STRUCTURE-REGULARIZED COMPRESSIVE TRACKING

Qing Guo, Wei Feng, Ce Zhou, Bin Wu*

School of Computer Science and Technology, Tianjin University, Tianjin, China
Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University, Tianjin, China
{tsingqguo, wfeng, zhouce,binwu}@tju.edu.cn

ABSTRACT

Compressive random projection is a powerful appearance model to derive effective Haar-like features from non-rotated 4D rectangles, which can support fast and reliable object tracking. In this paper, we show that such successful compressive tracking scheme can be further significantly improved by structural regularization. Specifically, we propose two effective structural regularizations. First, we find that, guided by superpixels, compressive random projection can always generate more discriminative features by sufficiently capturing the rich local structure information of images. Second, we present fast directional integration to enable low-cost extraction of feasible Haar-like features from arbitrarily rotated 5D rectangles to realize more accurate object localization. We compare the proposed structure-regularized compressive tracker with a number of state-of-the-art methods. Extensive experiments on challenging benchmark dataset validate the superior performance and comparable real-time speed of the proposed approach.

***Index Terms*—** Compressive tracking, structural regularization, superpixel guidance, fast directional integration

1. INTRODUCTION

Object tracking is a fundamental problem in computer vision. Real-world tracking tasks can be very challenging, especially when illumination variances, background clutter and complex global/local motions are involved [1–5]. In recent years, a lot of work has revealed the indispensable contribution of effective appearance models to achieving reliable tracking accuracy and efficiency [4, 6–13].

According to the particular type of appearance models they used, tracking algorithms can be roughly classified into generative and discriminative models. Generative models, such as subspace models and holistic templates, are good at real-time online tracking. But, they usually cannot handle rapid motion well and do not take good advantage of background information. In contrast, discriminative models formulate object tracking at each frame as a binary classification problem to localize the object rectangle (i.e. bounding box) from spatially-temporally-sampled candidates via local search [3, 9, 13–18]. Since they properly use both object and background information of previous frames as positive/negative samples to dynamically update the classifier, most recent trackers adopt the discriminative model.

Fast compressive tracking (FCT) [13] is a recent notable state-of-the-art discriminative tracker, which can simultaneously achieve promising tracking accuracy and real-time speed. The success of FCT comes from its powerful appearance model, compressive random projection [19, 20], by randomly constructing sparse measurement matrix $\mathbf{R}$ to project the high-dimensional multiscale image features into a low-dimensional compressed subspace with guaranteed discriminative power. Besides, the sparsity of $\mathbf{R}$ and the restricted isometry property make FCT both efficient and robust.

In this paper, we show that such successful compressive tracking scheme can be further improved significantly using *structural regularization*. In particular, we propose two effective structural regularizations. We compare the proposed structure-regularized compressive tracker with a number of state-of-the-art tracking methods. Extensive experiments on benchmark dataset validate the superior performance and comparable real-time speed of our approach.

Specifically, our first contribution is *superpixel-guided random projection* that can always generate more discriminative features by sufficiently capturing the rich local structure information of images. The advantage of *superpixel guidance* mainly lies in the capability of reliable superpixel segmentation [21, 22] of preserving most important image structure information, and are partially inspired by the recent success of part-based appearance features in object detection [23].

Our second contribution is *fast directional integration* that enables low-cost extraction of feasible Haar-like features from arbitrarily rotated 5D rectangles to realize more accurate object tracking and localization. Note, each entry of compressed feature of FCT is essentially the Haar-like feature [13] derived from spatially-scattered non-rotated 4D rectangles (x, y, w, h) with (x, y) being the position and (w, h) indicating the size of rectangles. Although Haar-like feature extracted from 4D rectangles is commonly used in object de-

* is corresponding author. Tel: (+86)-22-27406538.

This work is supported by the National Science and Technology Support Project (2013BAK01B01) and NSFC (61572354).

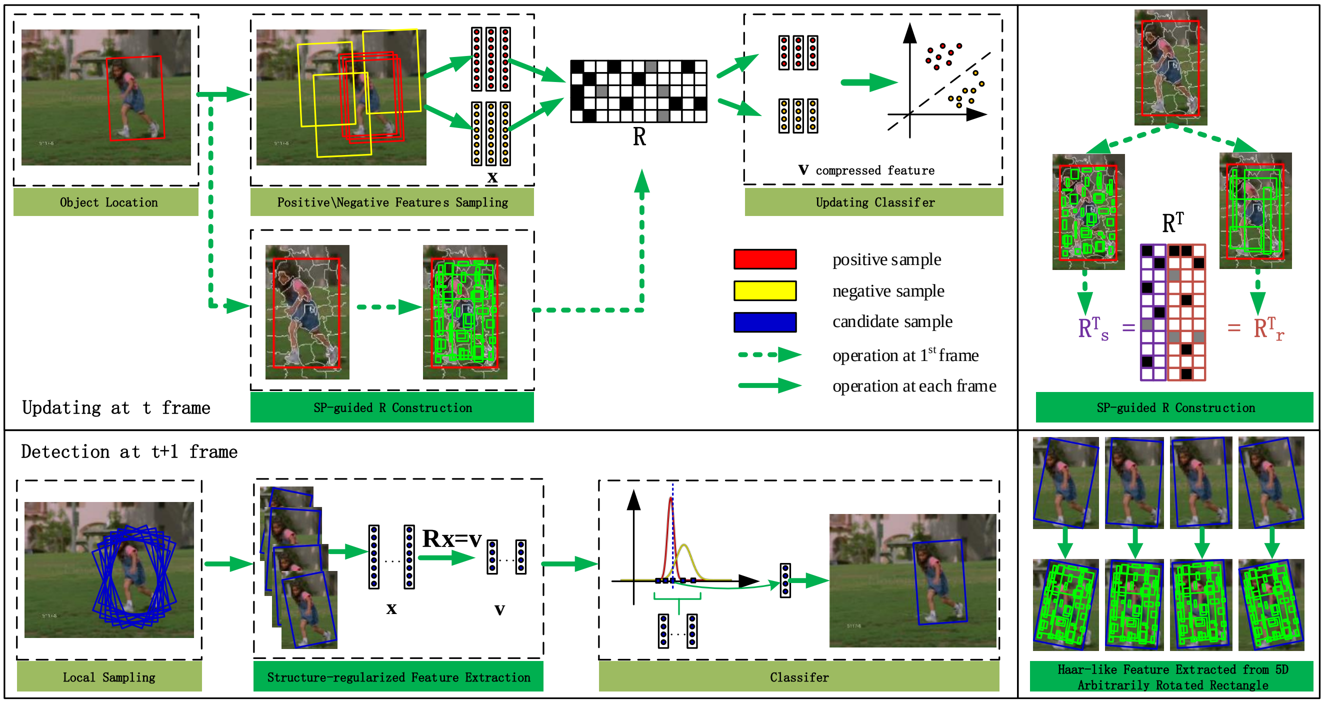

Fig. 1. Algorithmic flow of the proposed structure-regularized compressive tracking. Top and down left figures are the process of updating classifier and tracking at each frame. Top right shows the SP-guided measurement matrix construction at first frame. Down right shows that we can calculate rotated Haar-like feature efficiently. The right two figures are the main contributions of our work.

tection and tracking due to its simplicity and efficiency benefited from integral image [4, 13], it is not the best choice for accurate object localization, because real objects usually have non-rectilinear shapes. Hence, a tighter rotated bounding-box representation may certainly improve localization accuracy [24]. However, despite of several preliminary trials on fast integration over general polygonal shapes [24, 25], how to efficiently realize fast Haar-like feature extraction for arbitrarily rotated rectangles in a low-cost manner is still an open problem. The proposed method indeed provides a feasible approximate solution to low-cost directional integration of arbitrarily rotated 5D rectangles (x, y, w, h, θ), parameterized by position (x, y), size (w, h) and rotation θ.

2. PRELIMINARY

Random projection and compressive sensing. For a discriminative model, we want the extracted feature is in a low-dimensional space with strong discriminative power. With random projection [19], a lower-dimensional feature space $\mathbf{v} \in \mathbf{R}^n$ can be obtained by projecting the high-dimensional feature space $\mathbf{x} \in \mathbb{R}^m$ via a random measurement matrix $\mathbf{R} \in \mathbb{R}^{n\times m}$, i.e. $\mathbf{v} = \mathbf{R}\mathbf{x}$ with $n \ll m$. The dimension of $\mathbf{v}$ is low enough to get an efficient method. In compressive sensing theory, when $\mathbf{x}$ is K-sparse, the restricted isometry property is satisfied, which leads the $\mathbf{v}$ is approximately discriminative as the original feature $\mathbf{x}$. These two theoretical supports make it possible to build a robust but real-time tracking method. However, the fully randomized measurement matrix are often not feasible for real-world applications due to the cost of multiplying arbitrary matrices with signal vectors of high dimension. Then, the theory for constructing measurement matrix that is not fully random and contains structure information is improved [26].

Fast compressive tracking. Fast compressive tracking was proposed based on the above theories. $\mathbf{x} \in \mathbb{R}^m$ represents the multiscale image representation each entry of which is the average intensity of a rectangle [13]. $\mathbf{R}$ is created with each entry being 1, -1 and 0 with probability $\frac{1}{2\rho}$, $\frac{1}{2\rho}$ and $1 - \frac{1}{\rho}$ at first frame, where $\rho = o(m) = \frac{m}{(a \log 10m)}$, $a = 0.4$, which makes $\mathbf{R}$ very sparse. Each entry of $\mathbf{v}$ is essentially the Haar-like feature with randomly selected non-rotated 4D rectangles, which makes the FCT fail to calculate the compressed feature of rotated bounding-box.

3. OUR APPROACH

The proposed tracking framework is a discriminative model and mainly contains two parts: the updating of classifier and the detection by classifier, as shown in the left of Fig. 1. In the updating process at frame t, the positive/negative samples are selected according to the given object location. Then, the multiscale representation is used to represent each sample as a vector $\mathbf{x}$ [13]. $\mathbf{x}$ is projected to a compressed vector $\mathbf{v}$

through a measurement matrix $\mathbf{R}$ which is constructed by the guidance of SP at first frame and introduced in section 3.1. Finally, the vectors of positive/negative samples are used to train a naive Bayes classifier [13].

In the detection process at frame $t+1$, the dense sampling search is performed in translation and rotation spaces respectively and results in samples with 5 parameters, i.e. (x, y, w, h, θ). The compressed features $\mathbf{v}$ of these samples are calculated from $\mathbf{x}$ with the same $\mathbf{R}$. Then, the trained classifier assigns each sample a probability belonging to the object. The sample with maximum probability is selected as the object location at frame $t+1$. The dense sampling search usually ignores the rotation space with Haar-like feature for lacking an efficient method to calculate the integral of rotated rectangle. In section 3.2, we propose a method to fast calculating the integral of rotated rectangle with traditional integral image and quadtree segmentation result.

3.1. Superpixel-guided random projection

Instead of fully randomly creating $\mathbf{R}$, we propose to split $\mathbf{R} \in \mathbb{R}^{n\times m}$ into two parts $\mathbf{R} = [\mathbf{R_s^T}, \mathbf{R_r^T}]^T$ as shown in Fig. 1, where $\mathbf{R_s} \in \mathbb{R}^{n_1\times m}$ is constructed according to the superpixel structure; $\mathbf{R_r} \in \mathbb{R}^{n_2\times m}$ is constructed randomly as introduced in Section 2 with $n = n_1 + n_2$. In detail, at first frame, the bounding box of object is given. We segment the region with SLIC method [27] and obtain a set of superpixel S. According to the multiscale image feature used in FCT, we define s_j as a rectangle corresponding to the entry r_{*j} of $\mathbf{R}$. Then, we define $s_j \in S$, if all the pixels of s_j are included in a superpixel of S. Therefore, we create $\mathbf{R_s} = \{r_{ij}\}$ by

$$r_{ij} = \begin{cases} \sqrt{\rho}, & \text{with probability } \frac{1}{4\rho}, \text{ if } s_j \in S \\ 0, & \text{with probability } 1 - \frac{1}{2\rho}, \\ -\sqrt{\rho}, & \text{with probability } \frac{1}{4\rho}, \text{ if } s_j \in S, \end{cases} \tag{1}$$

where $\rho = o(m) = \frac{m}{a \log 10m}$ and $a = 0.4$. $\mathbf{R_s}$ is much sparser than $\mathbf{R_r}$ for only about $\frac{1}{2\rho}$ of the entries being nonzero comparing to $\frac{1}{\rho}$ in $\mathbf{R_r}$. Furthermore, SPs cluster similar and adjacent pixels, being seem as parts of the object to extract class independent proposals [28]. Therefore, the feature calculating from SPs should be more discriminative than randomly generated. However, the fully SP-based construction is local, which may affect the performance. The $\mathbf{R_r}$ is data-independent and easily capture more global information. Therefore, $\mathbf{R_s}$ and $\mathbf{R_r}$ are combined to construct the final measurement matrix. The experiment in section 5 shows the performance of different ratio between n_1 and n_2.

3.2. Fast directional integration

We propose a method to fast calculate the integral of arbitrary rotated rectangle with the traditional integral image and quadtree segmentation, which helps to obtain a tighter rotated boundingbox representation with the compressed features.

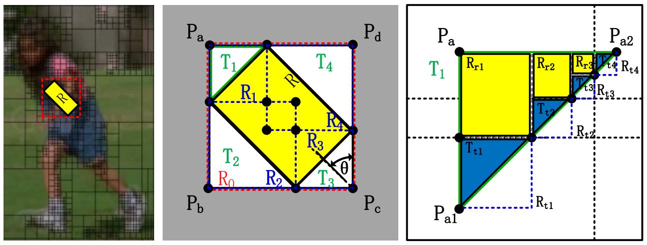

Fig. 2. Fast directional integration. Left: quadtree segmentation result and the integral of rotated rectangle R (yellow) we want to calculate; Middle: calculating integral of R in homogeneous region; Right: calculating the integral of right triangle T_1 with quadtree segmentation; black dash lines are quadtree segmentation boundaries.

Given an image $\mathrm{I} \in \mathbb{R}^{M\times N}$ with $y \in [0, M)$ and $x \in [0, N)$, $\mathrm{I}(x, y)$ is the intensity at (x, y). The integral of a rectangle, e.g. R_0, is the sum of intensity of the pixels within it and defined as $\mathrm{A}(\mathrm{R}_0)$. The integral image of I is calculated and defined as H with $\mathrm{H}(x, y) = \sum_{x_t=0}^{x}\sum_{y_t=0}^{y}\mathrm{I}(x_t, y_t)$. Then, the integral of a non-rotated rectangle, e.g. R_0, can be computed as

$$\mathrm{A}(\mathrm{R}_0) = \mathrm{H}(\mathrm{P_a}) + \mathrm{H}(\mathrm{P_c}) - \mathrm{H}(\mathrm{P_b}) - \mathrm{H}(\mathrm{P_d}). \tag{2}$$

As shown in Fig. 2 (left), our goal is to calculate the integral of a rotated rectangle, i.e. $\mathrm{A}(\mathrm{R})$. Obviously, $\mathrm{A}(\mathrm{R})$ can be easily obtained via calculating the integral of R_0 and minus the integral of four right triangles, as defined in Eq. (3) and shown in Fig. 2 (middle). If the image is homogeneous, the integral of right triangle, i.e. $\mathrm{A}(\mathrm{T}_k)$ is equal to the half of the integral of rectangle R_k, i.e. $\mathrm{A}(\mathrm{R}_k)$. However, a real image is always inhomogeneous, which makes the simple operation to calculate the triangle integration failed. Nevertheless, it is reasonable to assume the local homogeneous of a real image, with which we can calculate the integral approximatively. That is, the problem is transformed to calculate the integral of right triangles, i.e. $\mathrm{A}(\mathrm{T}_k)$, as

$$\mathrm{A}(\mathrm{R}) = \mathrm{A}(\mathrm{R}_0) - \sum_{k=1}^{4}\mathrm{A}(\mathrm{T}_k). \tag{3}$$

We use the quadtree segmentation to subdivide an image into blocks and get $\mathrm{Q} \in \mathbb{R}^{M\times N}$ shown in Fig. 2 (left). Since quadtree splits the images into blocks according to the intensity variance, the block is approximately homogeneous. For a upper left right triangle, i.e. T_1 in Fig. 2 (right), the quadtree boundaries separate it into several right triangles (blue regions) and non-rotated rectangles (yellow regions). The integral of these rectangles can be calculated through Eq. (2); the integral of a right triangle T_{ti} in a quadtree block can be approximatively calculated by halving the integral of the rectangle R_{ti} for the homogeneity of the block with $i = 1, 2, 3, 4$. As a result, T_1 can be approximated by combining the integral of rectangles and right triangles as defined in Eq. (4). We can obtain the integration of $\mathrm{T}_{2,3,4}$ in the same way. Then, the integral of arbitrary rotated rectangle can be calculated through Eq. (3) with the complexity being less than

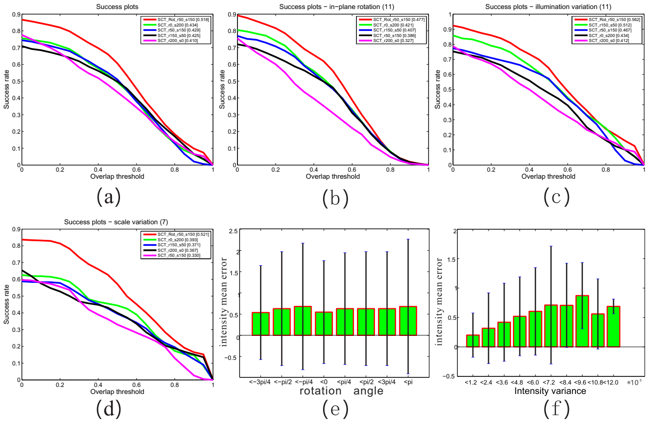

Fig. 3. (a)-(d) show the average success plots of the 5 trackers for all 20 videos (a), 11 videos with in-plane rotation (b), 11 videos with illumination variation (c) and 7 videos with scale variation (d), respectively. Note, the 4 trackers named as SCT_rn_2_sn_1[AUC] denotes the SCT tracker using measurement matrix $\mathbf{R}$ with n_1 items generated by superpixel guidance and n_2 resulted from random projection. AUC is the area under curve. Thus, SCT_r200_s0 is the original FCT tracker [13]. SCT_Rot represents the SCT tracker considering rotated rectangles. (e), (f): Intensity mean error distribution of 100 thousand rectangles w.r.t. increasing rotation angles and intensity variance, respectively.

$O(2(w\cos\theta + h\sin\theta))$ with w , h and θ being the width, height and rotation degree of the rectangle, respectively. Although the complexity is related to the size of the rectangle, arbitrary angle of rotation can be calculated via this way with pre-computed integral image and quadtree segmentation. Other methods for rotated rectangle integration should build integral image for each angle, which is not suitable in tracking to sample rectangles with various angles [24,25].

$$\begin{aligned} \mathrm{A}(\mathrm{T}_1) &= \sum_{i=1}^{3}\mathrm{A}(\mathrm{R}_{\mathrm{r}i}) + \sum_{i=1}^{4}\mathrm{A}(\mathrm{T}_{\mathrm{t}i}) \\ &= \sum_{i=1}^{3}\mathrm{A}(\mathrm{R}_{\mathrm{r}i}) + \sum_{i=1}^{4}\frac{1}{2}\mathrm{A}(\mathrm{R}_{\mathrm{t}i}). \end{aligned} \tag{4}$$

4. EXPERIMENTS

We demonstrate the proposed method in three parts. First, we demonstrate that SP-guidance $\mathbf{R}$ construction and rotation transform do improve the performance of FCT method. We also analyse the error source of directional integration, showing that proposed method can calculate the integral of rotated rectangle efficiently with tolerable error. Finally, we compare the proposed tracking framework with 15 trackers.

4.1. Setup

We analysis and evaluate the performance of our tracker with 20 challenge sequences which contains 10 interference factors, e.g. illumination and scale variation, in/out-plane rotation, deformation, occlusion, motion blur, fast motion, background clutter and motion blur. Then, we can obtain 10 subset sequences according to different factors as done in [2, 4] and test the trackers on the whole 20 sequences and 10 subset sequences, respectively. We name our method SCT and use 15 trackers including top 14 trackers in [2] and the Fast Compressive Tracker (FCT) [13] as the baselines to evaluate the performance. The top 14 trackers are, Struck [18], sparsity-based collaborative model (SCM) [7], adaptive structural local sparse appearance model based (ASLA) tracker [8], circulant structure (CSK) tracker [9], $\ell_1$tracker (L1T) [10], online AdaBoost method (OAB) [15], visual tracking decomposition (VTD) [29], tracking by sampling trackers (VTS) [30], distribution field tracking (DFT) [11], local sparse and K-selection (LSK) [31], online multiple instance learning (MIL) tracker [16], tracking by detection (TLD) [17], multi-task tracker (MTT) [32] and fragment tracker (Frag) [33].

We use two metrics for evaluating the performance of above methods: bounding box overlap score S_{ovl} and center location error L_{err}. With the ground truth of bounding box B_{gt} and the tracked bounding box B_{t}, we have: $S_{\text{ovl}} = \frac{|B_{\text{gt}} \cap B_{\text{t}}|}{|B_{\text{gt}} \cup B_{\text{t}}|}$; $L_{\text{err}} = \mathrm{ED}(\mathrm{C}(B_{gt}), \mathrm{C}(B_t))$, where $\cap$ and $\cup$ represent the intersection and union of B_{gt} and B_t; $|\cdot|$ is the number of pixels in the bounding box; $\mathrm{C}(\cdot)$ is center position of a bounding box; $\mathrm{ED}(\cdot)$ is the Euclidean distance. For each metric, we can define a threshold, t_{ovl} or t_{err}, to define the success of a tracker in each frame. Then, we obtain the success plot for S_{ovl} and precision plot for L_{err} [2]. The area under curve (AUC) of each plot is then calculated.

4.2. Improvement by structural regularization

As introduced in section 3.1, the measurement matrix $\mathbf{R} \in \mathbb{R}^{n\times m}$ contains two parts: $\mathbf{R_s} \in \mathbb{R}^{n_1\times m}$ from SP-guidance, and $\mathbf{R_r} \in \mathbb{R}^{n_2\times m}$ from random selecting. Based on the original FCT ($n_1 = 0$), we fix $n = 200$ and change the ratio between $n_1 = \{0, 50, 150, 200\}$ and $n_2 = \{200, 150, 50, 0\}$ to track the 20 sequences to show the advantage of the SP guidance construction of $\mathbf{R}$ and the influence of the ratio between n_1 and n_2. The results are shown in Fig. 3 (a-d). The performance of FCT ($n_1 = 0, n_2 = 200$) does be improved by introducing SP-guidance with $n_1 > 0$. When the $\mathbf{R}$ is fully constructed through SP-guidance, i.e. $n_1 = 200, n_2 = 0$, we get the best results. Further, by considering rotation transform with the proposed fast directional integration, SCT outperform the original FCT with great performance improvement in all videos, especially for the rotation, illumination variation and scare variation videos.

4.3. Error analysis of directional integration

The error source of the directional integration comes from the calculating integral of right triangle in a quadtree block. To

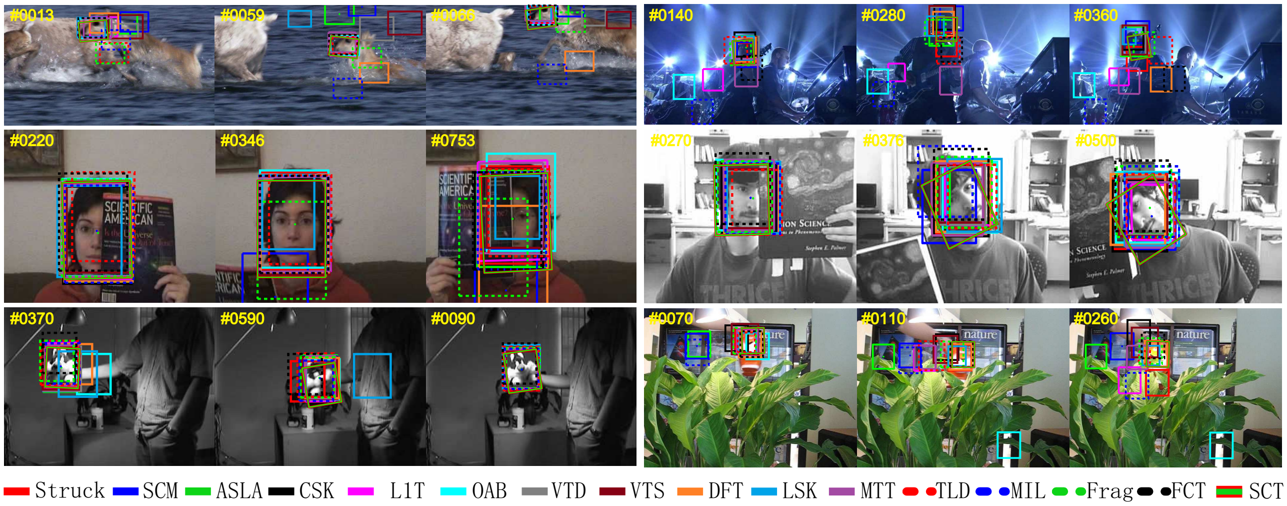

Fig. 4. Some real tracking results of all compared methods.

Table 1. Average AUC rank of 16 methods on 10 subset sequences. Rank1 and Rank2 are the rank sorted by precision plot AUC and success plot AUC, respectively. Smaller rank is better.

	Struck	SCM	ASLA	CSK	L1T	OAB	VTD	VTS	DFT	LSK	MTT	TLD	MIL	Frag	FCT	SCT
Rank1	3.82	9.64	10.91	5.55	10.73	10.73	5.27	5.82	4.27	9.18	11	7.27	7.18	10.27	9.27	**2.73**
Rank2	3.73	9.09	11.09	5.09	10.45	11.09	5.81	6.72	3.45	8.81	11.27	7.82	7.09	10.55	9	**2.54**

analyse the error of the method, we randomly select 50 images from the testing sequences. Then, we construct a 100 thousand rectangle set by selecting 2000 rectangles with random position, size and rotation from each image. For each rectangle, we calculate its integral and intensity mean by the proposed method and the summation method being the ground truth, respectively. The intensity of a pixel is between 0 and 255. The error is defined as the difference between the directional integration result and the ground truth and is analyzed according to the rotation angles and intensity variance of the pixels in each rectangle. As shown in Fig. 3 (e),(f), we have following observations: 1) mean error of all rectangles is less than one; 2) different rotation angles have similar error, inferring the method can handle arbitrary rotated rectangles; 3) the mean error enlarges with the increasing of rectangle intensity variance, being less than one. In our experiments, the error is small enough to keep the tracking working and capturing the rotation transform, which makes the tracking framework more precise, as shown in Fig. 3.

4.4. Comparative results

We compare our algorithm with 15 methods and fix our parameters as follows: $n_1 = 150$, $n_2 = 50$; degree search range is set as $[-\frac{\pi}{64}, \frac{\pi}{64}]$ with search step $\frac{\pi}{128}$; We extract the SPs with SLIC at first frame by setting 600 pixels for each SP with spatial compact parameter as 10; The split threshold used in quadtree segmentation is set as 0.05. Other parameter are the same with the original FCT. The tracking results are shown in Fig. 4, 5 and Table 1. In Fig. 4, with the fast directional integration, our method is able to capture the object rotation transform with a tight rotated rectangle, i.e. the deer,shaking, faceocc2 and sylv sequences, which should help to improve the performance. It also shows that our method is robust to the occlusion and illumination changes. The Fig. 5 shows that our method achieves better performance than the state-of-the-art methods on 20 challenge sequences by introducing the structure regularization into the FCT. It also shows that our method achieve good performances in rotation (best) and

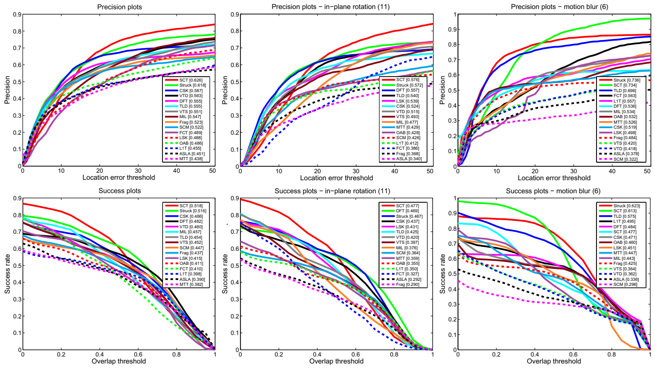

Fig. 5. Average precision plots (top) and average success plots (down) of 16 methods for 20 benchmark video sequences, 11 rotation sequences and 6 motion blur sequences, respectively. The legends are sorted descendingly according to the AUC value.

motion blur (second best) videos. For each subset sequence, we can obtain a rank for each tracker. Then, we can obtain the average rank on 10 subset sequences as shown in Table 1. SCT has the lowest rank, which demonstrates that SCT does well to handle various challenge factors.

5. CONCLUSION

In this paper, we have proposed two effective structural regularizations to the successful compressive tracking scheme. Our major contributions are two-fold. First, we find that, superpixel-guided random projection can always produce more discriminative tracking features by capturing the rich local structure information of images. Second, we present fast directional integration to enable low-cost extraction of feasible Haar-like features from arbitrarily rotated 5D rectangles to realize more accurate object tracking and localization. Extensive experiments on challenging benchmark dataset validate the superior performance and comparable real-time speed of structure-regularized compressive tracker over state-of-the-art methods. In the future, we plan to further explore how to use structural regularization to realize fast and accurate video object segmentation.

6. REFERENCES

[1] S. Salti, A. Cavallaro, and L. Di Stefano, “Adaptive appearance modeling for video tracking: Survey and evaluation,” *IEEE TIP*, vol. 21, no. 10, pp. 4334–4348, 2012.

[2] Y. Wu, J. Lim, and M.-H. Yang, “Online object tracking: A benchmark,” in *CVPR*, 2013.

[3] J. Xiao, R. Stolkin, and A. Leonardis, “Single target tracking using adaptive clustered decision trees and dynamic multi-level appearance models,” in *CVPR*, 2015.

[4] T. Zhang, S. Liu, C. Xu, S. Yan, and B. Ghanem, “Structural sparse tracking,” in *CVPR*, 2015.

[5] W. Feng, F.-P. Tian, Q. Zhang, N. Zhang, L. Wan, and J. Sun, “Fine-grained change detection of misaligned scenes with varied illuminations,” in *ICCV*, 2015.

[6] S. Wang, H. Lu, and F. Yang, “Superpixel tracking,” in *ICCV*, 2011.

[7] W. Zhong, H. Lu, and M.-H. Yang, “Robust object tracking via sparsity-based collaborative model,” in *CVPR*, 2012.

[8] X. Jia, H. Lu, and M.-H. Yang, “Visual tracking via adaptive structural local sparse appearance model,” in *CVPR*, 2012.

[9] F. Henriques, R. Caseiro, P. Martins, and J. Batista, “Exploiting the circulant structure of tracking-by-detection with kernels,” in *ECCV*, 2012.

[10] C. Bao, Y. Wu, H. Ling, and H. Ji, “Real time robust l1 tracker using accelerated proximal gradient approach,” in *CVPR*, 2012.

[11] L. Sevilla-Lara and E. Learned-Miller, “Distribution fields for tracking,” in *CVPR*, 2012.

[12] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel, “Part-based visual tracking with online latent structural learning,” in *CVPR*, 2013.

[13] K. Zhang, L. Zhang, and M. Yang, “Fast compressive tracking,” *IEEE TPAMI*, vol. 36, no. 10, pp. 2002–2015, 2014.

[14] S. Avidan, “Ensemble tracking,” in *CVPR*, 2005.

[15] H. Grabner, M. Grabner, and H. Bischof, “Real-time tracking via on-line boosting,” in *BMVC*, 2006.

[16] B. Babenko, M.-H. Yang, and S. Belongie, “Visual tracking with online multiple instance learning,” in *CVPR*, 2009.

[17] Z. Kalal, J. Matas, and K. Mikolajczyk, “Pn learning: Bootstrapping binary classifiers by structural constraints,” in *CVPR*, 2010.

[18] S. Hare, A. Saffari, and P. H. S. Torr, “Struck: Structured output tracking with kernels,” in *ICCV*, 2011.

[19] E. Candes and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?,” *IEEE TIT*, vol. 52, no. 12, pp. 5406–5425, 2006.

[20] M. F. Duarte and Y. C. Eldar, “Compressed sensing,” *IEEE TIT*, vol. 52, no. 4, pp. 1289–1306, 2006.

[21] L. Li, W. Feng, L. Wan, and J. Zhang, “Maximum cohesive grid of superpixels for fast object localization,” in *CVPR*, 2013.

[22] W. Feng, J. Jia, and Z.-Q. Liu, “Self-validated labeling of Markov random fields for image segmentation,” *IEEE TPAMI*, vol. 32, no. 10, pp. 1871–1887, 2010.

[23] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part based models,” *IEEE TPAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.

[24] M.-T. Pham, Y. Gao, V.-D. D. Hoang, and T.-J. Cham, “Fast polygonal integration and its application in extending haar-like features to improve object detection,” in *CVPR*, 2010.

[25] A. L. C. Barczak, M. J. Johnson, and C. H. Messom, “Real-time computation of haar-like features at generic angles for detection algorithms,” *Res. Lett. Inf. Math. Sci.*, vol. 9, no. 1, pp. 98–111, 2006.

[26] M. F. Duarte and Y. C. Eldar, “Structured compressed sensing: From theory to applications,” *IEEE TSP*, vol. 59, no. 9, pp. 4053–4085, 2011.

[27] R. Radhakrishna, A. Shaji, K. Smith, A. Licchi, P. Fua, and S. Susstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” *IEEE TPAMI*, vol. 34, no. 11, pp. 2274–2282, 2012.

[28] K. Van de Sande, J. Uijlings, T. Gevers, and A. W. M. Smeulders, “Segmentation as selective search for object recognition,” in *ICCV*, 2011.

[29] J. Kwon and K. M. Lee, “Visual tracking decomposition,” in *CVPR*, 2010.

[30] J. Kwon and K. M. Lee, “Tracking by sampling trackers,” in *ICCV*, 2011.

[31] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, “Robust tracking using local sparse appearance model and k-selection,” in *CVPR*, 2011.

[32] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *CVPR*, 2012.

[33] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *CVPR*, 2006.