Real-time Tracking via Deformable Structure Regression Learning

Xian Yang*, Quan Xiao†, Shoujue Wang† and Peizhong Liu‡
*Lab of Artificial Neural Networks, Institute of Semiconductors, CAS. China
†Lab of High Dimensional Biomimetic Informatics and its Applications, SINANO, CAS. China
‡College of Engineering, Huaqiao University, Fujian, China
xyang2011@sinano.ac.cn, qxiao2012@sinano.ac.cn, sjwang2008@sinano.ac.cn, pzliu@hqu.edu.cn

***Abstract*—Visual object tracking is a challenging task because designing an effective and efficient appearance model is difficult. Current online tracking algorithms treat tracking as a classification task and use labeled samples to update appearance model. However, it is not clear to evaluate instance confidence belong to the object. In this paper, we propose a simple and efficient tracking algorithm with a deformable structure appearance. In our method, model updates with continuous labeled samples which are dense sampling. In order to improve the accuracy, we introduce a couple-layer regression model which prevents negative background from impacting on the model learning rather than traditional classification. The proposed DSR tracker runs in real-time and performs favorably against state-of-the-art trackers on various challenging sequences.**

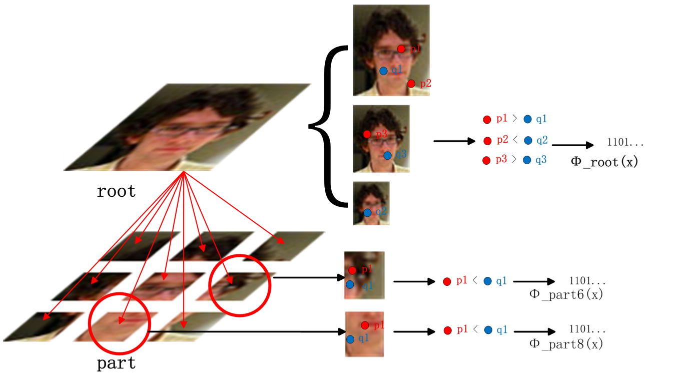


Fig. 1. Illustration of 2-layer deformable structure appearance model. The top part is the root layer representation, and the bottom part shows the local layer representation.

I. Introduction

Visual object tracking remains a challenging research topic in computer vision during the past decades [1] caused by many factors such as appearance variations, real-time processing requirements and low-quality camera sensors. It is difficult to deal with due to 1) prior target information is little and usually got from the first frame; 2) the large appearance changes of target object and the background; 3) the boundary between the object of interest and the background is ambiguous.

A typical tracking system consists of three main components: appearance modeling, motion estimation and search strategy. How to design a robust appearance model is the crucial point in tracking system. This is generally separated into two parts: visual representation and learner modeling [1]. Visual representation focuses on how to construct robust features for object expression. Learner modeling concentrates on building models which can adapt online to object appearance change. In our tracker, the appearance model is implemented by a 2-layer deformable structure model trained with Brief feature as object expression. Learning model combines with weak regression ferns as appearance and shape model. Search strategy is fast dense sampling search.

Many proposed methods treat the tracking problem as a classification task [2]–[6]. These methods need a set of labeled training instances to determine the decision boundary for separating the target object. However, it is not clear to evaluate instance label. In this paper, we argue that classifier model is suboptimal due to the different objective between classification and the object location estimation. Moreover, it is unreliable that a large number of training samples need to be selected and labeled. To avoid it, structured output prediction [7] has been applied and got an outstand performance in recent evaluation [8]. Researchers also proposed dense sampling to study generative learning model [9], [10]. These methods didn't limit samples to binary labels, blurred the line between classification and regression, and work efficiently. It shows that dense sampling with continuous outputs for regression model is a promising direction. Inspired by the above, we propose a deformable structure regression (DSR) tracker. We build an appearance model of context and constraint information by a 2-layer deformable structure model, and propose a object location confidence regression for model update. Experiments on many challenging sequences demonstrate that DSR tracker performs favorably when compared with several state-of-the-art algorithms.

In summary, we focus on a tracking algorithm with efficient and simple binary feature extraction, dense sampling strategy with continuous label outputs and learning model with deformable structure regression. The main contributions of this paper are as follows:

1) A 2-layer deformable structure appearance model to represent the object robustly;

2) A fast dense sampling strategy to extract object feature for training and detection;

3) A Coupled-layer Regression learning model to combine root-part and appearance-shape information.

The rest of the paper is organized as follows: we discuss related work in Section II. Section III introduces our DSR tracker with new appearance model and learning algorithm, and in Section IV we perform extensive experimental comparison with the state-of-the-art. The conclusion is drawn in Section V.

II. Related Works

This section reviews the related approaches for the components of our tracker. We consider that the recent approaches provide good mechanisms for building a tracker.

A. Tracking-by-detection

Tracking-by-detection is a algorithm which attempts to learn a classifier to distinguish a target object from its local background [2]. There are some issues need to be raised. Firstly, the classifier is to predict instance labels which is different from the goal of tracking to estimate object location. Some approaches try to overcome this problem by structure output [7] or context learning [9]–[11]. Secondly, training samples are weighted equally and labeled based on intuitions. Some attempts are proposed to solve it [3], [4], [11]. The learning model which we present tries to overcome all these problem with couple-layer regression framework.

B. Deformable parts models

A deformable parts model (DPM) [12] consists of a set of part filters, a set of deformations that provide geometric information regarding the expected placement of parts in a patch, a scoring function that provides the basis for combining the deformations and part-filter responses, and a strategy to find the given target [13]. Felzenszwalb et al. [12] describe a complex training procedure for the parts structure selections and initializations. However, in tracking application, so complex online training model is unacceptable. In our implementation, simple and efficient part structure settings are designed for online updating. Experiments show the simple part structure is sufficient to obtain state-of-the-art performance.

C. Brief feature and Ferns

Recently, numerous binary descriptors have been proposed which compute directly on image patches [14]–[16]. They are fast to compute, with a very small memory footprint and perform well. These characteristics shows such descriptors are perfect candidates for real-time applications. However, binary features are rare used in visual tracking systems. Most of tracking systems use general Haar-like feature or HOG feature as image representation [2], [3]. They believe that binary descriptors tend to be less robust and less accurate than complicated approaches. In this paper, Brief descriptor is used [16] as object representation. Our previous works [17] also regarded image patch space as a hypercube, and thought that Brief is a set of random hyperplanes $H = \{h_1, \dots, h_m\} \in \mathbb{R}^n$. Although binary descriptor is computed by simple testing directly to image patch, the experiments show that our binary deformable model is robust to appearance variations.

To make use of binary feature efficiency, we build fern [18] for model learning which avoid the step of computing complex float parameters. Fern is proposed as classifier for fast keypoints recognition. We modify the fern to adapt to regression learning which is discussed in detail in Section III-C.

III. Tracking Structure

A. Appearance model

The 2-layer deformable structure appearance model is shown in Fig. 1. An object instance is represented by $(\boldsymbol{x}, y, h)$ where $\boldsymbol{x}$ is location patch, y is continuous label which denotes the confidence of an object location. Latent variable $h = (V, \vec{\boldsymbol{p}})$ where V is the index that $V = a$ is the root node, $V = b_{i=1,2...}$ is the part node, and $\vec{\boldsymbol{p}}$ is the position of the node. The root layer has 1 node which represents the entire target. The root node has 9 child nodes $(b_i, \vec{\boldsymbol{p}})$ at the local layer in a 3 by 3 grid layout, each of which represents one part of the target. To simplify, the number of layers and the part numbers are fixed for different targets. The feature vector is defined as follow:

$$\boldsymbol{\Phi}(\boldsymbol{x}, y, h) = \left(\boldsymbol{\Phi}_B\left(\boldsymbol{x}, V\right), \boldsymbol{\Phi}_S\left(h\right)\right) \tag{1}$$

$\boldsymbol{\Phi}_B(\boldsymbol{x}, V)$ are the appearance features which contain Brief descriptors. We followed the implementations of [16] to calculate the part features. In addition, an image pyramid of the object was built to calculate the root feature which is shown in Fig. 1. We represent root feature $\boldsymbol{\Phi}_B(\boldsymbol{x}, a) \in \mathbb{R}^n$ where $n = (whl)^s$. With the multi-scale, the dimensionality n of the feature increases further which is typically in the order of 10^8 to 10^{12}. This high-dimensional feature is necessary to obtain good performance [19]. Similar approaches have been successful applied in detection and verification [19], [20].

The $\boldsymbol{\Phi}_S(h)$ are shape constraints $\boldsymbol{\Phi}_S(\vec{\boldsymbol{p}}_a, \vec{\boldsymbol{p}}_{b_i}), \forall a, b_i \in Ch(a)$ which encode the root-part pairwise spacial relationship. In detail, the shape constraints for a root-part pair (a, b_i) are defined as $\boldsymbol{\Phi}_S(\vec{\boldsymbol{p}}_a, \vec{\boldsymbol{p}}_{b_i}) = (\Delta\mu, \Delta\nu, \Delta\mu^2, \Delta\nu^2)$, where $(\Delta\mu, \Delta\nu)$ is the displacement of node b_i relative to its reference position by root node a.

Next, we define the score of our appearance model with two parts: 1) appearance score that measures the location confidence; 2) deformation score that measures the shape constraint of the deformable structure. Mathematically, the total score is defined as:

$$\begin{aligned} score(\boldsymbol{x}, h) = {} & w \cdot \boldsymbol{H}\left(\boldsymbol{\Phi}_B\left(\boldsymbol{x}, a\right)\right) \\ & + \sum_{i=1}^{9} \left[w \cdot \boldsymbol{H}\left(\boldsymbol{\Phi}_B\left(\boldsymbol{x}, b_i\right)\right) + \boldsymbol{w} \cdot \boldsymbol{\Phi}_S\left(a, b_i\right)\right] \end{aligned} \tag{2}$$

where $\boldsymbol{H}\left(\boldsymbol{\Phi}_B\left(\boldsymbol{x}, V\right)\right)$ is the confidence score of the object location, Section III-C discusses it in detail.

B. Detection

Rather than a sliding window search strategy for detection, we propose a fast feature representation method with dense sampling. Different from sample in advance and then extract

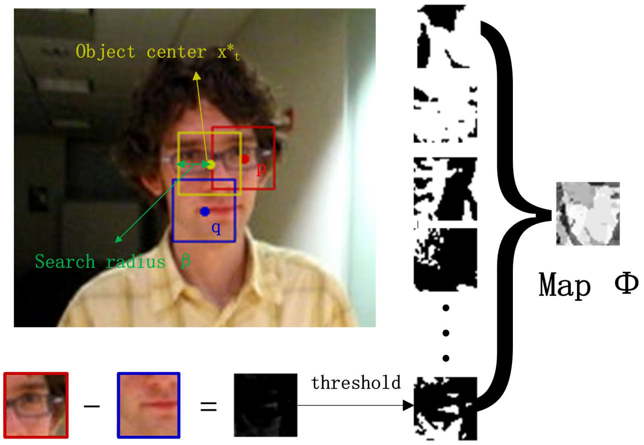


Fig. 2. Illustration of fast feature extraction with dense sampling. p,q is a pair pixels defined by Brief. We dense sample around this pair, and get a set of binary tests. We combine these samples' binary tests to build their Brief descriptor, namely feature map Φ_B.

feature for each instance, we combine sampling and feature extraction as one step which is shown in Fig. 2. When $(t+1)$-th frame arrived, we fast sample densely $\mathbf{X}^{\beta} = \{\boldsymbol{x} | \left|I_{t+1}(\boldsymbol{x}) - I_t(\boldsymbol{x}^*)\right| < \beta\}$ surrounding the old object location, where $\boldsymbol{x}$ is the image patch, $\boldsymbol{x}^*$ is the object patch and I_t is the sample location at the t-th frame. Then regression fern is applied to find the parts location with the maximum confidence

$$\boldsymbol{x}_{i,t+1}^* = \arg\max \boldsymbol{H}\left(\boldsymbol{\Phi}_B\left(\boldsymbol{x}_{i,t}, h_{i,t}\right)\right)$$

Based on the parts location, we can search the root location according to

$$\begin{aligned}\boldsymbol{x}_{t+1}^* &= \arg\max score(\boldsymbol{x}, h)\\ &= \arg\max\{w \cdot \boldsymbol{H}\left(\boldsymbol{\Phi}_B\left(\boldsymbol{x}, a\right)\right)\\ &+ \sum_{i=1}^{9}\left[w \cdot \boldsymbol{H}\left(\boldsymbol{\Phi}_B\left(\boldsymbol{x}, b_i\right)\right) + \boldsymbol{w} \cdot \boldsymbol{\Phi}_S\left(a, b_i\right)\right]\}\end{aligned} \tag{3}$$

With the new location patch $\boldsymbol{x}_{t+1}^*$, the tracker learning model can be updated.

C. Learning

A coupled-layer regression model is built for model updating, which is inspired by [10], [18], [21]. We dense sample a set of patches $\mathbf{X}^{\alpha} = \{\boldsymbol{x} | \left|I_t(\boldsymbol{x}) - I_t(\boldsymbol{x}^*)\right| < \alpha\}$. We set learning radius $\alpha = \min(w, h)/2$, where w, h is the initial size of the object bounding box. We represent these patches feature $\boldsymbol{\Phi}_B(x)$ with the method that we proposed in Section III-B.

1) Confidence map: Most trackers treat learning model as a classification [1] with examples pairs $(\boldsymbol{x}, y)$ where $y = \pm 1$ is the label. However, equal weighted samples confuse the classifier [4]. In this paper, we don't limit discrete label for the samples and measure the continuous label with a confidence map. The confidence map is modeled with the distance between target center and sample location as follow:

$$y(\boldsymbol{x}) = 1 - \theta \left|I(\boldsymbol{x}) - I(\boldsymbol{x}^*)\right| \tag{4}$$

where $\theta = 1/\alpha$ is scale factor. The closer the location is to the current tracked position, the larger confidence probability is.

2) Regression fern construction and update: To adapt to our binary deformable appearance model, we build a linear regression fern for online model learning. Suppose we are given a set of instance pairs $(\boldsymbol{\Phi}_B^N(\boldsymbol{x}), y)$, the task of regression fern is to learn the probability that the instance belongs to the target. Since fern is simple, enlarging fern dimension N is required for accuracy. However, the computational complexity will be prohibitively high for real-time object tracking. To balance the storing requirement and the performance, we partition feature $\boldsymbol{\Phi}_B^N$ into M groups of size $S = \frac{N}{M}$ [18]. Each group $\boldsymbol{\Phi}_{Bm}^S$ is a independent regression fern, and we compute the instance confidence probability as follow:

$$\boldsymbol{H}\left(\boldsymbol{\Phi}_{Bm}^S = k\right) = \bar{y} = \frac{\sum^N y}{N_k} \tag{5}$$

where N_k is the instances number of $\boldsymbol{\Phi}_{Bm}^S = k$. When a new frame comes, the regression fern is updated incrementally

$$\boldsymbol{H}_{t+1}\left(\boldsymbol{\Phi}_{Bm}^S\right) = (1-\gamma)\boldsymbol{H}_t\left(\boldsymbol{\Phi}_{Bm}^S\right) + \gamma \boldsymbol{H}_{t,new}\left(\boldsymbol{\Phi}_{Bm}^S\right) \tag{6}$$

where $\gamma > 0$ is the learning rate for update. Many weak responses are combined in a Naive Bayesian [22] way.

$$\begin{aligned}\boldsymbol{H}\left(\boldsymbol{\Phi}_B\right) &= \boldsymbol{H}\left(\boldsymbol{\Phi}_{B1}^S, \boldsymbol{\Phi}_{B2}^S, \ldots, \boldsymbol{\Phi}_{BM}^S\right)\\ &= \prod_{l=1}^{M} \boldsymbol{H}\left(\boldsymbol{\Phi}_{Bl}^S\right)\end{aligned} \tag{7}$$

3) Shape constraint update: We denote w_t the weight parameters to predict confidence score for the t-th frame defined by (2). To train the learning model, we solve the optimization problem via

$$\boldsymbol{w}_{t+1} = \arg\min\left\{\frac{1}{N}\sum_{i=1}^{N}\left[score\left(\boldsymbol{x}, h\right) - y\right]^2 + \frac{\lambda}{2}\|\boldsymbol{w}\|^2\right\} \tag{8}$$

where $score(\boldsymbol{x}, h)$ is defined by (2). This is a standard linear regression. We consider the sub-gradient of the above objective,

$$\nabla_t = \lambda \boldsymbol{w}_t + \frac{1}{N}\sum_{i=1}^{N}\left\{\left[score\left(\boldsymbol{x}, h\right) - y\right] \cdot score'\right\} \tag{9}$$

where $score'$ is the derivative of confidence score. We then update $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \nabla_t$ with step size $\eta_t = \frac{1}{\lambda t}$, it can be written as

$$\boldsymbol{w}_{t+1} = (1 - \eta_t\lambda)\boldsymbol{w}_t - \frac{\eta_t}{N}\sum_{i=1}^{N}\left\{\left[score\left(\boldsymbol{x}, h\right) - y\right] \cdot score'\right\} \tag{10}$$

D. Analysis and Discussion

We note that 2-layer deformable structure and simple regression model are two prime characteristics of our proposed tracker. Our algorithm is illustrated in Algorithm 1. In addition, our tracker achieves robust performance as discussed below:

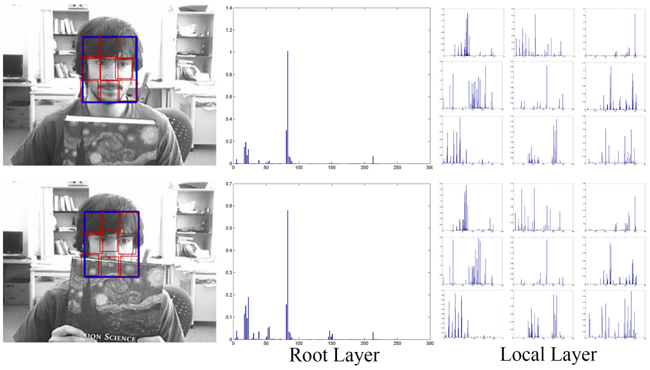


Fig. 3. Illustration of the reason that our appearance model is robust to occlusion. The right part is the confidence score histogram of regression fern.

1) Robustness to occlusion: As shown in Fig. 3, our proposed algorithm is robust to heavy occlusion as most of part appearance models are not occluded. These unobstructed part models prevent the tracker drift away. In addition, the root regression fern is slightly affected by occlusion (See Fig. 3), which demonstrates that Brief descriptor is effective in handling appearance variation.

2) Robustness to ambiguity of the distractor: If tracking the target only base on its appearance information, the tracker will be distracted easily because of similar appearances. Multiple instance learning [3] schemes to alleviate this problem. Our dense sampling strategy solve the problem. Different from most trackers treat the positive samples equally, we give each sample a continuous label which represents the sample probability belong to the target accurately, and replace the classifier with regression model. These strategy help our tracker distinguish target from distractor.

Algorithm 1: Deformable Structure Regression Tracking

```
Input: $(t+1)^{th}$video frame $I_{t+1}$, model $\boldsymbol{H}_t, \boldsymbol{w}_t$, target
    location patch $\boldsymbol{x}_t^*$
    // detection
    for $i = 1, 2 \ldots, 9$ do
        $\boldsymbol{x}_{i,t+1}^* = \arg\max_{\boldsymbol{x} \in \boldsymbol{X}^\gamma} \boldsymbol{H}\left(\boldsymbol{\Phi}_B\left(\boldsymbol{x}_i, b_i\right)\right)$
    end for
    $\boldsymbol{x}_{t+1}^* = \arg\max score(\boldsymbol{x}, h)$ // apply (3)
    // learning
    $\boldsymbol{H}_{t,new} \leftarrow \boldsymbol{\Phi}_B(\boldsymbol{x}_{t+1}), y$ // apply (5)
    $\boldsymbol{\Phi}_S(h) \leftarrow I_{t+1}(\boldsymbol{x}_{t+1}^*), I_{t+1}(\boldsymbol{x}_{i,t+1}^*)$
    // update
    $\boldsymbol{H}_{t+1} \leftarrow \gamma \boldsymbol{H}_{t,new} + (1-\gamma)\boldsymbol{H}_t$
    $\boldsymbol{w}_{t+1} \leftarrow (1-\eta\lambda)\boldsymbol{w}_t - \frac{\eta}{N}\sum_{i=1}^{N}\left\{[score\left(\boldsymbol{x}, h\right) - y] \cdot score'\right\}$
Output: new location patch $\boldsymbol{x}_{t+1}^*, \boldsymbol{H}_{t+1}, \boldsymbol{w}_{t+1}$
```

IV. Experiments

In this section, we compare our tracker with 10 state-of-the-art trackers on 18 challenging video clips. For fair evaluation, we use the original source codes [1] in which parameters of each method are tuned for best performance. The 10 evaluated trackers are: compressive tracking (CT) [5], structured output tracker (Struck) [7], TLD tracker [23], MIL tracker [3], circulant structure kernel tracker (CSK) [10], online Adaboost tracker (OAB) [2], fragment tracker (Frag) [24], local-global tracker (LGT) [21], visual tracking decomposition (VTD) method [25] and distribution field tracker (DFT) [26]. The parameters of our tracker are *fixed* for all the experiments. Since the algorithms involve some random parameters, we repeat the experiment 10 times on each sequence, and present the averaged results. Our proposed algorithm is implemented in C++ on an AMD A8-5600K 3.60 GHz APU with 8 GB RAM. The average running time of our tracker is 27.4 frames per second (FPS).

A. Experimental setup

The parameters are set as follows. The search radius for detection is set to $\beta = 20$ and we dense sample about 1600 samples. Ferns parameters are set to $S = 8$, $M_{root} = 20$ and $M_{part} = 15$. In addition, the learning rate γ in (6) and λ in (10) are set to 0.125 and 0.095.

B. Experimental results

Two metrics are used for quantitative analysis with 10 tracking algorithms. The first evaluation metric is center location error (CLE). The other is success rate which is defined as $score = \frac{|R_g \cap R_t|}{|R_g \cup R_t|}$, where R_g is the ground truth bounding box and R_t is the tracked bounding box, and the result is considered as successful tracked if the $score > 0.5$.

1) Overall performance: Table I shows the experimental result in terms of CLE, and Tabel II reports SR of various tracking results. Our tracking algorithm achieves the best or second best performance on most test sequences. Fig. 4 shows some tracking results of different trackers. For the reason of clarity, we just present our tracker compared with the CT [5], struck [7], TLD [23] ,MIL [3] and CSK [10] methods which are similar to our work and perform well.

2) Appearance variation and occlusion: There are large appearance variations in the sequences such as *faceocc2* (#480, #740 of the *faceocc2* in the Fig. 4). Only Struck, CSK and our DSR adapt to illumination change and pose variation well. In addition, Struck and our DSR achieve favorable performance on the *girl* sequence due to taking full advantage of context information. The *david* sequence contains pose and scale variation. Nearly all the other trackers fail to track the target except CT and DSR algorithms. The favorable performance of our DSR tracker is attributed to our deformable structure appearance model (as discussed in Section III-D).

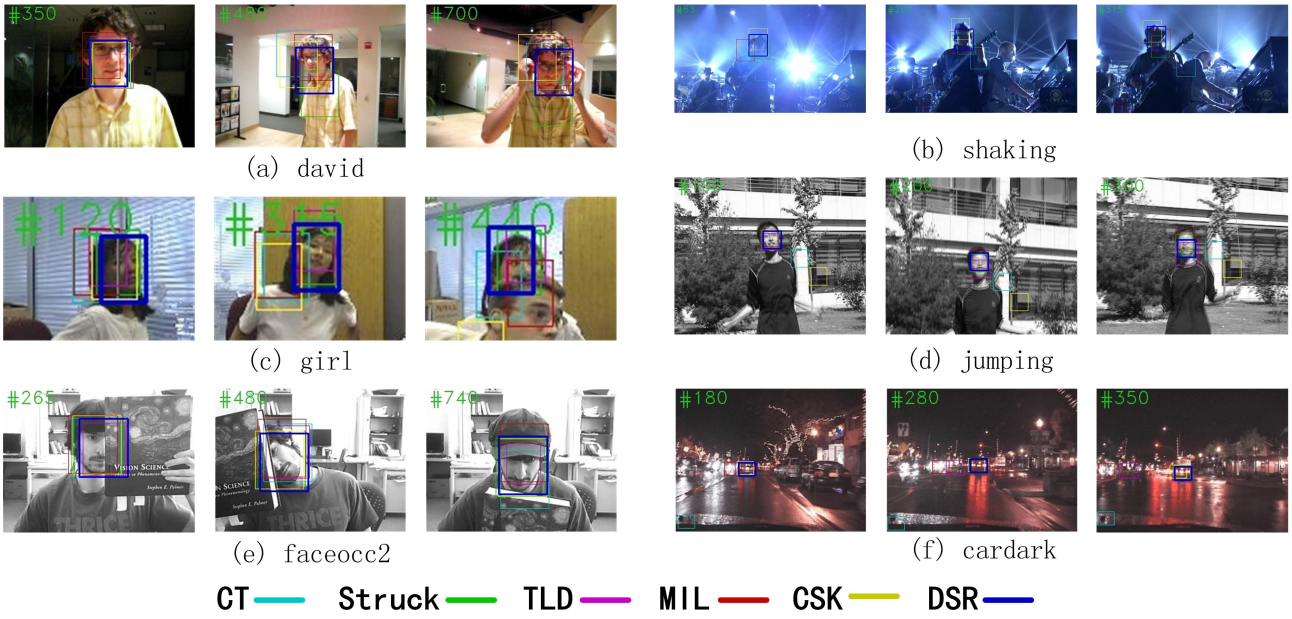
3) Background clutter and abrupt motion: The target objects in the *shaking*, *jumping* and *cardark* sequences undergo fast movements in the cluttered background. However, the proposed DSR achieves the best performances in these sequences. For example, in the *cardark* sequence, most trackers drift to background except Struck and DSR (See #280, #350 of the *cardark* in the Fig. 4). Although the target and the background have similar appearance, the structure relationship is utilized

TABLE I
CENTER LOCATION ERROR(CLE)(IN PIXELS). RED FONTS INDICATE THE BEST PERFORMANCE WHILE THE SECOND BEST ONES IS SHOWN IN BLUE.

Sequence	CT [5]	Struck [7]	TLD [23]	MIL [3]	CSK [10]	OAB [2]	Frag [24]	LGT [21]	VTD [25]	DFT [26]	DSR
faceocc2	19	6	12	14	7	20	16	13	8	8	6
boy	9	4	5	13	21	3	40	12	8	107	4
football	12	17	14	12	16	73	5	13	14	9	4
fish	11	3	7	24	41	87	22	27	17	9	6
faceocc1	26	19	27	30	12	25	11	37	20	24	9
freeman1	119	14	40	11	127	36	10	56	10	10	6
skating1	151	84	146	140	9	44	150	95	9	175	23
fleetface	59	23	41	63	26	52	68	43	46	68	17
trellis	42	7	31	72	19	98	59	15	32	45	21
david	10	43	15	17	18	22	82	27	12	43	11
jumping	48	7	7	10	86	46	6	26	42	67	5
girl	19	3	10	14	19	6	21	15	9	24	5
sylvester	9	6	7	15	10	15	15	15	20	45	8
shaking	80	30	37	24	18	191	192	55	17	27	9
singer2	127	175	58	23	186	187	89	35	44	22	23
suv	72	50	13	82	576	31	42	68	57	111	18
doll	22	9	16	17	45	12	14	15	7	60	4
cardark	119	1	28	44	4	3	36	25	17	59	3

TABLE II
SUCCESS RATE(SR)(%). RED FONTS INDICATE THE BEST PERFORMANCE WHILE THE SECOND BEST ONES IS SHOWN IN BLUE.

Sequence	CT [5]	Struck [7]	TLD [23]	MIL [3]	CSK [10]	OAB [2]	Frag [24]	LGT [21]	VTD [25]	DFT [26]	DSR
faceocc2	61	87	59	71	95	60	67	75	83	82	91
boy	73	100	86	49	84	100	45	73	79	48	100
football	76	75	45	74	78	36	88	78	72	84	90
fish	78	100	89	37	4	4	51	45	63	86	93
faceocc1	67	89	73	65	97	74	97	73	91	80	95
freeman1	11	21	22	13	16	27	20	23	24	24	35
skating1	5	38	21	8	63	46	10	68	72	16	68
fleetface	53	68	52	48	69	56	48	53	67	61	73
trellis	12	70	47	6	15	13	37	63	42	52	53
david	24	27	88	6	24	14	7	52	36	28	66
jumping	1	93	88	62	5	5	84	9	12	12	97
girl	10	92	57	20	42	91	49	73	50	25	89
sylvester	69	90	67	43	61	67	64	76	74	36	76
shaking	4	9	36	18	73	1	7	37	90	80	93
singer2	1	4	10	47	4	3	20	65	44	60	43
suv	23	57	91	13	57	76	71	68	48	5	83
doll	61	39	72	39	33	71	58	23	74	25	81
cardark	0	100	54	18	98	94	25	39	68	34	97



(a) david

(b) shaking

(c) girl

(d) jumping

(e) faceocc2

(f) cardark

Fig. 4. Screenshots of tracking results.

TABLE III

COMPARISONS OF MODELS. L IS THE LAYER NUMBER, N_p IS THE PARTS NUMBER OF THE LOCAL LAYER. RED FONTS INDICATE THE BEST PERFORMANCE.

	Sequences	david	girl	faceocc2	doll
CLE	$L = 1$	19	18	10	7
	$L = 2, N_p = 4$	17	12	9	8
	$L = 2, N_p = 9$	11	5	6	4
	$L = 2, N_p = 16$	12	4	6	4
SR	$L = 1$	55	77	79	74
	$L = 2, N_p = 4$	53	82	83	75
	$L = 2, N_p = 9$	66	89	91	81
	$L = 2, N_p = 16$	53	88	92	83

in our algorithm to help distinguish background. In addition, large number of dense samples and accurate sample confidence are used by our tracker when the learning model updates, thereby avoid distraction.

C. Analysis of 2-layer structure model

In order to study how much gain is obtained by deformable appearance model, besides our proposed model (2-layer with 3×3 parts) introduced before, we also implemented three other models: only root appearance model, 2-layer with 2×2 parts and 2-layer with 4×4 parts. The comparison of these 4 models are performed on $david$, $girl$, $faceocc2$ and $doll$ sequences. Table III shows the performance of these models. It is clear that adding part models improves the performance. However, too many parts will not cause much improvement or even increase the computation. In conclusion, 2-layer with 9 parts is the best choice for appearance model.

V. CONCLUSION

In this paper, we propose a real-time robust tracking algorithm with deformable structure regression. Two-layer appearance model is proposed which is robust to appearance variations of occlusion, pose variations and illumination changes. Coupled-layer regression learning model combines global and local information, prevents drift away caused by noisy background or misaligned samples. Numerous experiments with state-of-the-art algorithms on challenging sequences demonstrated that the proposed tracker achieves favorable performance in terms of accuracy, robustness and speed.

ACKNOWLEDGMENT

The authors thank all researchers who are committed to visual tracking, especially those who provide source code.

REFERENCES

[1] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. v. d. Hengel, "A survey of appearance models in visual object tracking," *arXiv preprint arXiv:1303.4803*, 2013.

[2] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. BMVC*, vol. 1, 2006, p. 6.

[3] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," vol. 33, no. 8, pp. 1619–1632, 2011.

[4] K. Zhang and H. Song, "Real-time visual tracking via online weighted multiple instance learning," *Pattern Recognition*, 2012.

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

[6] ——, "fast compressive tracking," *IEEE Transactions on Pattern Analysis and Machine*, 2014.

[7] S. Hare, A. Saffari, and P. H. Torr, "Struck: Structured output tracking with kernels," in *Proc. Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 263–270.

[8] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[9] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 2544–2550.

[10] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 702–715.

[11] K. Zhang, L. Zhang, M.-H. Yang, and D. Zhang, "Fast tracking via spatio-temporal context learning," *arXiv preprint arXiv:1311.1939*, 2013.

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," vol. 32, no. 9, pp. 1627–1645, 2010.

[13] T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik, "Fast, accurate detection of 100,000 object classes on a single machine," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[14] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proc. Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 2564–2571.

[15] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 2548–2555.

[16] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: Computing a local binary descriptor very fast," vol. 34, no. 7, pp. 1281–1298, 2012.

[17] W. Shoujue and L. Jiangliang, "First step to multi-dimensional space biomimetic informatics M," *National Defense Industry Press, Beijing*, 2008.

[18] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua, "Fast keypoint recognition using random ferns," vol. 32, no. 3, pp. 448–461, 2010.

[19] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification."

[20] G. Duan, C. Huang, H. Ai, and S. Lao, "Boosting associated pairing comparison features for pedestrian detection," in *Proc. Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, 2009, pp. 1097–1104.

[21] L. Cehovin, M. Kristan, and A. Leonardis, "An adaptive coupled-layer visual model for robust visual tracking," in *Proc. Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 1363–1370.

[22] A. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," *Advances in neural information processing systems*, vol. 14, p. 841, 2002.

[23] Z. Kalal, J. Matas, and K. Mikolajczyk, "Pn learning: Bootstrapping binary classifiers by structural constraints," in *Proc. Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 49–56.

[24] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006, pp. 798–805.

[25] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 1269–1276.

[26] E. G. Learned-Miller and L. S. Lara, "Distribution fields for tracking," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2012.