Part Context Learning for Visual Tracking

Guibo Zhu
gbzhu@nlpr.ia.ac.cn
Jinqiao Wang
jqwang@nlpr.ia.ac.cn
Chaoyang Zhao
chaoyang.zhao@nlpr.ia.ac.cn
Hanqing Lu
luhq@nlpr.ia.ac.cn

National Laboratory of Pattern Recognition,
Institute of Automation,
Chinese Academy of Sciences,
Beijing, China.

Abstract

Context information is widely used in computer vision for tracking arbitrary objects. Most existing works focus on how to distinguish the tracked object from background or inter-frame object similarity information or key-points supporters as their auxiliary information to assist them in tracking. However, in most cases, how to discover and represent both the intrinsic property inside the object and surrounding information is still an open problem. In this paper, we propose a unified context learning framework that can capture stable structure relations of in-object parts, context parts and the object itself to enhance the tracker's performance. The proposed Part Context Tracker (PCT) consists of an appearance model, an internal relation model and an context relation model. The appearance model represents the appearances of the object and parts. The internal relation model utilizes the parts inside the object to describe the spatio-temporal structure property directly, while the context relation model takes advantage of the latent intersection between the object and background parts. Then the appearance model, internal relation model and context relation model are embedded in a max-margin structured learning framework. Furthermore, a simple robust update strategy using median filter is utilized, which can deal with appearance change effectively and alleviate the drift problem. Extensive experiments are conducted on various benchmark dataset, and the comparisons with state-of-the-arts demonstrate the effectiveness of our work.

1 Introduction

Visual tracking is a fundamental problem in computer vision and has a wide range of applications including surveillance, and human-computer interaction [20, 31]. For a visual tracking algorithm, it should be designed to cope with the inevitable appearance changes due to occlusion, rotation, illumination, etc. Recent progresses in object tracking [1, 3, 16, 18, 25, 35, 36] have yielded a steady increase in performance, but designing a robust algorithm to track generic objects in presence of occluded and deformable targets is still a major challenge.

To overcome this difficulty, numerous models have been designed, most of them focus on building a strong appearance model to encode the variations of the object appearance and distinguishing it from the background. Some methods [1, 5, 6, 17] exploit multiple object

fragments or patches to represent the object appearance effectively. Meanwhile, context information can be utilized to track and it has been employed recently in several tracking methods [9, 15, 26, 33].

Global context is often used in tracking to assist classifying the object with background information. However, global context cannot deal with the object deformation problem, while the local part context interactions are relatively stable. When the target appearance changes gradually, the intrinsic property of internal interaction between the parts inside object and context interaction between object and background are relative stable in spatio-temporal 3D space of tracking. To explore the structure property and stable relationship for overcoming complex environments, we propose a novel part context model which comprises appearance model, internal relation model and context relation model. The internal relation model utilizes the parts inside the object to describe the spatio-temporal structure property directly, while the context relation model takes advantage of the latent intersection between the object and background parts or the contour information.

1.1 Related work

For online tracking in unconstrained environments, merely learning the descriptive [1, 21, 22, 36] or discriminative features [3, 14, 16] of the target cannot ensure the robustness of the system. Yang *et al*. [28] constructed a context-aware tracker (CAT) to track random field around the target instead of the target. The tracker in [15] utilizes strong motion coupling constraints to locate the target even when the target is invisible, with the help of some available related context information. However, detecting and matching all of the local features are expensive and the motion of the object is not easily predicted. Dinh *et al*. [9] developed a new context framework based on distracters and supporters. Wen *et al*. [26] proposed a spatio-temporal context method in which temporal context captures the historical appearance information and spatial context model integrates key-points based contributors. Generally, [9, 15, 26] work with the key points as auxiliary information, the main differences are how to utilize supporters or distracters. Although the introduction of context in these trackers expands the available information which can be obtained from the scene, it may collapse when motion blur occurs due to the utilization of key-points descriptors.

An object detection approach with structured output SVM [24] was proposed in [4]. Motivated by this success, structured learning was applied to online visual tracking [16, 29]. Inspired by deformable part-based appearance models [2, 12, 38], Zhang and van der Maaten [34] proposed a structure preserving model and Yao *et al*. [30] presented part-based with latent structural learning for tracking. Although the two approaches pay attention to the parts of the object and their deformation cost, there are still many intrinsic properties in object tracking (e.g. temporal constraints, context information) which have not been considered.

1.2 Our approach

The Part Context Tracker (PCT) consists of an appearance model, an internal relation model and an context relation model. The internal relation model formulates the temporal relations of the object itself or the in-object parts themselves and the spatio-temporal relations between the object and in-object parts. The context relation model constructs the spatio-temporal relations between the in-object parts and the context parts and the temporal relations of the context parts themselves. Hence the physical properties and the appearance information are

considered in the optimization process through parts and relations. The contributions are as follows:

(1) We first propose a unified context framework which formulates the single object tracking as a part context learning problem.

(2) The in-object parts and context parts are selected so that we not only pay attention to the appearance of object, but also focus on the relations among the object, the in-object parts and the context parts.

(3) A simple robust update strategy using median filter is utilized, thereby enabling the tracker to deal with appearance change effectively and alleviate the drift problem.

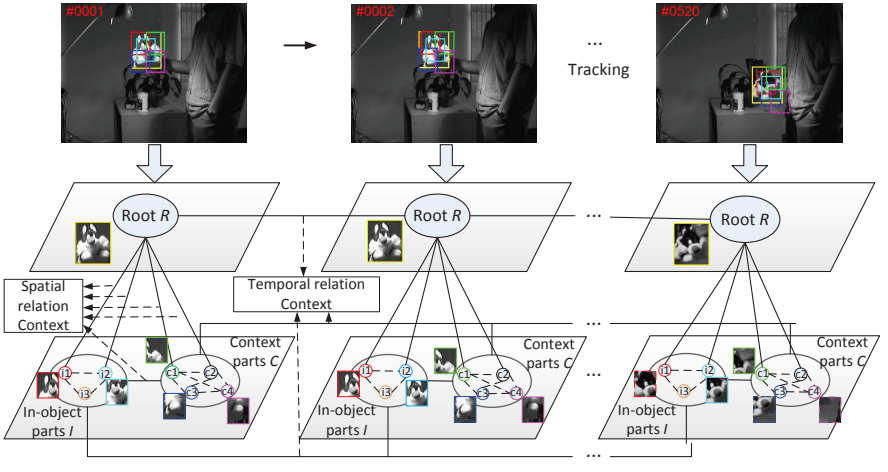

Figure 1: Illustration of our Part Context Tracking framework using the "sylvester" video.

2 Part Context Tracking

In this section, we show how to represent and track an object with parts in a unified framework. We first introduce the part context formulation and then describe the model training problem with a structured learning approach. After the learning mechanism, we develop an online learning strategy to update the model parameters efficiently.

2.1 Model definition

Our framework not only models the object with in-object parts, but also incorporates the interaction between the object and background with context parts. The deformable configuration [11, 13] together with the temporal structure of these parts are also considered in.

In Fig. 1, with the object bounding box as the root R, the in-object parts I are defined as the parts selected inside R, which covers part of the object appearance. The context parts C are selected from the overlapping area between the object and the background. For a target with K in-object parts and M context parts, the configuration is denoted as $B = (B_0, B_1, ..., B_K, B_{K+1}, ..., B_{K+M})$. Where B_0 stands for the target bounding box R, $(B_1, ..., B_K) \in I$ are the K in-object part boxes, and $(B_{K+1}, ..., B_{K+M}) \in C$ are the M context part boxes. The

corresponding features of the root and parts are represented as $X = (x_0, ..., x_K, x_{K+1}, ..., x_{K+M})$. In a word, our framework models the object with three components:

$$M = M_A + M_I + M_C, \tag{1}$$

where M_A, M_I and M_C are the appearance model, the internal relation model and the context relation model respectively.

For online tracking, an appearance model is essential. It represents the intrinsic property of one object or the discriminative information between the object and background. To better mine the information, we factorize the appearance model M_A as Eq. (1):

$$\begin{aligned} M_A &= A_R + A_I + A_C \\ &= w_R^T \Phi_R(x_0) + \sum_{i=1}^{K} w_I^T \Phi_I(x_i) + \sum_{i=K+1}^{K+M} w_C^T \Phi_C(x_i). \end{aligned} \tag{2}$$

where A_R, A_I and A_C are the global root appearance model, in-object parts appearance model and context parts model separately. Φ_R, Φ_I and Φ_C denote the root appearance feature, the in-object parts appearance feature and the context parts appearance feature. w_R, w_I and w_C are the weights of appearance features correspondingly. x_i is the i^{th} part corresponding to bounding box $B_i = (c_i, r_i, w_i, h_i)$ with center location $B_{i,c} = (c_i, r_i)$, width w_i and height h_i.

In addition to the appearance model, all relatively stable spatio-temporal relations between the object and its corresponding parts frame-to-frame should be utilized in tracking. Therefore we design the internal relation model to formulate the interactions between root and the in-object parts, which includes the spatial constraints and the temporal constraints between them, we define M_I as:

$$\begin{aligned} M_I &= S_I + E_R + E_I \\ &= \sum_{i=1}^{K} w_{R,I}^T \Phi_{R,I}(x_0, x_i) + \sum_{t=-H}^{-1} w_{t,R}^T \Phi(x_0^t, x_0) + \sum_{t=-H}^{-1} \sum_{i=1}^{K} w_{t,I}^T \Phi(x_i^t, x_i) \end{aligned} \tag{3}$$

where S_I, E_R, and E_I are spatial relation between root and in-object parts, temporal relation between root and their historical roots, and temporal relation between in-object parts and their historical information respectively. $\Phi_{R,I}(x_0, x_i)$ denotes the spatial interaction function between the root B_0 and in-object part B_i. $\Phi(x_0^t, x_0)$ is the temporal relation function of the bounding box B_0 in the last t^{th} frame and the current frame. Likely, $\Phi(x_i^t, x_i)$ is the bounding box B_i's temporal relation function. $w_{R,I}$, $w_{t,R}$ and $w_{t,I}$ are the weights correspondingly. Similar to [12], the spatial interaction between x_i and x_j is $f_c = (c_j - c_i, r_j - r_i)$ and:

$$\Phi(x_i, x_j) = (f_c, f_c^2). \tag{4}$$

Herein, f_c and f_c^2 can preserve the relative and absolute information between x_i and x_j. For detail, the temporal relation function $\Phi_{x_i^t, x_i}$ can be represented as:

$$\Phi(x_i^t, x_i) = exp(-(||B_{i,c}^t - B_{i,c}||^2 / \delta^2)) \tag{5}$$

where δ is a constant value, H is upper bound of the last frames

Except internal relations inside the object, some information in latent intersection area between the object and background is neglected by previous works, such as the partial contour and the object are consensus in motion. To make full use of the information, we formulate the context relation model to express the interactions between root and the context parts,

which also includes the spatial and temporal constraints between them. Similar to Eq. (3), we describe the context relation model mathematically as:

$$\begin{aligned} M_C &= S_C + S_{C,I} + E_C \\ &= \sum_{j=1}^{M} w_{R,C}^T \Phi_{R,C}(x_0, x_i) + \sum_{i=1}^{K}\sum_{j=1}^{M} w_{C,I}^T \Phi_{C,I}(x_i, x_j) + \sum_{t=-H}^{-1}\sum_{j=1}^{M} w_{t,C}^T \Phi(x_j^t, x_j) \end{aligned} \tag{6}$$

where S_C, $S_{C,I}$ and E_C denote spatial relation between root and context parts, spatial relation between in-object parts and context parts, and temporal relation between context parts and their historical information. $\Phi_{R,C}(x_0, x_j)$ denotes the spatial interaction function between the root B_0 and context part B_j, $\Phi_{C,I}(x_i, x_j)$ denotes the spatial interaction function between the in-object part B_i and the context part B_j. $\Phi(x_j^t, x_j)$ denotes the bounding box B_j's temporal relation function. $w_{R,C}$, $w_{C,I}$ and $w_{t,C}$ are the weights corresponding to $\Phi_{R,C}(x_0, x_j)$, $\Phi_{C,I}(x_i, x_j)$ and $\Phi(x_j^t, x_j)$ respectively.

For the linear property, the model of object and its configuration can be simplified as:

$$M = w^T \Phi(X) \tag{7}$$

where

$$w = [w_R^T, w_I^T w_C^T, w_{R,I}^T, w_{R,C}^T, w_{I,C}^T, w_{t,R}^T, w_{t,I}^T, w_{t,C}^T]^T, \tag{8}$$

$$\begin{aligned} \Phi(X) = &[\Phi_R^T(x_0), \Phi_I^T(x_i), \Phi_C^T(x_i), \sum_{i=1}^{K} \Phi_{R,I}^T(x_0, x_i), \sum_{t=-H}^{-1} \Phi^T(x_0^t, x_0), \sum_{t=-H}^{-1}\sum_{i=1}^{K} \Phi^T(x_i^t, x_i), \\ &\sum_{j=1}^{M} \Phi_{R,C}^T(x_0, x_i), \sum_{i=1}^{K}\sum_{j=1}^{M} \Phi_{C,I}^T(x_i, x_j), \sum_{t=-H}^{-1}\sum_{j=1}^{M} \Phi^T(x_j^t, x_j)]^T \end{aligned} \tag{9}$$

w is the model parameter we need to learn. Given a configuration B in a frame F, there needs a function to measure how best the configuration B matches object model M. We compute the similarity score as follows,

$$S(F, B, M) = S(F, B, M_A) + S(F, B, M_I) + S(F, B, M_C) \tag{10}$$

2.2 Optimization

In this section, we will describe the optimization of the proposed discriminative model from three aspects: inference, model learning and update strategy.

2.2.1 Inference

Even if the object appearance varies because of the influence of internal and external factors, the intrinsic relation between the object and intra-object parts or context parts could remain relatively stable in a short-time or long period time. Based on the assumption, given the definition of M, a model is constructed to constrain the deformation of parts by modeling their temporal and spatial relationships with the root. Adding pairwise or higher order interactions between arbitrary parts can capture more structural information, but it will result a loopy graph which is not efficient in inference. To avoid the problem, we not only keep the model to be tree-structured, but also introduce the temporal relationships based on the historical information without adding large computational complexity.

In tracking, there is strong correlation across continuous frames. We set the search radius r around the last object location. The bigger r is, the larger the computational complexity is. Standard sliding window procedure is used to scan images in different locations of the search radius with a fixed scale to determine how much the special window corresponds to the object. For each scanning window in image F, we first fit the window to structure model M to get the part configuration on it, and then calculate the score of the window according to the inferred configuration by Eq. (1)-(10).

The fitting step aims to find the candidate object's configuration B^* with the highest match score according to the learned model M on all configurations of a sliding window. Mathematically, the optimization problem is to find B^* that satisfies:

$$B_0^* = \arg\max_B S(F,B,M) = \arg\max_B w^T \Phi(X) \tag{11}$$

The score of each part in the model is independent once the root is specified, so that we can maximize the following problem instead:

$$B_i^* = \arg\max_{B_i} S(F,B_i,M) = \arg\max_{B_i} w^T \Phi(x_i) \tag{12}$$

where $\Phi(x_i)$ are the related items with x_i in $\Phi(X)$. The complexity of maximizing one single sliding window is high, but benefiting from the generalized distance transform proposed in [10], the average complexity in simultaneously optimizing all the the sliding windows is of linear complexity with the search radius.

2.2.2 Model learning

Like other trackers [3, 16, 34], to enhance its adaptivity and robustness, we need to update the model online. In general, most of online trackers use the tracked object configuration in previous frames as positive examples to update. We argue this method and choose to update the parameters in our structural model while the last object configuration satisfied some conditions (e.g. an update threshold or occlusion detection). We utilize the method of setting an adaptive update threshold for clarity. While exceeding the threshold, we assume it as a true positive example (F,X,B) at frame F.

Function S measures the compatibility between training pairs, and gives a high score to those which are well matched. By Eq. (7), it can be learned in a large-margin framework from a set of training sample pairs $\{(F,B_1),...,(F,B_n)\}$ by minimizing the following convex object function:

$$\min_{w,\eta\geq 0} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\eta_i \quad s.t. \forall i, \forall B \neq B_i : \langle w, \delta\Phi_i(B)\rangle \geq \Delta(B_i,B) - \eta_i \tag{13}$$

where $\delta\Phi_i(B) = \Phi(F,B_i) - \Phi(F,B)$. This optimization aims to ensure that the value of $S(F,B_i,M)$ is greater than $S(F,B,M)$ for $B \neq B_i$, by a margin which depends on a loss function Δ. Herein, $\Delta(B_i,B)$ measures dissimilarity between B_i and B, as in [4, 16, 34]:

$$\Delta(B_i,B) = 1 - \frac{B \cap B_i}{B \cup B_i}. \tag{14}$$

where the two bounding boxes B and B_i are both measured in pixels.

For training the structure SVM efficiently, we adopt the cutting plane algorithm [24] to select the most violated constraints to train. The most violated constraint can transform to structure SVM loss ℓ by configuration B:

$$\ell(w;F,B) = \max_{B}[S(F,B_i,M) - S(F,B,M) + \Delta(B,B_i)] \tag{15}$$

Like [7, 34], we use a passive-aggressive algorithm to perform the parameter update in the tracking process. The passive-aggressive algorithm sets the step size in such a way as to substantially decrease the loss without parameter updating too large. In particular, the passive-aggressive update algorithm uses the following parameter updates:

$$w \leftarrow w - \frac{\ell(w;F,B)}{||d||^2 + 0.5} d. \tag{16}$$

Herein, $d = \nabla_w S(w;F,\widehat{B}) - \nabla_w S(w;F,B)$ is gradient of the structured SVM loss, and $\widehat{B} = \arg\max_B (S(w;F,B) + \Delta(B_i,B)$.

2.2.3 Update strategy

When to update the object model is one of the critical problems to avoid drifting inherently in tracking process. If the object is occluded by other object, the model don't need to be updated. But if the object is self-occluded (e.g. rotation) or appearance changes due to illumination, the model updating is necessary. However, evaluating whether and when the appearance changes (e.g. occlusion) is a difficult problem. Therefore, most of the tracking algorithms update the appearance model every frame.

Like [34], we only update the weight vector w_i corresponding to part bounding box B_i when the exponentiated score for that object exceeds some threshold to avoid erroneous update of our appearance model. Different from [34], the threshold is adaptive and generated by median filter for leveraging the adaptivity and stability of the object model. The initial threshold V_1 in the first frame is set as a constant. Then the threshold in t^{th} frame V_t is:

$$V_t = \underset{i=\{0,1,\ldots,K+M\}}{median} exp(\frac{area(B_0)}{area(B_i)} w_i^T \Phi(x_i)) \tag{17}$$

where $area(B_i)$ denotes the area of bounding box B_i, and $K+M$ are the total number of parts. In particular, we only update the w_i whenever $exp(\frac{area(B_0')}{area(B_i')} w_i^T \Phi(x_i)) >= max(V_1, V_t)$, $max(V_1, V_t)$ is to get the maximum value from V_1 and V_t.

3 Experiments

To evaluate the performance of our part context learning tracker (PCT), ten challenging sequences from prior works [3, 19, 22, 23, 27, 32] are used. The sequences have different challenging aspects such as illumination variation, occlusion and out-of-view, etc. The quantitative comparison results with several state-of-the-art trackers: MIL [3], TLD [18], ContextTracker (CXT) [9], Struck [16], SCM [37], PartTracker (PT) [30], structure preserving tracker (SPOT) [34] and our tracker are shown in Fig. 2 and Table 2. Their source codes or binary codes are provided by the authors and the parameters are tuned finely. All algorithms are compared in terms of the same initial positions in the first frame in [27].

Implementation Details The scale, number and location of parts are influential to the tracking algorithm. Since this paper doesn't focus on part initialisation, we use a simple heuristic method to initialize the parts in first frame as Fig. 1 where the part scale is 0.618 times of the object, the number K of in-parts is two and the number M of context parts is three. HOG feature [8, 12] is used as the appearance model's feature. In Eq. (13), $C = 1$. PCT ran at about 1 fps using matlab on the desktop (Intel Core Dual CPUs, i5-2400, 3.1 GHz, 4G RAM).

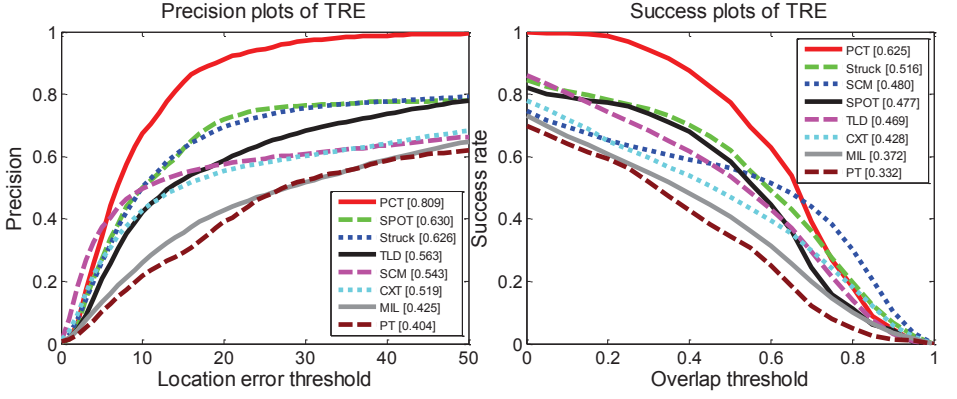

Figure 2: Plots of overall performance comparison for the ten videos following the same evaluation protocol proposed in the benchmark [27]. The proposed method ("PCT") obtain better performance in terms of precision (left) and success (right) plot

Evaluation of the update schema In order to evaluate the performance of different update thresholds and our update strategy, we conduct the experiments on *Sylvester* and *MountainBike* sequences using different update thresholds and our strategy. From the experimental results as Table 1, we can see that the update threshold affects the performance of our trackers heavily because it determines the update rate and the leverage between adaptivity and stability of the tracker. To reduce the time complexity in parameter fine-tuning and make our tracker more robust, the median filter update strategy we propose can get the competitive results from Table 1.

Table 1: Comparison results of average error center location in pixel between different update thresholds and adaptive median filter update thresholds. The bold and underlined represents the best and second respectively.

Thresholds	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	Ours
Sylvester	10.2	35.9	15.1	<u>5.9</u>	6.7	15.4	13.9	15.6	10.2	11.1	**5.5**
MountainBike	9.4	10.0	9.2	9.7	9.8	49.2	90.2	<u>7.9</u>	12.9	89.2	**7.6**

Comparison of different tracking approaches Overall, our method outperforms them consistently (shown in Fig. 2). Table 2 summarises the average center location error performance of the compared tracking algorithms over the sequences. As mentioned in Table 2, our tracker outperforms the structure SVM based trackers such as Struck [16], PT [30] and SPOT [34] in most of the sequences for more context information which are used in our tracker. Fig. 3 shows the center location error per frame with the compared trackers and why some trackers lose the target in several key frames. We can see that our tracking algorithm

obtains the best result on eight sequences, As for sequences *David* and *Suv*, the tracking performance of our tracker is slightly lower than SCM [37], it's partly due to our tracker doesn't process the scale variation or the boundary well. In addition, Fig. 4 shows the comparison on different subsets such as occlusion and illumination. It shows that our method can handle occlusion, illumination and out-of-view well. In general, the robustness of our PCT tracker lies in the context parts with spatial and temporal compositional structures which are discriminatively trained online to account for the variations.

Table 2: Average center errors between the tracking results and ground truth. The bold and underlined represent the best, the second respectively.

Methods	Trellis	Singer2	David	Suv	Lemming	Liquor	Tiger1	Tiger2	Sylvester	M.Bike
MIL	71.5	22.5	24.4	82.2	171.2	141.9	37.3	29.7	11.9	73.0
TLD	31.1	58.3	5.1	13.0	16.0	37.6	49.5	37.1	7.3	216.1
CXT	7.0	163.6	6.1	9.9	61.4	131.8	45.4	41.4	14.8	178.8
Struck	6.9	174.3	9.9	49.8	37.8	91.0	128.7	21.6	6.3	8.6
SCM	7.0	113.6	**4.3**	**4.6**	185.7	99.2	93.5	141.2	8.0	10.6
PT	8.2	173.8	46.6	35.3	135.5	94.9	33.3	47.5	6.3	9.1
SPOT	4.1	220.2	4.6	11.6	149.4	10.3	23.4	38.1	9.3	181.8
PCT	**4.0**	**11.1**	4.8	6.9	**7.3**	**5.4**	**11.8**	**18.9**	**5.5**	**7.6**

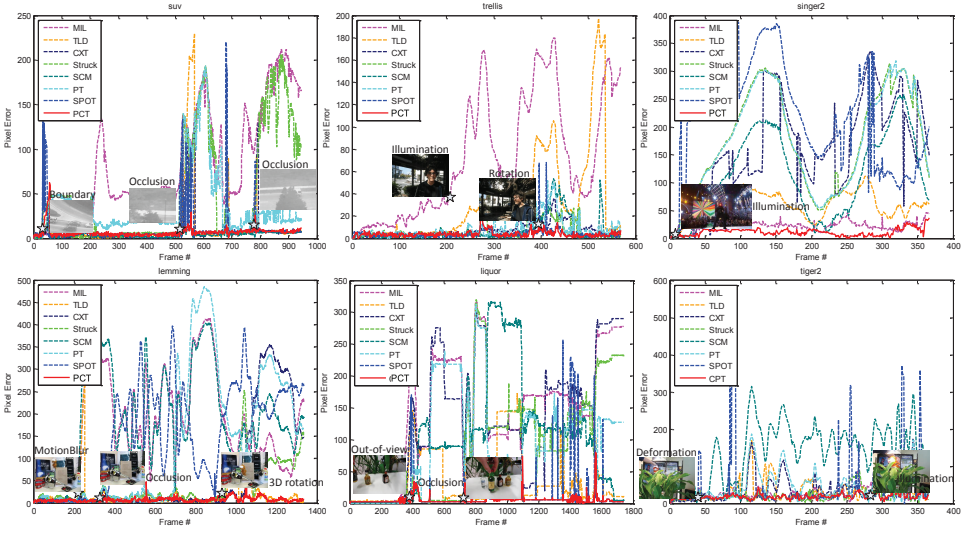

Figure 3: Comparisons on the center distance error per frame corresponding to Table 2.

4 Conclusion

This paper presents a unified context framework for simultaneously tracking and learning objects with spatial and temporal context information. Our PCT tracker consists of an appearance model, an internal relation model and an context relation model in a structured learning framework, which is robust to certain conditions of occlusion, illumination and out-of-view. To avoid the drifting problem due to update, we propose a novel update strategy to decide when to train some parts of the object model online. Experiments on challenging video sequences show that PCT tracker performs better than several state-of-the-art approaches.

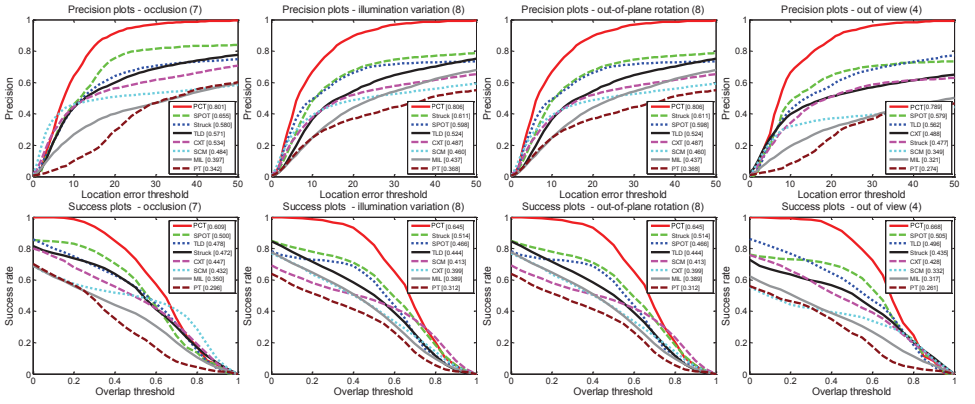

Figure 4: Several comparisons in different subsets(occlusion, illumination variation, out-of-plane rotation, deformation and out-of-view) divided based on main variation of the object to be tracked. The details of the subsets refer to [27]. The proposed method ("PCT") obtains better or comparable performance in all the subsets.

5 Acknowledgement

This work was supported by 973 Program (2010CB327905) and National Natural Science Foundation of China (61273034, 61070104, and 61202325).

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, volume 1, pages 798–805. IEEE, 2006.

[2] Y. Amit and A. Trouvé. Pop: Patchwork of parts models for object recognition. *IJCV*, 75(2):267–282, 2007.

[3] B. Babenko, M. H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, pages 983–990. IEEE, 2009.

[4] M. B. Blaschko and C.H. Lampert. Learning to localize objects with structured output regression. In *ECCV*, pages 2–15. Springer, 2008.

[5] L. Cehovin, M. Kristan, and A. Leonardis. Robust visual tracking using an adaptive coupled-layer visual model. *IEEE-TPAMI*, 35(4):941–953, 2013.

[6] W. Chang, C. Chen, and Y. Hung. Tracking by parts: A bayesian approach with component collaboration. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(2):375–388, 2009.

[7] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585, 2006.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893. IEEE, 2005.

[9] T.B. Dinh, N. Vo, and G. Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In *CVPR*, pages 1177–1184. IEEE, 2011.

[10] P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell University, 2004.

[11] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005.

[12] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010.

[13] M.A Fischler and R.A Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92, 1973.

[14] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *BMVC*, pages 47–56, 2006.

[15] H. Grabner, J. Matas, L. Van Gool, and P. Cattin. Tracking the invisible: Learning where the object might be. In *CVPR*, pages 1285–1292. IEEE, 2010.

[16] S. Hare, A. Saffari, and P.H. Torr. Struck: Structured output tracking with kernels. In *ICCV*, pages 263–270. IEEE, 2011.

[17] K. Junseok and L. Kyoung Mu. Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling. In *CVPR*, pages 1208–1215. IEEE, 2009.

[18] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE-TPAMI*, 34 (7):1409–1422, 2012.

[19] J. Kwon and K. M. Lee. Visual tracking decomposition. In *CVPR*, pages 1269–1276. IEEE, 2010.

[20] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A.V.D. Hengel. A survey of appearance models in visual object tracking. *TIST*, 4(4):58, 2013.

[21] X. Mei and H. Ling. Robust visual tracking using $\ell 1$ minimization. In *CVPR*, pages 1436–1443. IEEE, 2009.

[22] D. Ross, J. Lim, R. Lin, and M. H. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1-3):125–141, 2008.

[23] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. Prost: Parallel robust online simple tracking. In *CVPR*, pages 723–730. IEEE, 2010.

[24] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, and Y. Singer. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(9), 2005.

[25] J. Wang, L. Duan, Z. Li, J. Liu, H. Lu, and J. Jin. A robust method for tv logo tracking in video streams. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 1041–1044. IEEE, 2006.

[26] L. Wen, Z. Cai, Z. Lei, D. Yi, and S.Z. Li. Online spatio-temporal structural context learning for visual tracking. In *ECCV*, pages 716–729. Springer, 2012.

[27] Y. Wu, J. Lim, and M. H. Yang. Online object tracking: A benchmark. In *CVPR*, pages 2411–2418. IEEE, 2013.

[28] M. Yang, Y. Wu, and G. Hua. Context-aware visual tracking. *PAMI*, 31(7):1195–1209, 2009.

[29] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel. Robust tracking with weighted online structured learning. In *ECCV*, pages 158–172. Springer, 2012.

[30] 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.

[31] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *CSUR*, 38(4):13, 2006.

[32] Q. Yu, T. Dinh, and Gé. Medioni. Online tracking and reacquisition using co-trained generative and discriminative trackers. In *ECCV*, pages 678–691. Springer, 2008.

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

[34] L. Zhang and L. van der Maaten. Structure preserving object tracking. In *CVPR*, pages 1838–1845. IEEE, 2013.

[35] L. Zhang and L. van der Maaten. Preserving structure in model-free tracking. *IEEE-TPAMI*, 36(4):756–769, 2014.

[36] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via multi-task sparse learning. In *CVPR*, pages 2042–2049. IEEE, 2012.

[37] W. Zhong, H. Lu, and M.H. Yang. Robust object tracking via sparsity-based collaborative model. In *CVPR*, pages 1838–1845. IEEE, 2012.

[38] L. Zhu, Y. Chen, A. Yuille, and W. Freeman. Latent hierarchical structural learning for object detection. In *CVPR*, pages 1062–1069. IEEE, 2010.