Fast Visual Tracking via Dense Spatio-Temporal Context Learning

Kaihua Zhang[1], Lei Zhang[2], Qingshan Liu[1], David Zhang[2], and Ming-Hsuan Yang[3]

[1]S-mart Group, Nanjing University of Information Science & Technology
[2]Dept. of Computing, The Hong Kong Polytechnic University
[3]Electrical Engineering and Computer Science, University of California at Merced
zhkhua@gmail.com,cslzhang@comp.polyu.edu.hk,qsliu@nuist.edu.cn,
csdzhang@comp.polyu.edu.hk,mhyang@ucmerced.edu

Abstract. In this paper, we present a simple yet fast and robust algorithm which exploits the dense spatio-temporal context for visual tracking. Our approach formulates the spatio-temporal relationships between the object of interest and its locally dense contexts in a Bayesian framework, which models the statistical correlation between the simple low-level features (i.e., image intensity and position) from the target and its surrounding regions. The tracking problem is then posed by computing a confidence map which takes into account the prior information of the target location and thereby alleviates target location ambiguity effectively. We further propose a novel explicit scale adaptation scheme, which is able to deal with target scale variations efficiently and effectively. The Fast Fourier Transform (FFT) is adopted for fast learning and detection in this work, which only needs 4 FFT operations. Implemented in MATLAB without code optimization, the proposed tracker runs at 350 frames per second on an i7 machine. Extensive experimental results show that the proposed algorithm performs favorably against state-of-the-art methods in terms of efficiency, accuracy and robustness.

1 Introduction

Visual tracking is one of the most active research topics due to its wide range of applications such as motion analysis, activity recognition, surveillance, and human-computer interaction, to name a few [29]. The main challenge for robust visual tracking is to handle large appearance changes of the target object and the background over time due to occlusion, illumination changes, and pose variation. Numerous algorithms have been proposed with focus on effective appearance models, which are based on the target appearance [8,1,28,22,17,18,19,23,21,31] or the difference between appearances of the target and its local background [11,16,14,2,30,15]. However, if the appearances are degraded severely, there does not exist enough information extracted for robustly tracking the target, whereas its existing scene can provide useful context information to help localizing it.

In visual tracking, a local context consists of a target object and its immediate surrounding background within a determined region (see the regions inside the red rectangles in Figure 1). Most of local contexts remain unchanged as changes between two consecutive frames can be reasonably assumed to be smooth as the time interval is usually small (30 frames per second (FPS)). Therefore, there exists a strong spatio-temporal

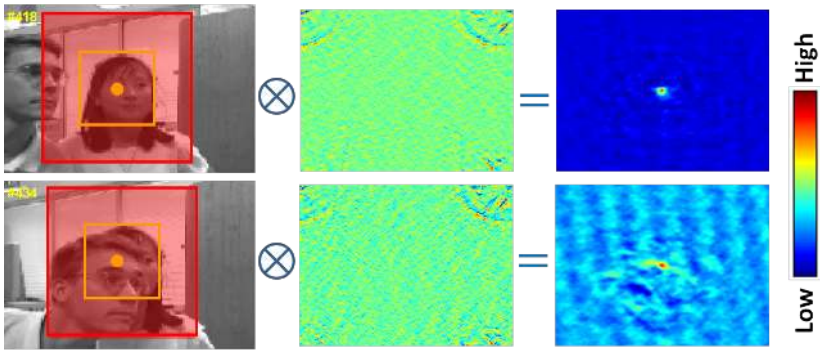
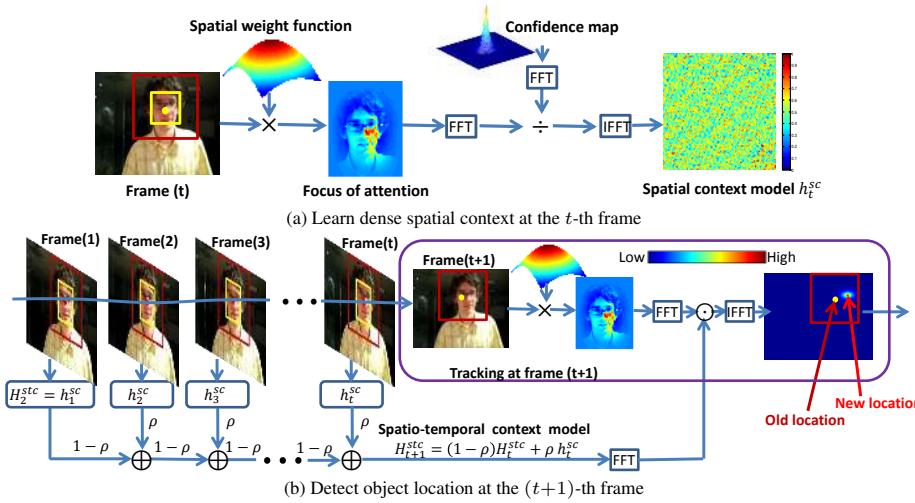


Fig. 1. The proposed method handles heavy occlusion well by learning dense spatio-temporal context information. Note that the region inside the red rectangle is the context region which includes the target and its surrounding background. Left: although the target appearance changes much due to heavy occlusion, the spatial relationship between the object center (denoted by solid yellow circle) and most of its surrounding locations in the context region is almost unchanged. Middle: the learned spatio-temporal context model (some regions have similar values which show the corresponding regions in the left frames have similar spatial relations to the target center.). Right: the learned confidence map.

relationship between the local scenes containing the object in consecutive frames. For instance, the target in Figure 1 undergoes heavy occlusion which makes the object appearance change significantly. However, the local context containing the object does not change much as the overall appearance remains similar and only a small part of the context region is occluded. Thus, the presence of local context in the current frame helps to predict the object location in the next frame. This temporally proximal information in consecutive frames is the temporal context which has been recently applied to object detection [10]. Furthermore, the spatial relation between an object and its local context provides specific information about the configuration of a scene (see middle column in Figure 1) which helps to discriminate the target from background when its appearance changes much.

2 Related Works

Most tracking algorithms can be categorized as either generative [22,17,18,19,23,21,31] or discriminative [11,16,14,2,30,15] based on their appearance models. A generative tracking method learns an appearance model to represent the target and searches for image regions with best matching scores as the results. While it is critical to construct an effective appearance model in order to handle various challenging factors in tracking, the involved computational complexity is often increased at the same time. Furthermore, generative methods discard useful information surrounding target regions that can be exploited to better separate objects from backgrounds. Discriminative methods treat tracking as a binary classification problem with local search which estimates decision



(a) Learn dense spatial context at the t-th frame

(b) Detect object location at the $(t+1)$-th frame

Fig. 2. Basic flow of our tracking algorithm. The local context regions are inside the red rectangles while the target locations are indicated by the yellow rectangles. FFT denotes the Fast Fourier Transform and IFFT is the inverse FFT.

boundary between an object image patch and the background. However, the objective of classification is to predict instance labels which is different from the goal of tracking to estimate object locations [14]. Moreover, while some efficient feature extraction techniques (e.g., integral image [11,16,14,2,30] and random projection [30]) have been proposed for visual tracking, there often exist a large number of samples from which features need to be extracted for classification, thereby entailing computationally expensive operations. Generally speaking, both generative and discriminative tracking algorithms make trade-offs between effectiveness and efficiency of an appearance model. Notwithstanding much progress has been made in recent years, it remains a challenging task to develop an efficient and robust tracking algorithm.

Recently, several methods [27,13,9,25] exploit context information to facilitate visual tracking via mining the information of regions with consistent motion correlations to the target object. In [27], a data mining method is used to extract segmented regions surrounding the object as auxiliary objects for collaborative tracking. To find consistent regions, key points surrounding the object are first extracted to help locating the object position in [13,9,25]. The SIFT or SURF descriptors are then used to represent these consistent regions. However, computationally expensive operations are required in representing and finding consistent regions. Furthermore, due to the sparsity natures of key points and auxiliary objects, some consistent regions that are useful for locating the object position may be discarded. In contrast, the proposed algorithm does not have these problems because all the local regions surrounding the object are considered as the potentially consistent regions, and the motion correlations between the objects and its local contexts in consecutive frames are learned by the spatio-temporal context model that is efficiently computed by FFT.

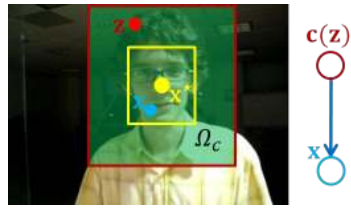


Fig. 3. Graphical model representation of spatial relationship between object and its dense local context. The dense local context region Ω_c is inside the red rectangle which includes object region surrounding by the yellow rectangle centering at the tracked result $\mathbf{x}^\star$. The context feature at location $\mathbf{z}$ is denoted by $\mathbf{c}(\mathbf{z}) = (I(\mathbf{z}), \mathbf{z})$ including a low-level appearance representation (i.e., image intensity $I(\mathbf{z})$) and location information.

In this paper, we propose a fast and robust tracking algorithm which exploits dense spatio-temporal context information. Figure 2 illustrates the basic flow of our algorithm. First, we learn a spatial context model between the target object and its local surrounding background based on their spatial correlations in a scene by solving a deconvolution problem. Next, the learned spatial context model is used to update a spatio-temporal context model for the next frame. Tracking in the next frame is formulated by computing a confidence map as a convolution problem that integrates the dense spatio-temporal context information, and the best object location can be estimated by maximizing the confidence map (See Figure 2 (b)). Finally, based on the estimated confidence map, a novel explicit scale adaptation scheme is presented, which renders an efficient and accurate tracking result.

The key contributions of the proposed algorithm are summarized as follows:

- To the best of our knowledge, it is the first work to use dense context information for visual tracking and achieves fast and robust results.
- We propose a novel explicit scale update scheme to deal with the scale variations of the target efficiently and effectively.
- The proposed algorithm is simple and fast that needs only 4 FFTs at 350 FPS in MATLAB.
- The proposed algorithm has the merits of both generative and discriminative methods. On the one hand, the context includes target and its neighbor background, thereby making our method have the merits of discriminative models. On the other hand, the context is a whole of target and background, rendering our method the merits of generative models.

3 Problem Formulation

The tracking problem is formulated by computing a confidence map which estimates the object location likelihood:

$$m(\mathbf{x}) = P(\mathbf{x}|o), \tag{1}$$

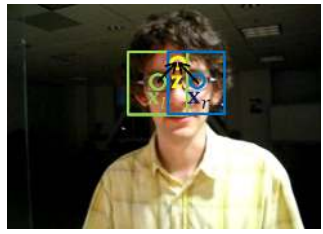


Fig. 4. Illustration of the characteristic of the non-radially symmetric function $h^{sc}(\cdot)$ in (3). Here, the left eye is the tracked target denoted by $\mathbf{x}_l$ whose context is inside the green rectangle while $\mathbf{x}_r$ represents the right eye which is a distractor with context inside the blue rectangle. Although $\mathbf{z}$ has similar distance to $\mathbf{x}_l$ and $\mathbf{x}_r$, their spatial relationships are different (i.e., $h^{sc}(\mathbf{x}_l - \mathbf{z}) \neq h^{sc}(\mathbf{x}_r - \mathbf{z})$), and this helps discriminating $\mathbf{x}_l$ from $\mathbf{x}_r$.

where $\mathbf{x} \in \mathbb{R}^2$ is an object location and o denotes the object present in the scene. (1) is equal to the posterior probability $P(o|\mathbf{x})$ because we use uniform prior $P(o)$ for the target presence for simplicity. In the following, the spatial context information is used to estimate (1) and Figure 3 shows its graphical model representation.

In Figure 3, the object location $\mathbf{x}^\star$ (i.e., coordinate of the tracked object center) is tracked. The context feature set is defined as $X^c = \{\mathbf{c}(\mathbf{z}) = (I(\mathbf{z}), \mathbf{z}) | \mathbf{z} \in \Omega_c(\mathbf{x}^\star)\}$ where $I(\mathbf{z})$ denotes image intensity at location $\mathbf{z}$ and $\Omega_c(\mathbf{x}^\star)$ is the neighborhood of location $\mathbf{x}^\star$ that is twice the size of the target object. By marginalizing the joint probability $P(\mathbf{x}, \mathbf{c}(\mathbf{z})|o)$, the object location likelihood function in (1) can be computed by

$$\begin{aligned} m(\mathbf{x}) &= P(\mathbf{x}|o) \\ &= \textstyle\sum_{\mathbf{c}(\mathbf{z}) \in X^c} P(\mathbf{x}, \mathbf{c}(\mathbf{z})|o) \\ &= \textstyle\sum_{\mathbf{c}(\mathbf{z}) \in X^c} P(\mathbf{x}|\mathbf{c}(\mathbf{z}), o) P(\mathbf{c}(\mathbf{z})|o), \end{aligned} \tag{2}$$

where the conditional probability $P(\mathbf{x}|\mathbf{c}(\mathbf{z}), o)$ models the spatial relationship between the object location and its context information which helps to resolve ambiguities when the degraded image measurements allow different interpretations, and $P(\mathbf{c}(\mathbf{z})|o)$ is a context prior probability which models appearance of the local context. The main task in this work is to learn $P(\mathbf{x}|\mathbf{c}(\mathbf{z}), o)$ as it bridges the gap between object location and its spatial context.

3.1 Spatial Context Model

The conditional probability function $P(\mathbf{x}|\mathbf{c}(\mathbf{z}), o)$ in (2) is defined as

$$P(\mathbf{x}|\mathbf{c}(\mathbf{z}), o) = h^{sc}(\mathbf{x} - \mathbf{z}), \tag{3}$$

where $h^{sc}(\mathbf{x} - \mathbf{z})$ is a function (see Figure 4 and Section 3.4) with respect to the relative *distance* and *direction* between object location $\mathbf{x}$ and its local context location $\mathbf{z}$, thereby encoding the spatial relationship between an object and its spatial context.

Note that $h^{sc}(\mathbf{x}-\mathbf{z})$ is not a radially symmetric function (i.e., $h^{sc}(\mathbf{x}-\mathbf{z}) \neq h^{sc}(|\mathbf{x}-\mathbf{z}|)$), and takes into account different spatial relationships between an object and its local contexts. This helps to resolve ambiguities when similar objects appear in close proximity. For example, when a method tracks an eye based only on appearance (denoted by $\mathbf{x}_l$) in the *davidindoor* sequence shown in Figure 4, the tracker may be easily distracted to the right one (denoted by $\mathbf{x}_r$) because both eyes and their surrounding backgrounds have similar appearances (when the object moves fast and the search region is large). However, in the proposed method, while the locations of both eyes are at similar distances to location $\mathbf{z}$, their relative locations to $\mathbf{z}$ are different, resulting in different spatial relationships, i.e., $h^{sc}(\mathbf{x}_l-\mathbf{z}) \neq h^{sc}(\mathbf{x}_r-\mathbf{z})$. That is, the non-radially symmetric function h^{sc} helps to resolve ambiguities effectively.

3.2 Context Prior Model

In (2), the context prior probability is related to the context appearance which is simply modeled by

$$P(\mathbf{c}(\mathbf{z})|o) = I(\mathbf{z})w_\sigma(\mathbf{z}-\mathbf{x}^\star), \tag{4}$$

where $I(\cdot)$ is image intensity that represents appearance of context and $w_\sigma(\cdot)$ is a Gaussian weighted function defined by

$$w_\sigma(\mathbf{z}-\mathbf{x}^\star) = ae^{-\frac{|\mathbf{z}-\mathbf{x}^\star|^2}{\sigma^2}}, \tag{5}$$

where a is a normalization constant that restricts $P(\mathbf{c}(\mathbf{z})|o)$ in (4) to range from 0 to 1 that satisfies the definition of probability and σ is a scale parameter.

In (4), it models focus of attention that is motivated by the biological visual system which concentrates on certain image regions requiring detailed analysis [24]. The closer the context location $\mathbf{z}$ is to the currently tracked target location $\mathbf{x}^\star$, the more important it is to predict the object location in the coming frame, and larger weight should be set (please refer to Figure 2 (a)). Different from our algorithm that uses a spatially weighted function to indicate the importance of context at different locations, there exist other methods [3,26] in which spatial sampling techniques are used to focus more detailed contexts at the locations near the object center (i.e., the closer the location is to the object center, the more context locations are sampled).

3.3 Confidence Map

The confidence map of an object location is modeled as

$$m(\mathbf{x}) = P(\mathbf{x}|o) = be^{-|\frac{\mathbf{x}-\mathbf{x}^\star}{\alpha}|^\beta}, \tag{6}$$

where b is a normalization constant, α is a scale parameter and β is a shape parameter (please refer to Figure 5).

The confidence map $m(\mathbf{x})$ in (6) takes into account the prior information of the target location which is able to handle the location ambiguity problem effectively. The object location ambiguity problem often occurs in visual tracking which adversely affects tracking performance. In [2], a multiple instance learning technique is adopted to

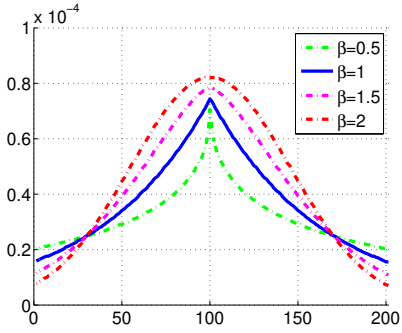


Fig. 5. Illustration of 1-D cross section of the confidence map $m(\mathbf{x})$ in (6) with different parameters β. Here, the object location $\mathbf{x}^\star = (100, 100)$.

handle the location ambiguity problem with favorable tracking results. The closer the location is to the currently tracked position, the larger probability that the ambiguity occurs with (e.g., predicted object locations that differ by a few pixels are all plausible solutions and thereby cause ambiguities). In our method, we resolve the location ambiguity problem by choosing a proper shape parameter β. As illustrated in Figure 5, a large β (e.g., $\beta = 2$) results in an oversmoothing effect for the confidence value at locations near to the object center, failing to effectively reduce location ambiguities. On the other hand, a small β (e.g., $\beta = 0.5$) yields a sharp peak near the object center, and activates much fewer positions when learning the spatial context model. This in turn may lead to overfitting in searching for the object location in the coming frame. We find that robust results can be obtained when $\beta = 1$ in our experiments.

3.4 Fast Learning Spatial Context Model

Based on the confidence map function (6) and the context prior model (4), our objective is to learn the spatial context model (3). Putting (6), (4) and (3) together, we formulate (2) as

$$\begin{aligned} m(\mathbf{x}) &= be^{-|\frac{\mathbf{x}-\mathbf{x}^\star}{\alpha}|^\beta} \\ &= \textstyle\sum_{\mathbf{z}\in\Omega_c(\mathbf{x}^\star)} h^{sc}(\mathbf{x}-\mathbf{z})I(\mathbf{z})w_\sigma(\mathbf{z}-\mathbf{x}^\star) \\ &= h^{sc}(\mathbf{x}) \otimes (I(\mathbf{x})w_\sigma(\mathbf{x}-\mathbf{x}^\star)), \end{aligned} \tag{7}$$

where $\otimes$ denotes the convolution operator.

We note (7) can be transformed to the frequency domain in which the Fast Fourier Transform (FFT) algorithm [20] can be used for fast convolution. That is,

$$\mathcal{F}(be^{-|\frac{\mathbf{x}-\mathbf{x}^\star}{\alpha}|^\beta}) = \mathcal{F}(h^{sc}(\mathbf{x})) \odot \mathcal{F}(I(\mathbf{x})w_\sigma(\mathbf{x}-\mathbf{x}^\star)), \tag{8}$$

where $\mathcal{F}$ denotes the FFT function and $\odot$ is the element-wise product. Therefore, we have

$$h^{sc}(\mathbf{x}) = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(be^{-|\frac{\mathbf{x}-\mathbf{x}^\star}{\alpha}|^\beta})}{\mathcal{F}(I(\mathbf{x})w_\sigma(\mathbf{x}-\mathbf{x}^\star))}\right), \tag{9}$$

where $\mathcal{F}^{-1}$ denotes the inverse FFT function. The spatial context model h^{sc} learns the relatively spatial relations between different pixels (please refer to Figure 4 and Section 3.1) in a Bayesian framework.

4 Proposed Tracking Algorithm

Figure 2 shows the basic flow of our algorithm. The tracking problem is formulated as a detection task. We assume that the target location in the first frame has been initialized manually or by some object detection algorithms. At the t-th frame, we learn the spatial context model $h_t^{sc}(\mathbf{x})$ (9), which is used to update the spatio-temporal context model $H_{t+1}^{stc}(\mathbf{x})$ (12) to reduce noise introduced by target appearance variations. H_{t+1}^{stc} is then applied to detect the object location in the $(t+1)$-th frame. When the $(t+1)$-th frame arrives, we crop out the local context region $\Omega_c(\mathbf{x}_t^\star)$ based on the tracked location $\mathbf{x}_t^\star$ at the t-th frame and construct the corresponding context feature set $X_{t+1}^c = \{\mathbf{c}(\mathbf{z}) = (I_{t+1}(\mathbf{z}), \mathbf{z}) | \mathbf{z} \in \Omega_c(\mathbf{x}_t^\star)\}$. The object location $\mathbf{x}_{t+1}^\star$ in the $(t+1)$-th frame is determined by maximizing the new confidence map

$$\mathbf{x}_{t+1}^\star = \arg\max_{\mathbf{x} \in \Omega_c(\mathbf{x}_t^\star)} m_{t+1}(\mathbf{x}), \tag{10}$$

where $m_{t+1}(\mathbf{x})$ is represented as

$$m_{t+1}(\mathbf{x}) = H_{t+1}^{stc}(\mathbf{x}) \otimes (I_{t+1}(\mathbf{x}) w_{\sigma_t}(\mathbf{x} - \mathbf{x}_t^\star)), \tag{11}$$

which is deduced from (7) and can use FFT for fast convolution. Here, H_{t+1}^{stc} derives from the spatial context model h_t^{sc} with a low-pass temporal filtering processing and hence is able to reduce the noise introduced by abrupt appearance changes of I_{t+1}.

4.1 Update of Spatio-Temporal Context

The spatio-temporal context model is updated by

$$H_{t+1}^{stc} = (1-\rho) H_t^{stc} + \rho h_t^{sc}, \tag{12}$$

where ρ is a learning parameter and h_t^{sc} is the spatial context model computed by (9) at the t-th frame. We note (12) is a temporal filtering procedure which can be easily observed in frequency domain

$$H_\omega^{stc} = F_\omega h_\omega^{sc}, \tag{13}$$

where $H_\omega^{stc} \triangleq \int H_t^{stc} e^{-j\omega t} dt$ is the temporal Fourier transform of H_t^{stc} and similar to h_ω^{sc}. The temporal filter F_ω is formulated as

$$F_\omega = \frac{\rho}{e^{j\omega} - (1-\rho)}, \tag{14}$$

where j denotes imaginary unit. It is easy to validate that F_ω in (14) is a low-pass filter [20]. Therefore, our spatio-temporal context model is able to effectively filter out image noise introduced by appearance variations, leading to more stable results.

4.2 Update of Scale

According to (11), the target location in the current frame is found by maximizing the confidence map derived from the weighted context region surrounding the previous target location. However, the scale of the target often changes over time. Therefore, the scale parameter σ in the weight function w_σ (5) should be updated accordingly. We propose the scale update scheme as

$$\begin{cases} s'_t = \sqrt{\frac{m_t(\mathbf{x}^\star_t)}{m_{t-1}(\mathbf{x}^\star_{t-1})}}, \\ \overline{s}_t = \frac{1}{n}\sum_{i=1}^{n} s'_{t-i}, \\ s_{t+1} = (1-\lambda)s_t + \lambda \overline{s}_t, \\ \sigma_{t+1} = s_t \sigma_t, \end{cases} \tag{15}$$

where $m_t(\cdot)$ is the confidence map at the t-th frame that is computed by (11), and s'_t is the estimated scale between two consecutive frames. To avoid oversensitive adaptation and to reduce noise introduced by estimation error, the estimated target scale s_{t+1} is obtained through filtering in which $\overline{s}_t$ is the average of the estimated scales from n consecutive frames, and $\lambda > 0$ is a fixed filter parameter (similar to ρ in (12)). The derivation details of (15) can be found at `http://www4.comp.polyu.edu.hk/~cslzhang/STC/STC.htm`.

4.3 Analysis and Discussion

We note that the low computational complexity is one prime characteristic of the proposed algorithm. In learning the spatial context model (9), since the confidence map (11) and the scale updating (15) can be pre-computed only once before tracking, there are only 4 FFT operations involved for processing one frame. The computational complexity for computing each FFT is only $\mathcal{O}(MN\log(MN))$ for the local context region of $M \times N$ pixels, thereby resulting in a fast method (350 FPS in MATLAB on an i7 machine). More importantly, the proposed algorithm achieves robust results as discussed bellow.

Difference with related work. It should be noted that the proposed dense spatio-temporal context tracking algorithm is significantly different from recently proposed approaches that use FFT for efficient computation [5,4,15].

In [5,4], the formulations are based on correlation filters that are directly obtained by classic signal processing algorithms. At each frame, correlation filters are trained using a large number of samples, and then combined to find the most correlated position in the next frame. In [15], the filters proposed by [5,4] are kernelized and used to achieve more stable results. The proposed algorithm is significantly different from [5,4,15] in several aspects. First, our algorithm models the spatio-temporal relationships between the object and its local contexts which is motivated by the human visual system that exploits context to help resolving ambiguities in complex scenes efficiently and effectively. Second, our algorithm focuses on the regions which require detailed analysis, thereby effectively reducing the adverse effects of background clutters and leading to more robust results. Third, our algorithm handles the object location ambiguity problem using the confidence map with a proper prior distribution, thereby achieving more

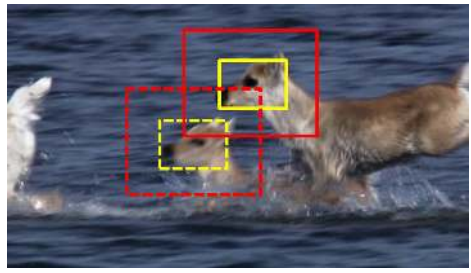


Fig. 6. Illustration of why the proposed model is equipped to handle distractor. The target inside the yellow dotted rectangle is the distractor. The different surrounding contexts can well discriminate target from distactor.

stable and accurate performance for visual tracking. Finally, our algorithm solves the scale adaptation problem while the other FFT-based tracking methods [5,4,15] only track objects with a fixed scale and achieve less accurate results than our method.

Robustness to occlusion and distractor. As shown in Figure 1, the proposed algorithm handles heavy occlusion well as most of context regions are not occluded which have similar relative spatial relations (see middle column of Figure 1) to the target center. This helps to determine the target center. Figure 6 illustrates that our method is robust to distractor (i.e., the bottom left object). If tracking the target only based on its appearance information, the tracker will be distracted to the top right one because of their similar appearances. Although the distractor has similar appearance to the target, most of their surrounding contexts have different appearances which are useful to discriminate target from distractor.

5 Experiments

We evaluate the proposed spatio-temporal context (STC) tracking algorithm using 18 video sequences with challenging factors including heavy occlusion, drastic illumination changes, pose and scale variation, non-rigid deformation, background cluster and motion blur. We compare the proposed STC tracker with 18 state-of-the-art methods in which the context tracker [9] and the FFT-based trackers [4,15] (i.e., ConT, MOS and CST in Table 1) are included. For other context-based tracking methods [27,13,25], their source codes are not available for evaluation and the implementations require some technical details and parameters not discussed therein. The parameters of the proposed algorithm are *fixed* for all the experiments. For other trackers, we use either the original source or binary codes provided in which parameters of each tracker are tuned for best results. The 18 trackers we compare with are: scale mean-shift (SMS) tracker [7], fragment tracker (Frag) [1], semi-supervised Boosting tracker (SSB) [12], local orderless tracker (LOT) [21], incremental visual tracking (IVT) method [22], online AdaBoost tracker (OAB) [11], multiple instance learning tracker (MIL) [2], visual tracking decomposition method (VTD) [17], L1 tracker (L1T) [19], tracking-learning-detection

Table 1. Success rate (SR)(%). **Red** fonts indicate the best performance while the **blue** fonts indicate the second best ones. The total number of evaluated frames is 7,591.

Sequence	SMS	Frag	SSB	LOT	IVT	OAB	MIL	VTD	L1T	TLD	DF	MTT	Struck	ConT	MOS	CT	CST	LGT	**STC**
animal	13	3	51	15	4	17	83	**96**	6	37	6	87	93	58	3	92	**94**	7	**94**
bird	33	64	13	5	78	**94**	10	9	44	42	**94**	10	48	26	11	8	47	**89**	65
bolt	58	41	18	89	15	1	92	3	2	1	2	2	8	6	25	**94**	39	74	**98**
cliffbar	5	24	24	26	47	66	71	53	24	62	26	55	44	43	6	**95**	93	81	**98**
chasing	72	77	62	20	82	71	65	70	72	76	70	95	85	53	61	79	**96**	95	**97**
car4	10	34	22	1	**97**	30	37	35	94	88	26	22	96	90	28	36	44	33	**98**
car11	1	1	19	32	54	14	48	25	46	67	78	59	18	47	**85**	36	48	16	**86**
cokecan	1	3	38	4	3	53	18	7	16	17	13	85	**94**	20	2	30	86	18	**87**
downhill	81	89	53	6	87	82	33	**98**	66	13	94	54	87	71	28	82	72	73	**99**
dollar	55	41	38	40	21	16	46	39	39	39	**100**	39	**100**	**100**	**89**	87	**100**	**100**	**100**
davidindoor	6	1	36	20	7	24	30	38	18	**96**	64	94	71	82	43	46	2	95	**100**
girl	7	70	49	91	64	68	28	68	56	79	59	71	**97**	74	3	27	43	51	**98**
jumping	2	34	81	22	**100**	82	**100**	**87**	13	76	12	**100**	18	**100**	6	**100**	**100**	5	**100**
mountainbike	14	13	82	71	**100**	**99**	18	**100**	61	26	35	**100**	98	25	55	89	**100**	74	**100**
ski	22	5	65	55	16	58	33	6	5	36	6	9	**76**	43	1	60	1	**71**	68
shaking	2	25	30	14	1	39	83	**98**	3	15	84	2	48	12	4	84	36	48	**96**
sylvester	70	34	67	61	45	66	77	33	40	**89**	33	68	81	84	6	77	84	**85**	78
woman	52	27	30	16	21	18	21	35	8	31	93	19	**96**	28	2	19	21	66	**100**
Average SR	35	35	45	35	49	49	52	49	40	62	53	59	**75**	62	26	62	60	68	**94**

(TLD) method [16], distribution field tracker (DF) [23], multi-task tracker (MTT) [31], structured output tracker (Struck) [14], context tracker (ConT) [9], minimum output sum of square error (MOS) tracker [4], compressive tracker (CT) [30], circulant structure tracker (CST) [15] and local-global tracker (LGT) [6]. For the trackers involving randomness, we repeat the experiments 10 times on each sequence and report the averaged results. Implemented in MATLAB, our tracker runs at 350 FPS on an i7 3.40 GHz machine with 8 GB RAM. The MATLAB source codes are available at `http://www4.comp.polyu.edu.hk/~cslzhang/STC/STC.htm`.

5.1 Experimental Setup

The size of context region is initially set to twice the size of the target object. The parameter σ_t of (15) is initially set to $\sigma_1 = \frac{s_h+s_w}{2}$, where s_h and s_w are height and width of the initial tracking rectangle, respectively. The parameters of the map function are set to $\alpha = 2.25$ and $\beta = 1$. The learning parameter $\rho = 0.075$. We note that as illustrated by Figure 2 (b), the weights from other frames are smaller than that from the current observation no matter how small ρ is set. Thus, the current observation is the most important one. The scale parameter s_t is initialized to $s_1 = 1$, and the learning parameter $\lambda = 0.25$. The number of frames for updating the scale is set to $n = 5$. To reduce effects of illumination change, each intensity value in the context region is normalized by subtracting the average intensity of that region. Then, the intensity in the context region multiplies a Hamming window to reduce the frequency effect of image boundary when using FFT [20,5].

5.2 Experimental Results

We use two evaluation criteria to quantitatively evaluate the 19 trackers: the center location error (CLE) and success rate (SR), both computed based on the manually labeled ground truth results of each frame. The score of success rate is defined as $score = \frac{area(R_t \bigcap R_g)}{area(R_t \bigcup R_g)}$, where R_t is a tracked bounding box and R_g is the ground truth bounding box, and the result of one frame is considered as a success if $score > 0.5$.

Table 2. Center location error (CLE)(in pixels) and average frame per second (FPS). **Red** fonts indicate the best performance while the **blue** fonts indicate the second best ones. The total number of evaluated frames is 7,591.

Sequence	SMS	Frag	SSB	LOT	IVT	OAB	MIL	VTD	L1T	TLD	DF	MTT	Struck	ConT	MOS	CT	CST	LGT	**STC**
animal	78	100	25	70	146	62	32	17	122	125	252	17	19	76	281	18	**16**	166	**15**
bird	25	13	101	99	13	**9**	140	57	60	145	12	156	21	139	159	79	20	**11**	15
bolt	42	43	102	**9**	65	227	**9**	177	261	286	277	293	149	126	223	10	210	12	**8**
cliffbar	41	34	56	36	37	33	13	30	40	70	52	25	46	49	104	**6**	**6**	10	**5**
chasing	13	9	44	32	6	9	13	23	9	47	31	**5**	6	16	68	10	**5**	6	**4**
car4	144	56	104	177	14	109	63	127	16	13	92	158	**9**	**11**	117	63	44	47	**11**
car11	86	117	11	30	**7**	11	8	20	8	12	**6**	8	9	8	8	9	8	16	**7**
cokecan	60	70	15	46	64	11	18	68	40	29	30	10	**7**	36	53	16	9	32	**6**
downhill	14	11	102	226	22	12	117	**9**	35	255	10	77	10	62	116	12	129	12	**8**
dollar	55	56	66	66	23	28	23	65	65	72	**3**	71	18	5	12	20	5	4	**2**
davidindoor	176	103	45	100	281	43	33	40	86	13	27	**11**	20	22	78	28	149	12	**8**
girl	130	26	50	12	36	22	34	41	51	23	27	23	**8**	34	126	39	43	35	**9**
jumping	63	30	11	43	**4**	11	**4**	17	45	13	73	7	42	**4**	155	6	**3**	89	**4**
mountainbike	135	209	11	24	**5**	11	208	7	74	213	155	7	8	149	16	11	**5**	12	**6**
ski	91	134	**10**	12	51	11	15	179	161	222	147	33	**8**	78	386	11	237	13	12
shaking	224	55	133	90	134	22	11	**5**	72	232	11	115	23	191	194	11	21	33	**10**
sylvester	15	47	14	23	138	12	9	66	49	**8**	56	18	9	13	65	9	**7**	11	11
woman	49	118	86	131	112	120	119	110	148	108	12	169	**4**	55	176	122	160	23	**5**
Average CLE	79	63	54	70	84	43	43	58	62	78	52	80	**19**	42	103	29	54	22	**8**
Average FPS	12	7	11	0.7	33	22	38	5	1	28	13	1	20	15	**200**	90	120	8	**350**

Table 1 and Table 2 show the quantitative results in which the proposed STC tracker achieves the best or second best performance in most sequences both in terms of center location error and success rate. Furthermore, the proposed tracker is the most efficient (350 FPS on average) algorithm among all evaluated methods. Although the CST [15] and MOS [4] methods also use FFT for fast computation, the CST method performs time-consuming kernel operations and the MOS tracker computes several correlation filters in each frame, making these two approaches less efficient than the proposed algorithm. Furthermore, both CST and MOS methods only track target with fixed scale, which achieve less accurate results than the proposed method with scale adaptation. Figure 7 shows some tracking results of different trackers. For presentation clarity, we only show the results of the top 7 trackers in terms of average success rates. More results can be found in the paper website `http://www4.comp.polyu.edu.hk/~cslzhang/STC/STC.htm`.

Illumination, scale and pose variation. There are large illumination variations in the evaluated sequences. The appearance of the target object in the *car4* sequence changes significantly due to the cast shadows and ambient lights (see #200, #250 in the *car4* sequence shown in Figure 7). Only the models of IVT, L1T, Struck and STC adapt to these illumination variations well. Likewise, only the VTD and our STC methods perform favorably on the *shaking* sequence because the object appearance changes drastically due to the stage lights and sudden pose variations. The *davidindoor* sequence contain gradual pose and scale variations as well as illumination changes. Note that most reported results using this sequence are only on subsets of the available frames, i.e., not from the very beginning of the *davidindoor* video when the target face is in nearly complete darkness. In this work, the full sequence is used to better evaluate the performance of all algorithms. Only the proposed algorithm is able to achieve favorable tracking results on this sequence in terms of both accuracy and success rate. This can be attributed to the use of dense spatio-temporal context information which facilitates filtering out noisy observations (as discussed in Section 4.1), enabling the proposed STC algorithm

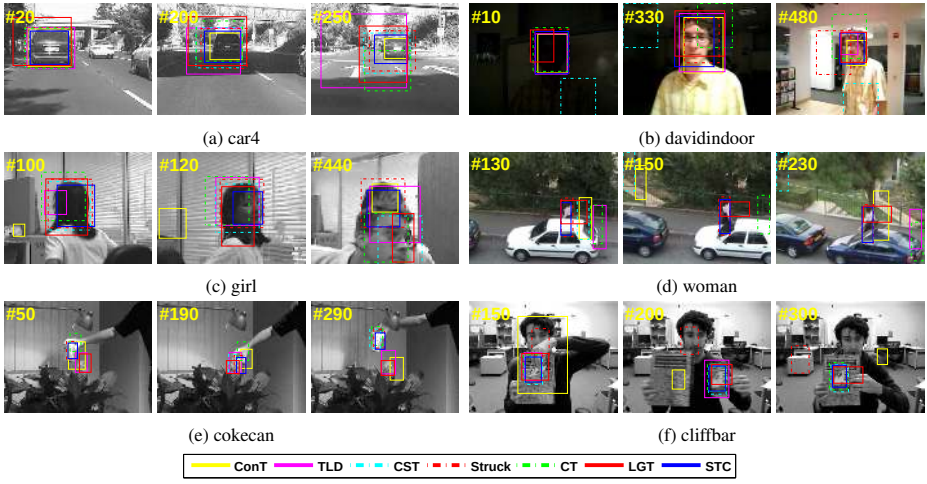


Fig. 7. Screenshots of tracking results. More results and videos can be found in the supplementary material.

to relocate the target when object appearance changes drastically due to illumination, scale and pose variations.

Occlusion, rotation, and pose variation. The target objects in the *woman*, *girl* and *bird* sequences are partially occluded at times. The object in the *girl* sequence also undergoes in-plane rotation (See $\#100, \#120$ of the *girl* sequence in Figure 7) which makes the tracking tasks difficult. Only the proposed algorithm is able to track the objects successfully in most frames of this sequence. The *woman* sequence has non-rigid deformation and heavy occlusion (see $\#130, \#150, \#230$ of the *woman* sequence in Figure 7) at the same time. All the other trackers fail to successfully track the object except for the Struck and the proposed STC algorithms. As most of the local contexts surrounding the target objects are not occluded in these sequences, such information facilitates the proposed algorithm relocating the object even they are almost fully occluded (as discussed in Figure 1).

Background clutter and abrupt motion. In the *animal*, *cokecan* and *cliffbar* sequences, the target objects undergo fast movements in the cluttered backgrounds. The target object in the *chasing* sequence undergoes abrupt motion with 360 degree out-of-plane rotation, and the proposed algorithm achieves the best performance in terms of both success rate and center location error. The *cokecan* video contains a specular object with in-plane rotation and heavy occlusion, which makes this tracking task difficult. Only the Struck and the proposed STC methods are able to successfully track most of the frames. In the *cliffbar* sequence, the texture in the background is very similar to that of the target object. Most trackers drift to background except for CT, CST, LGT and STC (see $\#300$ of the *cliffbar* sequence in Figure 7). Although the target and its local background have very similar texture, their spatial relationships and appearances of local contexts are different which are used by the proposed algorithm when learning

a confidence map (as discussed in Section 4.3). Hence, the proposed STC algorithm is able to separate the target object from the background based on the dense spatio-temporal context.

6 Conclusion

In this paper, we presented a simple yet fast and robust algorithm which exploits dense spatio-temporal context information for visual tracking. Two local context models (i.e., spatial context and spatio-temporal context models) were proposed which are robust to appearance variations introduced by occlusion, illumination changes, and pose variations. An explicit scale adaptation scheme was proposed which is able to adapt target scale variations effectively. The Fast Fourier Transform algorithm was used in both online learning and detection, resulting in an efficient tracking method that runs at 350 frames per second with MATLAB implementation. Numerous experiments with state-of-the-art algorithms on challenging sequences demonstrated that the proposed algorithm achieves favorable results in terms of accuracy, robustness, and speed.

Acknowledgements

Kaihua Zhang is supported in part by the NUIST Scientific Research Foundation under Grant S8113049001. Lei Zhang is supported in part by the Hong Kong Polytechnic University ICRG Grant (G-YK79). Ming-Hsuan Yang is supported in part by the NSF CAREER Grant #1149783 and NSF IIS Grant #1152576. Qingshan Liu is supported in part by NSFC under Grant 61272223 and NSF of Jiangsu Province under Grant BK2012045.

References

1. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: CVPR. pp. 798–805 (2006)
2. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. PAMI 33(8), 1619–1632 (2011)
3. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. PAMI 24(4), 509–522 (2002)
4. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: CVPR. pp. 2544–2550 (2010)
5. Bolme, D.S., Draper, B.A., Beveridge, J.R.: Average of synthetic exact filters. In: CVPR. pp. 2105–2112 (2009)
6. Cehovin, L., Kristan, M., Leonardis, A.: Robust visual tracking using an adaptive coupled-layer visual model. PAMI 35(4), 941–953 (2013)
7. Collins, R.T.: Mean-shift blob tracking through scale space. In: CVPR. vol. 2, pp. II–234 (2003)
8. Collins, R.T., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. PAMI 27(10), 1631–1643 (2005)
9. Dinh, T.B., Vo, N., Medioni, G.: Context tracker: Exploring supporters and distracters in unconstrained environments. In: CVPR. pp. 1177–1184 (2011)

10. Divvala, S.K., Hoiem, D., Hays, J.H., Efros, A.A., Hebert, M.: An empirical study of context in object detection. In: CVPR. pp. 1271–1278 (2009)
11. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via on-line boosting. In: BMVC. pp. 47–56 (2006)
12. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: ECCV. pp. 234–247 (2008)
13. Grabner, H., Matas, J., Van Gool, L., Cattin, P.: Tracking the invisible: Learning where the object might be. In: CVPR. pp. 1285–1292 (2010)
14. Hare, S., Saffari, A., Torr, P.H.: Struck: Structured output tracking with kernels. In: ICCV. pp. 263–270 (2011)
15. Henriques, J., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: ECCV. pp. 702–715 (2012)
16. Kalal, Z., Matas, J., Mikolajczyk, K.: Pn learning: Bootstrapping binary classifiers by structural constraints. In: CVPR. pp. 49–56 (2010)
17. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: CVPR. pp. 1269–1276 (2010)
18. Kwon, J., Lee, K.M.: Tracking by sampling trackers. In: ICCV. pp. 1195–1202 (2011)
19. Mei, X., Ling, H.: Robust visual tracking and vehicle classification via sparse representation. PAMI 33(11), 2259–2272 (2011)
20. Oppenheim, A.V., Willsky, A.S., Nawab, S.H.: Signals and systems, vol. 2. Prentice-Hall Englewood Cliffs, NJ (1983)
21. Oron, S., Bar-Hillel, A., Levi, D., Avidan, S.: Locally orderless tracking. In: CVPR. pp. 1940–1947 (2012)
22. Ross, D., Lim, J., Lin, R., Yang, M.H.: Incremental learning for robust visual tracking. IJCV 77(1), 125–141 (2008)
23. Sevilla-Lara, L., Learned-Miller, E.: Distribution fields for tracking. In: CVPR. pp. 1910–1917 (2012)
24. Torralba, A.: Contextual priming for object detection. IJCV 53(2), 169–191 (2003)
25. Wen, L., Cai, Z., Lei, Z., Yi, D., Li, S.: Online spatio-temporal structure context learning for visual tracking. In: ECCV. pp. 716–729 (2012)
26. Wolf, L., Bileschi, S.: A critical view of context. IJCV 69(2), 251–261 (2006)
27. Yang, M., Wu, Y., Hua, G.: Context-aware visual tracking. PAMI 31(7), 1195–1209 (2009)
28. Yang, M., Yuan, J., Wu, Y.: Spatial selection for attentional visual tracking. In: CVPR. pp. 1–8 (2007)
29. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys 38(4) (2006)
30. Zhang, K., Zhang, L., Yang, M.H.: Real-time compressive tracking. In: ECCV. pp. 864–877 (2012)
31. Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Robust visual tracking via multi-task sparse learning. In: CVPR. pp. 2042–2049 (2012)