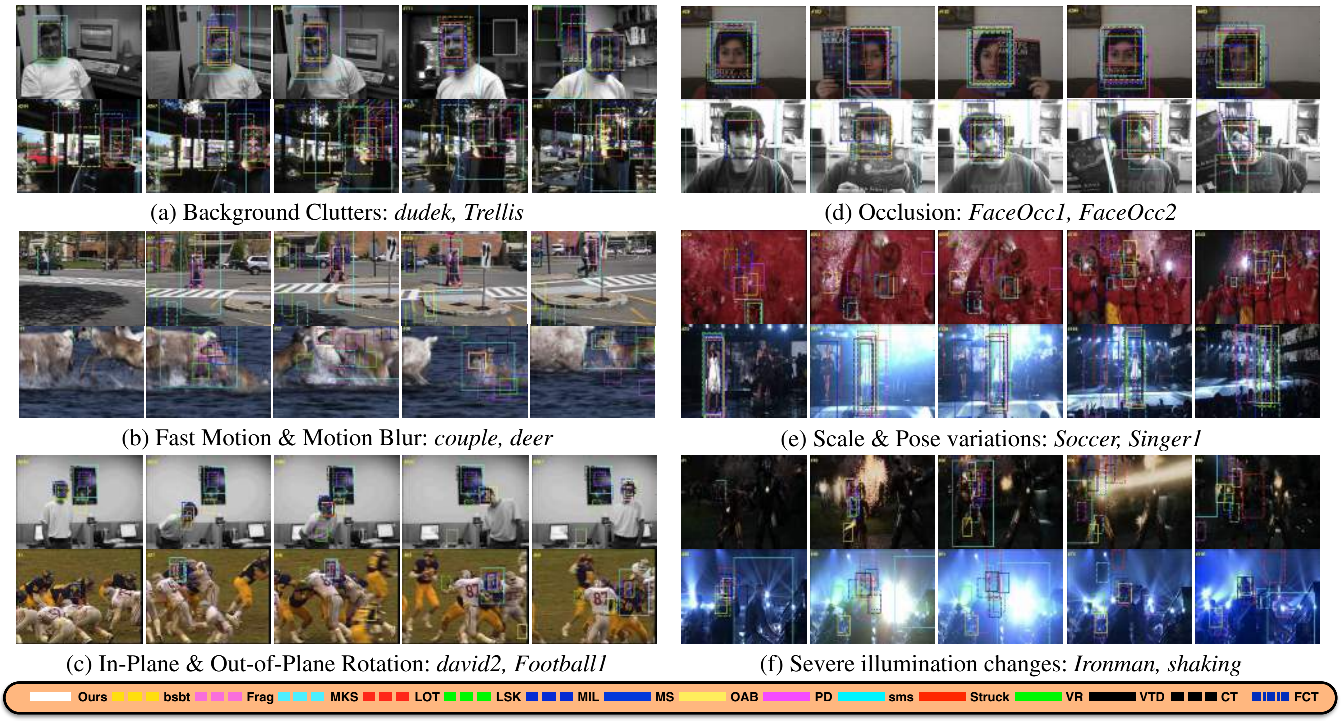
The work described in this paper is supported by National Natural Science Foundation of China (grants ♯ 41401525), the Fundamental Research Funds for the Central Universities (46000-3161001).

References

[1] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *CVPR, 2013 IEEE*. IEEE, 2013, pp. 2411–2418.

[2] S. He, Q. Yang, R. W. Lau, J. Wang, and M.-H. Yang, "Visual tracking via locality sensitive histograms," in *CVPR, 2013 IEEE*. IEEE, 2013, pp. 2427–2434.

[3] J. Zhang, S. Ma, and S. Sclaroff, "Meem: Robust tracking via multiple experts using entropy minimization," in *ECCV 2014*. Springer, 2014, pp. 188–203.



(a) Background Clutters: *dudek, Trellis*

(d) Occlusion: *FaceOcc1, FaceOcc2*

(b) Fast Motion & Motion Blur: *couple, deer*

(e) Scale & Pose variations: *Soccer, Singer1*

(c) In-Plane & Out-of-Plane Rotation: *david2, Football1*

(f) Severe illumination changes: *Ironman, shaking*

Fig. 3: Tracking results of qualitative comparisons. All of the sequences are divided to 6 parts by attributes.

[4] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1619–1632, 2011.

[5] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *CVPR, 2006 IEEE*, vol. 1. IEEE, 2006, pp. 798–805.

[6] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *CVPR, 2012 IEEE*. IEEE, 2012, pp. 1830–1837.

[7] M. Danelljan, F. S. Khan, M. Felsberg, and J. v. d. Weijer, "Adaptive color attributes for real-time visual tracking," in *CVPR, 2014, IEEE.* IEEE, 2014, pp. 1090–1097.

[8] C. Ma and C. Liu, "Two dimensional hashing for visual tracking ," *Computer Vision and Image Understanding*, pp. 83–94, 2015.

[9] M. Fei, J. Li, and H. Liu, "Visual tracking based on improved foreground detection and perceptual hashing," *Neurocomputing*, vol. 152, pp. 413–428, 2015.

[10] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 34, no. 4, pp. 435 – 443, 2004.

[11] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *ECCV 2012*. Springer, 2012, pp. 864–877.

[12] K. Zhang, L. Zhang, and M. H. Yang, "Fast compressive tracking," *Pattern Analysis and Machine Intelligence IEEE Transactions on*, vol. 36, no. 10, pp. 1–1, 2014.

[13] V. Sulić, J. Pers, M. Kristan, and S. Kovacic, "Dimensionality reduction for distributed vision systems using random projection," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, Aug 2010, pp. 380–383.

[14] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *Signal Processing, IEEE Transactions on*, vol. 56, no. 6, pp. 2346–2356, June 2008.

[15] E. Candes and T. Tao, "Decoding by linear programming," *Information Theory, IEEE Transactions on*, vol. 51, no. 12, pp. 4203–4215, Dec 2005.

[16] ——, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *Information Theory, IEEE Transactions on*, vol. 52, no. 12, pp. 5406–5425, Dec 2006.

[17] Y. Hua, B. Xiao, D. Feng, and B. Yu, "Bounded lsh for similarity search in peer-to-peer file systems," in *Parallel Processing, 2008. ICPP '08. 37th International Conference on*, Sept 2008, pp. 644–651.

[18] A. Torralba, "Contextual priming for object detection," in *IJCV*, 2003, p. 2003.

[19] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M. H. Yang, *Fast Visual Tracking via Dense Spatio-temporal Context Learning*. Springer International Publishing, 2014.

[20] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2, Aug 2004, pp. 28–31 Vol.2.

[21] S. Hare, A. Saffari, and P. H. Torr, "Struck: Structured output tracking with kernels," in *ICCV, 2011 IEEE*. IEEE, 2011, pp. 263–270.

[22] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *CVPR, 2010 IEEE*. IEEE, 2010, pp. 1269–1276.

[23] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *CVPR 2009. IEEE*. IEEE, 2009, pp. 983–990.

[24] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and k-selection," in *CVPR, 2011 IEEE*. IEEE, 2011, pp. 1313–1320.

[25] S. Avidan, D. Levi, A. Bar-Hillel, and S. Oron, "Locally orderless tracking," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1940–1947.

[26] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via online boosting." *Bmvc*, pp. 47–56, 2006.

[27] D. Comaniciu, V. Ramesh, P. Meer, S. Member, and S. Member, "Kernel-based object tracking," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003, pp. 564–577.

[28] R. T. Collins, "Mean-shift blob tracking through scale space," *In: Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR*, vol. 2, pp. II – 234–40, 2003.

[29] S. Stalder, H. Grabner, and L. Van Gool, "Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, 2009, pp. 1409–1416.

[30] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, 2005.