Object tracking based on adaptive updating of a spatial-temporal context model

Wanli Feng[1, 2], **Yigang Cen**[1,2], **Xianyou Zeng**[1,2], **Zhetao Li**[3], **Ming Zeng**[4], **Viacheslav Voronin**[5]

[1] School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China, 100044
[e-mail: 15120327@bjtu.edu.cn, ygcen@bjtu.edu.cn, 14112057@bjtu.edu.cn]
[2] Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing, China, 100044
[3] The college of information Engineering, Xiangtan University, Xiangtan, Hunan Province, China, 411105
[e-mail: liztchina@hotmail.com]
[4] School of Automation Science and Engineering, South China University of Technology, Guangzhou, China, 510640
[e-mail: Zengm@scut.edu.cn]
[5] Department of Radio-electronic systems, Don State Technical University, Rostov-on-Don, Russia, 346500
[e-mail: voronin_sl@mail.ru]
*Corresponding author: Zhetao Li

Received March 17, 2017; revised June 5, 2017; revised July 18, 2017; accepted July 24, 2017; published November 30, 2017

Abstract

Recently, a tracking algorithm called the spatial-temporal context model has been proposed to locate a target by using the contextual information around the target. This model has achieved excellent results when the target undergoes slight occlusion and appearance changes. However, the target location in the current frame is based on the location in the previous frame, which will lead to failure in the presence of fast motion because of the lack of a prediction mechanism. In addition, the spatial context model is updated frame by frame, which will undoubtedly result in drift once the target is occluded continuously. This paper proposes two improvements to solve the above two problems: First, four possible positions of the target in the current frame are predicted based on the displacement between the previous two frames, and then, we calculate four confidence maps at these four positions; the target position is located at the position that corresponds to the maximum value. Second, we propose a target reliability criterion and design an adaptive threshold to regulate the updating speed of the model. Specifically, we stop updating the model when the reliability is lower than the threshold. Experimental results show that the proposed algorithm achieves better tracking results than traditional STC and other algorithms.

***Keywords**: spatial-temporal context, confidence map, position prediction, adaptive threshold*

This work is supported by the National Natural Science Foundation of China (no. 61572067, 61379115); International (Regional) Project Cooperation and Exchanges of National Nature Science Foundation of China (no. 61611530710); Beijing Municipal Natural Science Foundation (no. 4162050); The Fundamental Research Funds for the Central Universities (no. 2017JBZ108); Natural Science Foundation of Guangdong Province (no. 2016A030313708).

https://doi.org/10.3837/tiis.2017.11.015 **ISSN : 1976-7277**

1. Introduction

Object tracking is a fundamental problem in computer vision due to its wide range of applications, such as motion recognition, video surveillance and human computer interaction [1-2]. The appearance model is a basic element in object tracking [3] because the object could suffer from large appearance changes caused by unstable illumination, occlusion and deformations of itself. In recent years, many algorithms have been proposed to establish a robust appearance model [4-10].

Appearance-based tracking algorithms can be divided into two branches, called generative and discriminative. Generative algorithms [4-7] pose tracking as matchings of the appearance model, and the unsatisfactory speeds in realistic situations limit their applications in the real world, while they perform well in a stationary environment. Discriminative algorithms [8-12] aim to train a classifier to separate the target from the background or obtain a confidence map. Then, the location with the maximum value is selected as the target. However, these algorithms discard the spatial relations between the target and the background, which is useful for classification. Recently, a novel spatial-temporal-context (STC) algorithm [13] was proposed to address the above problem. STC exploits the spatial relations between the target and the surrounding background inside a certain area called a local context. The spatial context model has been demonstrated to be robust to short-time occlusions because of subtle changes in the context between two consecutive frames. However, under the assumption that there is no mutation of the target location, STC could undergo failure when the target undergoes fast motion because of the lack of a prediction mechanism of the location. Moreover, the tracker is prone to drift in the presence of long-term occlusion, target deformation and rotation, which tend to contaminate the spatial model as it is updated frame by frame.

To overcome the above shortcomings, we propose to predict four potential locations in the next frame based on the displacement between the two previous frames, followed by calculating four confidence maps of the target location. Similarly, the location with the maximum value of these four maps is considered to be the new location. Additionally, we formulate the reliability of the object location through synthetic consideration of the peak-to-sidelobe ratio (*PSR*) [14] and a smoothness constraint of the confidence map (*SCCM*) [15] between the two consecutive frames. Additionally, an adaptive threshold is proposed to control the updating speed of the spatial model, i.e., we stop updating the spatial model once the reliability is smaller than the threshold.

The remainder of this paper is organized as follows: Section 2 gives a brief introduction to the STC tracker. The proposed algorithm is presented in detail in Section 3. Section 4 performs a series of experiments as well as the analysis. We conclude this paper in Section 5.

2. STC tracking algorithm

The main idea of the STC algorithm is to establish the spatial-temporal relationships between the target and the context. This task lies in calculating a confidence map of the target location $c(x)=p(x|o)$. Here, $c(x)$ can be decomposed easily with the Bayes formula:

$$\begin{aligned} c(x) &= \sum_{c(z)\in X^c} P(x,c(z)\,|\,o) \\ &= \sum_{c(z)\in X^c} P(x\,|\,c(z),o)P(c(z)\,|\,o) \end{aligned} \qquad (1)$$

Both $x,z\in R^2$ are position coordinates, and o refers to the existing target. The context feature set is $X^c=\{c(z)=(I(z),z)|z\in\Omega_c(x^*)\}$, where $\Omega_c(x^*)$ is the local context area that surrounds the target center x^*. $I(z)$ is the gray value at location z. Evidently, $P(x|c(z),o)$ is defined as follows:

$$P(x\,|\,c(z),o)=h^{sc}(x-z) \tag{2}$$

Equation (2) models the spatial relations between the target and its context area, where $h(\cdot)$ means the expected model, and the superscript '*sc*' is taken from the first letters of 'spatial-context'. To be specific, $P(x|c(z),o)$ means the probability that the center of the target is indeed at x, where the context feature $c(z)$ is positioned. The non-radial symmetry of this function is propitious for improving the tracker's accuracy when there exists another object that is similar to the target [13]. $P(c(z)|o)$ is the prior probability that characterizes the significance of the context z in predicting the target location. $P(x|c(z),o)$ plays a significant part in the STC algorithm.

2.1 Context priori probability and confidence map

As a metric of the context's importance for predicting the target location, the context prior probability is naturally defined as

$$P(c(z)\,|\,o)=I(z)w_\sigma(z-x^*), \tag{3}$$

where $I(\cdot)$ is the gray value, and $w_\sigma(z)=ae^{-|z|^2/\sigma^2}$ is a Gaussian weighting function with the normalization constant a and scale parameter σ. This definition is consistent with the visual characteristics of our human eyes, since a context with a closer distance to the current target center and a larger gray value would contribute more to the prediction of the target location.

The confidence map is formulated as

$$c(x)=p(x\,|\,o)=be^{-\left|(x-x^*)/\alpha\right|^\beta}, \tag{4}$$

where b is a normalization constant, α is the scale parameter, and β regulates the shape of this function with 1 in STC.

2.2 Tracking procedure

Based on the confidence map and context prior probability, we aim to learn the spatial context model. According to Eq. (2), Eq. (3) and Eq. (4), we reformulate Eq. (1) as

$$\begin{aligned} c(x)&=be^{-((x-x^*)/\alpha)^\beta}\\ &=\sum_{z\in\Omega_c(x^*)}h^{sc}(x-z)I(z)w_\sigma(z-x^*)\\ &=h^{sc}(x)\otimes(I(x)w_\sigma(x-x^*)), \end{aligned} \tag{5}$$

where $\otimes$ is a convolution operator. Note that Eq. (5) can be transformed to the frequency domain to accelerate the computational process:

$$h^{sc}(x)=F^{-1}\left(\frac{F(be^{-((x-x^*)/\alpha)^\beta})}{F(I(x)w_\sigma(x-x^*))}\right), \tag{6}$$

where F and F^{-1} denote the FT and IFT operators, respectively. Without loss of generality, after the target position is determined in the frame t, we can learn the spatial context model by Eq. (6), which can be used to update the spatial context model:

$$H_{t+1}^{stc} = (1-\rho)H_t^{stc} + \rho h_t^{sc}, \tag{7}$$

where the superscript '*stc*' is taken from the first letters of 'spatial-temporal-context', and ρ is the learning rate. The context feature set is constructed when the frame $t+1$ arrives, followed by calculating a confidence map (note that Eq. (7) is applicable only when $t \geq 2$. When $t=1$, we initialize the spatio-temporal context model as the spatial context model, i.e., $H_2^{stc} = h_1^{stc}$, and h_1^{stc} is computed by Eq. (6). Since H_{t+1}^{stc} is used to calculate a confidence map as in Eq. (8) and the target location in frame 1 is given, Eq. (8) is used only when the subscript of $c_{t+1}(x)$ is greater than or equals to 2, i.e., $t+1 \geq 2$. The initial spatio-temporal context model is H_2^{stc}):

$$c_{t+1}(x) = H_{t+1}^{stc}(x) \otimes (I_{t+1}(x) w_{\sigma_t}(x - x_t^*)) \tag{8}$$

The location with the maximum value is selected as the target location in the frame $t+1$:

$$x_{t+1}^* = \arg\max_{x \in \Omega_c(x_t^*)} c_{t+1}(x) \tag{9}$$

3. Proposed tracking algorithm

3.1 Location prediction

It can be seen from Eq. (8) that the context feature set X_{t+1}^c of the frame $t+1$ is constructed based on the previous location in frame *t* in the traditional STC. There will be a large error between X_{t+1}^c and the real context feature set of the frame $t+1$ when the target moves rapidly. To solve this problem, we first predict 4 possible locations of the target, as illustrated in **Fig. 1** (x_{t+1}^1, x_{t+1}^2, x_{t+1}^3, and x_{t+1}^4 are 4 predicted locations of the target in the frame $t+1$. Note that the target in the frame $t+1$ is calculated based on only the target location at the frame *t*, i.e., x_t in the original STC, and thus, we set x_{t+1}^1 to be the same as x_t to maintain close ties with the original STC).

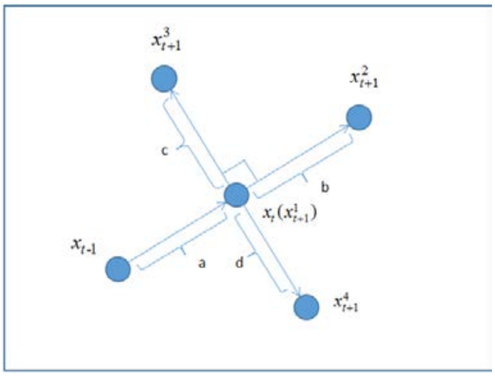

Fig. 1. Sketched diagram of the location prediction, where *a* is the displacement of the previous two frames, and *b*=*c*=*d*=*a*

Four context feature sets are constructed based on these four locations as well as the confidence maps. Then, the object location x_{t+1}^* in the $t+1$ frame is determined by maximizing these 4 confidence maps:

$$x_{t+1}^{*} = \arg\max(\max(c_{t+1}^{i}(x))),\ 1 \le i \le 4\ , \tag{10}$$

where $c_{t+1}^{i}(x)$ is represented as

$$c_{t+1}^{i}(x) = H_{t+1}^{stc}(x) \otimes (I_{t+1}(x) w_{\sigma_t}(x - x_{t+1}^{i})), \quad 1 \le i \le 4,$$

where x_{t+1}^{i} means the *i*-th predicted location in the frame $t+1$, as illustrated in **Fig. 1**.

3.2 Adaptive updating of the model

The fixed learning rate to update the spatial context model in Eq. (7) can easily introduce a spatial context model from an unreliable region with the target occluded or the appearance changing, as caused by deformation or rotation, which would have a detrimental impact on the subsequent frames. As demonstrated in [14], the *PSR* in the confidence map represents the sharpness of the peak, which can be used to measure the reliability of the tracking result, i.e., a higher *PSR* means a more reliable tracking result. In addition, the *SCCM* proposed in [15] can be used to judge whether the target is occluded or not during the tracking. *SCCM* is defined as

$$SCCM_t = \left\| \hat{f}^{t} - \hat{f}^{t-1} \oplus \Delta \right\|_2^2 , \tag{11}$$

where $\hat{f}^{t}$ and $\hat{f}^{t-1}$ denote the confidence maps of the frames *t* and $t-1$**,** respectively. Here, $\oplus$ means a shift operation of the confidence map, and Δ denotes the corresponding shift of the maximum value in the confidence maps from the frame $t-1$ to *t*.

Based on the above analysis, we propose a criterion for judging the reliability of the confidence map: $rel_t = PSR_t / SCCM_t$. Theoretically speaking, occlusion or rotation can bring a low *PSR* and high *SCCM*, and thus, *rel* can respond to occlusion and target rotation in a sensitive manner. To verify the validity of *rel* in real scenes, we select three typical fragments under the conditions of target rotation, occlusion and normal state, where the *rel* value of each frame is visualized as follows:

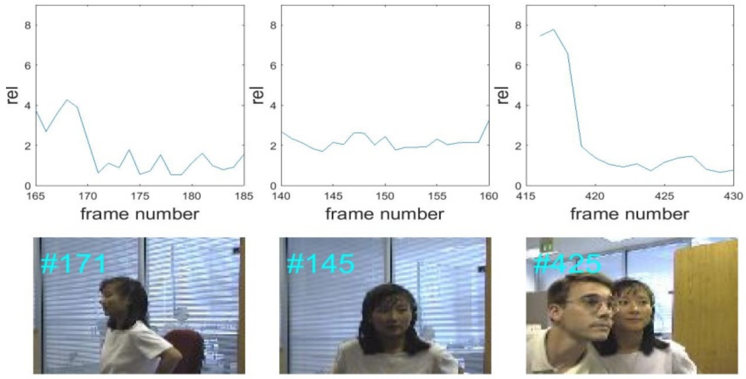

Fig. 2. Reliability of some frames

Apparently, the target state can be reflected by the *rel* value (see decreasing *rel* on #171, #425 accompanied by rotation and occlusion).

Ideally, we should stop updating the spatial context model when the reliability is not sufficiently high, such as in #171 and #425, as shown in **Fig. 2**. Therefore, we propose to set a threshold to control whether the model can be updated normally or not. The new mechanism for updating the spatial context model can be formulated as

$$H_{t+1}^{stc} = \begin{cases} (1-\rho)H_t^{stc} + \rho h_t^{sc}, & if \quad rel_t > threshold_t \\ H_t^{stc} & , \ else \end{cases} \tag{12}$$

It is important to set an appropriate threshold. Obviously, a higher threshold can slow down the process of model updating and vice versa. An adaptive threshold based on the reliability discussed above is proposed as follows:

$$threshold_t = \frac{std(rel(t-m:t-1))}{mean(rel(t-m/2:t-1))} \tag{13}$$

The numerator denotes the standard deviation of the reliability values of the *m* frames prior to the current frame, and the denominator denotes the mean values of the reliability values of the *m*/*2* frames prior to the current frame.

As is shown in **Fig. 3**, there are 4 possible trends of the *rel* value in *m* consecutive frames prior to the current frame:

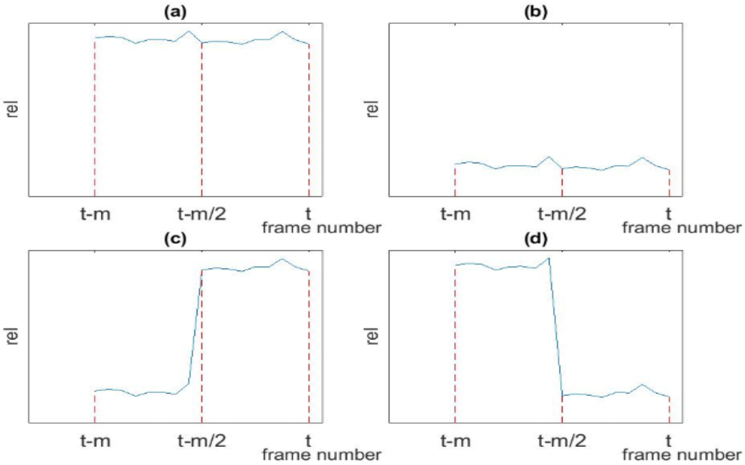

Fig. 3. Four trends of the *rel* values

(a) There are no occlusion or appearance changes of the target, which can be reflected by the stability of the *rel* values at a high level, and we must update the spatial model as usual. A small threshold is beneficial in such a case according to the relationship of the threshold and updating speed, as mentioned above. We note that the standard deviation of the *rel* value of these *m* frames appears to be small, while the mean value of *rel* of the latter *m*/*2* frame appears to be relatively large, and thus, a small threshold that is consistent with our expectation can be calculated by Eq. (13).

(b) There could exist occlusion or appearance changes of the target, which can be reflected by the stability of the *rel* values at a low level, and we must stop updating the model. A large threshold is beneficial in such a case. We note that the standard deviation of the *rel* value of these *m* frames appears to be small as well as the mean value. Thus, a relatively large threshold that is consistent with our expectation can be calculated by Eq. (13).

(c) Occlusion could be gradually disappearing from the target or the target's appearance begins to gradually stabilize downward, which can be reflected by the growing values of *rel*. A small threshold is beneficial to adapt to the changing scene. We note that the standard deviation of the ***rel*** values of these *m* frames appears to be large, while the mean value of the latter *m*/*2* frames becomes larger, also. Thus, a small threshold that is consistent with our expectation can be calculated by Eq. (13).

(d) There could be something that occludes the target gradually, which can be reflected by the declining values of *rel*, as shown in **Fig. 2**. A large threshold is beneficial to prevent the model from being polluted. We note that the standard deviation of the ***rel*** values of these *m* frames appears to be large, while the mean value of ***rel*** of the latter *m*/*2* frames appears to be relatively small. Thus, a large threshold that is consistent with our expectation can be calculated by Eq. (13).

3.3 Framework of the proposed algorithm

According to the above discussion, the tracking algorithm based on adaptive updating of the spatial-temporal context model (AU-STC) can be summarized as follows:

Algorithm 1. AU-STC algorithm

```
Inputs: Target location $x^{*}_{t-1}$, spatial-temporal model $H^{stc}_{t}$ .
    1. Predict four target locations and calculate the corresponding four confidence maps;
    2. The location with the maximum value is selected as the target location $x^{*}_{t}$ ;
    3. Learn the spatial context model $h^{sc}_{t}$ by Eq. (6);
    4. Calculate the reliability of the confidence map $rel_t$ and the threshold $threshold_t$ using Eq. (13);
    5. Update the spatial-temporal-model $H^{stc}_{t+1}$ by Eq. (12);
Outputs: Target location $x^{*}_{t}$ , spatial-temporal model $H^{stc}_{t+1}$, then take them as inputs.
```

4. Experiments and analysis

We evaluate the proposed algorithm using 30 (20 from [16], 6 from CAVIAR and 4 from VIVID datasets) public video sequences with challenging factors, including heavy occlusion, fast motion, non-rigid deformation and motion blur. We compare the proposed algorithm with traditional STC, compressive tracking (CT) [9] and Multiple Instance Learning (MIL) [12]. The parameters of the proposed algorithm are fixed for all the experiments. For other trackers, we use the original source code provided, in which the parameters of each tracker are tuned to obtain the best results. All our experiments are performed by using MATLAB R2015a on a 3.2 GHz Intel Core i5 PC with 4 GB RAM.

4.1 Parameter settings

The size of the context region is initially set to twice the target size. The parameters of the map function are set to $\alpha = 2.25$ and $\beta = 1$. The learning parameter $\rho = 0.075$, which is the same as in the traditional STC. The value of *m* in Eq. (13) is set to 5.

4.2 Qualitative evaluation

Some tracking results of different trackers are shown in **Figs. 4** to **6**. In these figures, red, blue, green and yellow denote the tracking results of our algorithm, MIL, STC and CT, respectively.

4.2.1 Long-term occlusion

The target in FaceOcc2 (**Fig. 4**(a)) experienced a long-term occlusion and a posture change before #707. Moreover, the target is occluded from #707 for a long time, and STC has lost the target (see #750). In Suv (**Fig. 4**(b)), MIL, STC and CT all failed after the target is occluded (see #571), whereas our algorithm can track the target until the last frame. Note that the haar feature used in CT easily falls on the occluded area, and the spatial context between the target and occluding object in STC could lead to cumulative failure. In Girl (**Fig. 4**(c)), MIL and CT lose the target after target rotation (see #80), and STC starts to track the occluding object unrealistically (see #470). In general, it can be seen that the context-based method (STC and our algorithm) has a natural advantage over unreliable haar features in such a low-resolution image. The target in Caviar1 (**Fig. 4**(d)) experienced a long-term occlusion and background clutter from #100. Moreover, the target is occluded from #169 for the second time, and we can see an unsatisfactory result in MIL and CT but not in our algorithm and STC.

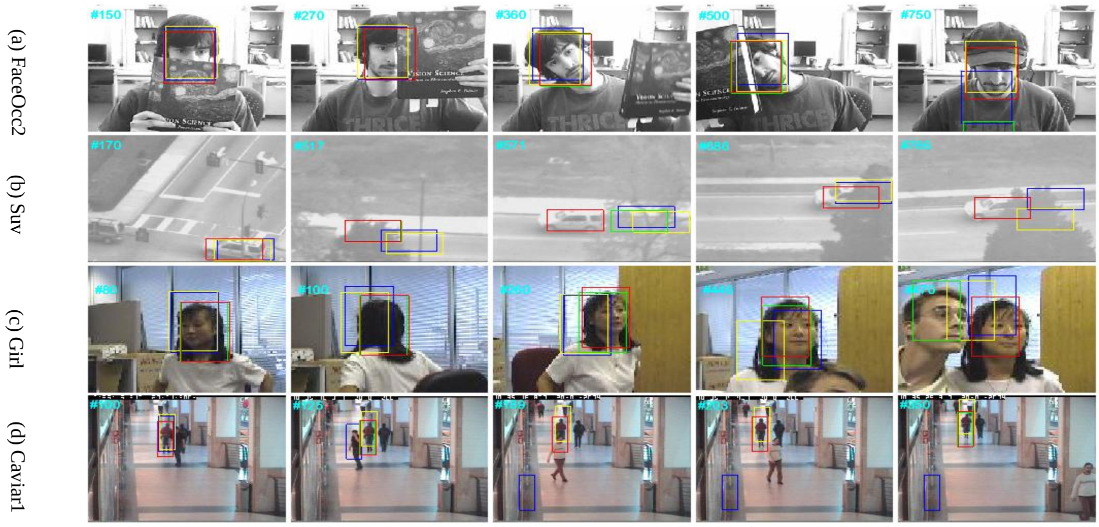

Fig. 4. Screenshots of the tracking results with long-term occlusion. In FaceOcc2, the yellow rectangle superposes the blue rectangle in #150, #270. The red rectangle superposes the green rectangle in #360, #500. In Suv, the red rectangle superposes the green rectangle in #170, #517, and the green rectangle has been lost in #686, #786. In Girl, the red rectangle superposes the green rectangle in #80, #100. In Caviar1, the red rectangle superposes the green rectangle in these five frames.

4.2.2 Deformation and rotation

In Skater (**Fig. 5**(a)), the target undergoes a 90 degree rotation and deformation 6 times (see #40, #62, #82, #100, #112, #125 in the original sequence). It can be seen that CT performs better than STC and our algorithm at the 6th rotation. This result occurs because the high frequency of rotation can be easily caught by the combination of 50 features in CT, while the context-based methods treat each context equally so that they cannot adapt to dramatic rotation well. However, we see that the adaptive threshold used in our algorithm obtains better results compared with STC. In shop (**Fig. 5**(b)), the target shares the same color with the background and rotates at the same time (see the 2th and 4th image); only our algorithm works well, and the other three trackers all failed from the beginning. Obviously, MIL, STC and our algorithm all perform well in the sequences MountainBike and Bolt (**Fig. 5**(c)), as the object rotates at a low frequency except for CT.

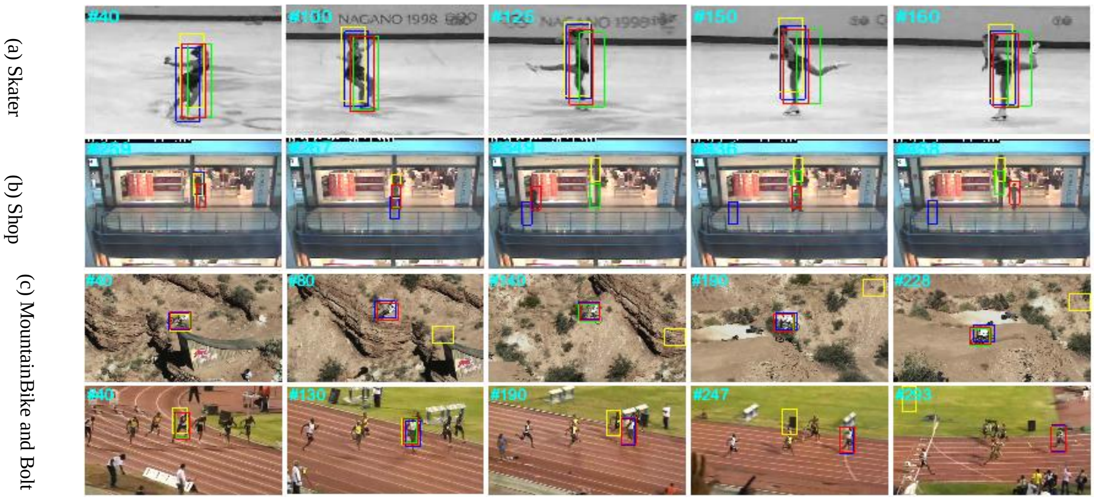

Fig. 5. Screenshots of the tracking results with deformation and rotation. In MountainBike, the red rectangle superposes the green rectangle in #40, #80, #190. In Bolt, the red rectangle superposes the green rectangle in #190, #247, #293.

4.2.3 Fast motion and Motion Blur

In BlurFace (**Fig. 6**(a)), with the target moving rapidly through the whole process, MIL and CT begin to lose the target from #150, and the same occurs with STC from #390. However, our algorithm performs accurately for the entire duration. In Dog (**Fig. 6**(b)), with the target moving rapidly through the whole process, CT and our algorithm both perform more accurately than STC and MIL. In Fish (**Fig. 6**(c)), the target undergoes fast jitter of the camera and illumination changes, yet little changes in the spatial context prompts similar performances of STC and our algorithm. Nevertheless, CT loses the target (see #60) because the haar feature is sensitive to illumination. In Deer (**Fig. 6**(d)), the target starts to move quickly at the beginning, and the other three trackers all fail for the reason that a fixed radius for generating candidate samples cannot adapt to the large displacement. The mechanism for location prediction of our algorithm will remedy this error.

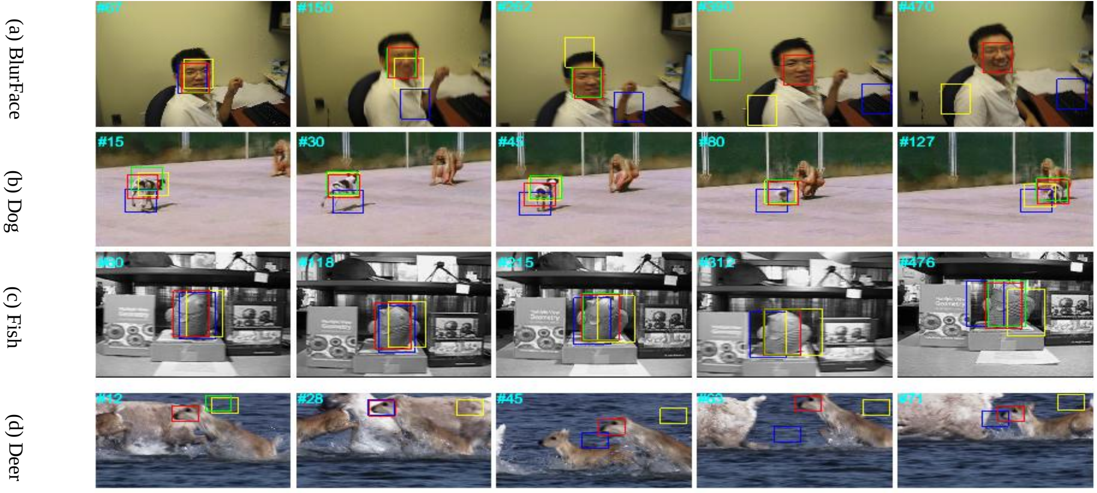

Fig. 6. Screenshots of the tracking results with fast motion and motion blur. In BlurFace, the red rectangle superposes the green rectangle in frames #67, #150, #262. In Fish, the red rectangle superposes the green rectangle in #60, #118, #215 and #312. In Deer, the green rectangle has been lost in #28, #45, #63 and #71.

4.3 Quantitative evaluation

Based on the above discussion, three evaluation criteria are employed to quantitatively assess the performances of the trackers: the center location error (CLE), overlap rate (OR), and success rate (SR) [16]. The CLE is defined as the Euclidean distance between the tracked bounding box and the ground truth bounding box. The OR is defined as $area(R_t \cap R_g)/area(R_t \cup R_g)$, where R_t is a tracked bounding box, and R_g is the ground truth bounding box in a frame. SR is used to illustrate the percentage of frames whose OR is higher than a threshold (usually 0.5) in a sequence.

CLE, OR, SR of these Four trackers for the 30 sequences are shown in **Tables 1-3** according to the attributes defined in Section 4.2 (the bold font in the tables denotes the best tracker for the corresponding sequence, and the total number of evaluated frames is 18379). The reasons for our algorithm's superiority can been seen as follows: first the prediction mechanism of the target location enlarges the search area of the target in the next frame, which is beneficial in tracking a fast-moving target. Second, when the target is occluded or rotated, a large threshold that limits the updating speed of the spatial context model is obtained based on the target reliability, which prevents the introduction of an unreliable model to contaminate a normal model.

Table 1. Center location error (CLE) (in pixels) and average frame per second (FPS)

Attributes	Sequences	MIL	CT	STC	Ours
Long-term occlusion	FaceOcc2	21	15	22	**9**
	Suv	73	69	173	**4**
	Girl	16	17	9	**2**
	Caviar1	96	22	4	**3**
	Caviar2	74	59	**4**	**4**
	RedTeam	10	13	9	**8**
	Coke	48	36	17	**15**
	FaceOcc1	38	**23**	198	32
	David3	30	90	10	**9**
Deformation and rotation	Skater	**11**	18	23	14
	Shop	88	73	52	**5**
	MountainBike	**7**	212	**7**	**7**
	Bolt2	9	76	**6**	**6**
	Egtest01	9	7	6	**5**
	Egtest02	16	17	12	**11**
	Egtest03	230	226	35	**12**
	Coupon	20	18	**2**	**2**
	MeetWalkTogethe	60	21	13	**11**
	David2	16	79	4	**2**
	TwoEnterShop	23	135	**10**	11
Fast motion and Motion Blur	BlurFace	153	108	114	**5**
	Dog	31	17	21	**16**
	Fish	17	28	**3**	**3**
	Deer	60	232	400	**7**

	BlurCar1	132	118	120	**8**
	ClifBar	32	**11**	41	15
Illumination Variation	CarDark	45	120	**2**	**2**
	Crowds	5	402	5	**4**
	Trellis	80	59	40	**20**
	LeftBox	12	11	11	**9**
	Average CLE	48	77	45	**9**
	Average FPS	9	30	**39**	33

Table 2. Overlap rate (OR)

Attributes	Sequences	MIL	CT	STC	Ours
Long-term occlusion	FaceOcc2	0.64	0.7	0.71	**0.79**
	Suv	0.25	0.2	0.52	**0.84**
	Girl	0.35	0.2	0.56	**0.72**
	Caviar1	0.23	0.5	**0.71**	**0.71**
	Caviar2	0.22	0.3	0.58	**0.59**
	RedTeam	0.61	0.6	**0.73**	0.71
	Coke	0.25	0.2	0.50	**0.53**
	FaceOcc1	0.54	**0.7**	0.21	0.65
	David3	0.50	0.2	0.66	**0.70**
Deformation and rotation	Skater	0.61	0.5	0.52	**0.63**
	Shop	0.10	0.0	0.10	**0.62**
	MountainBike	0.70	0.1	**0.72**	**0.72**
	Bolt2	0.62	0.2	**0.68**	**0.68**
	Egtest01	0.54	0.5	0.58	**0.64**
	Egtest02	0.48	0.5	**0.59**	**0.59**
	Egtest03	0.25	0.2	0.40	**0.59**
	Coupon	0.60	0.6	**0.92**	**0.92**
	MeetWalkTogether	0.09	0.3	0.44	**0.51**
	David2	0.38	0.0	0.74	**0.81**
	TwoEnterShop	0.24	0.1	0.48	**0.55**
Fast motion and Motion Blur	BlurFace	0.17	0.2	0.52	**0.85**
	Dog	0.2	0.5	0.52	**0.59**
	Fish	0.53	0.4	**0.83**	0.82
	Deer	0.36	0.0	0.04	**0.75**
	BlurCar1	0.11	0.1	0.50	**0.80**
	ClifBar	0.23	0.5	0.27	**0.53**
Illumination Variation	CarDark	0.15	0.0	0.79	**0.80**
	Crowds	0.70	0.0	0.63	**0.73**
	Trellis	0.26	0.3	0.49	**0.57**
	LeftBox	0.54	0.6	0.57	**0.63**
	Average OR	0.38	0.3	0.55	**0.68**

Table 3. Success rate (SR) (%)

Attributes	**Sequences**	**MIL**	**CT**	**STC**	**Ours**
Long-term occlusion	FaceOcc2	87	96	89	**100**
	Suv	14	26	57	**98**
	Girl	23	13	75	**97**
	Caviar1	30	37	**98**	**98**
	Caviar2	27	38	**61**	**61**
	RedTeam	30	47	**60**	**60**
	Coke	8	15	48	**56**
	FaceOcc1	61	**97**	25	85
	David3	64	30	88	**91**
Deformation and rotation	Skater	82	70	57	**88**
	Shop	6	3	12	**84**
	MountainBike	**100**	17	96	97
	Bolt2	88	26	**100**	**100**
	Egtest01	47	43	**62**	56
	Egtest02	42	**65**	60	60
	Egtest03	20	27	30	**52**
	Coupon	84	87	**100**	**100**
	MeetWalkTogether	3	13	35	**50**
	David2	30	0	99	**100**
	TwoEnterShop	6	5	37	**53**
Fast motion and Motion Blur	BlurFace	20	24	63	**100**
	Dog	21	**65**	62	**65**
	Fish	53	23	**100**	**100**
	Deer	48	4	4	**100**
	BlurCar1	5	11	58	**99**
	ClifBar	4	42	31	**61**
Illumination Variation	CarDark	10	0	99	**100**
	Crowds	90	0	81	**99**
	Trellis	26	32	65	**74**
	LeftBox	60	83	69	**86**
	Average SR	39	34	64	**82**

5. Conclusions

In this paper, we propose to predict four possible target locations where four confidence maps are calculated. In addition, a reliability criterion of the target location with a threshold is introduced for updating the spatial model adaptively. We analyze four trends of reliability values, based on which an adaptive threshold is determined to control whether the spatial context model is updated or not. These two improvements result in a better performance in terms of fast motion, occlusion, deformation and so on, and experimental results show that the SR of our algorithm is 28% higher than the traditional STC.

Except the above advantages, we find that our algorithm cannot adapt to drastic changing target size well. Moreover, when a target disappears and then re-appears after a long period of time, our algorithm will perform disappointingly because of a lack of re-detection mechanism. Our future work will focus on introducting a robust scale adaptation scheme for tracking an object in varying sizes. In addition, we will explore efficient detection modules for persistent tracking.

References

[1] Y. Wu, J. Lim and M. H. Yang, "Object Tracking Benchmark," *IEEE Trans. of Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834-1848, September, 2015. Article (CrossRef Link)

[2] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara and A. Dehghan, "Visual Tracking: An Experimental Survey," *IEEE Trans. of Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442-1468, July, 2014. Article (CrossRef Link)

[3] A. D. Jepson, D. J. Fleet and T. F. Elmaraghi, "Robust Online Appearance Models for Visual Tracking," *IEEE Trans. of Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296-1311, Ocotober, 2003. Article (CrossRef Link)

[4] D. A. Ross, J. Lim, R. S. Lin and M. H. Yang, "Incremental Learning for Robust Visual Tracking," *International Journal of Computer Vision*, vol. 77, no. 1, pp. 125-141, May, 2008. Article (CrossRef Link)

[5] Y. Tang, Y. Li, S. Ge, J. Luo and H. Ren, "Distortion invariant joint-feature for visual tracking in catadioptric omnidirectional vision," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 2418-2423, May 26-30, 2015. Article (CrossRef Link)

[6] X. Mei and H. Ling, "Robust Visual Tracking and Vehicle Classification via Sparse Representation," *IEEE Trans. of Software Engineering*, vol. 33, no. 11, pp. 2259-2272, November, 2011. Article (CrossRef Link)

[7] S. Zhang, H. Yao, H. Zhou, X. Sun and S. Liu, "Robust visual tracking based on online learning sparse representation," *Neurocomputing*, vol. 100, no. 1, pp. 31-40, January, 2013. Article (CrossRef Link)

[8] Z. Kalal, K. Mikolajczyk and J. Matas, "Tracking-Learning-Detection," *IEEE Trans. of Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409-1422, July, 2012. Article (CrossRef Link)

[9] K. Zhang, L. Zhang and M. H. Yang, "Fast Compressive Tracking," *IEEE Trans. of Pattern* Analysis and Machine Intelligence, vol. 36, no. 10, pp. 2002-2015, Ocotober, 2014. Article (CrossRef Link)

[10] D. S. Bolme, J. R. Beveridge, B. A. Draper and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2544-2550, June 13-18, 2010. Article (CrossRef Link)

[11] J. F. Henriques, R. Caseiro, P. Martins and J. Batista. "High-Speed Tracking with Kernelized Correlation Filters," *IEEE Trans. of Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583-596, March, 2015. Article (CrossRef Link)

[12] B. Babenko, M. H. Yang and S. Belongie, "Visual tracking with online Multiple Instance Learning," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 983-990, June 20-25, 2009. Article (CrossRef Link)

[13] K. Zhang, L. Zhang, Q. Liu, D. Zhang and M. H. Yang, "Fast Visual Tracking via Dense Spatio-temporal Context Learning," in *Proc. of the European Conf. on Computer Vision*, pp. 127-141, September 6-12, 2014. Article (CrossRef Link)

[14] Y. Tang, M. Lao, F. Lin and D. Wu, "Structural spatio-temporal transform for robust visual tracking," in *Proc. of IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 1105-1109, March 20-25, 2016. Article (CrossRef Link)

[15] T. Liu, G. Wang and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proc. of IEEE Conf. Computer Vision and Pattern Recognition*, pp. 4902-4912, June 7-12, 2015. Article (CrossRef Link)

[16] Y. Wu, J. Lim and M. H. Yang, "Online Object Tracking: A Benchmark," in *Proc. of IEEE Conf. Computer Vision and Pattern Recognition*, pp. 2411-2418, June 23-28, 2013. Article (CrossRef Link)

Wanli Feng received the BS degree from Northwest Normal University in 2015. He has been pursuing his master's degree in signal and information processing at the Beijing Jiaotong University since 2015. His current research interests include visual tracking and machine learning.

Yigang Cen received the Ph.D. degree in control science engineering from Huazhong University of Science and Technology in 2006. He is currently a professor and a supervisor of doctor students with Beijing Jiaotong University, Beijing, China. His research interests include machine vision, compressed sensing, sparse representation, low-rank matrix reconstruction, and wavelet construction theory.

Xianyou Zeng received the M.S degree from South China Normal University. He is currently a Ph.D. student in the College of Computer and Information Technology at Beijing Jiaotong University. His research interests include image processing and visual tracking.

Zhetao Li received the Ph.D. degree in Computer Application Technology from Hunan University in 2010. He is currently a professor of College of Information Engineering, Xiangtan University. His current researches focus on signal processing and wireless communication.

Ming Zeng received the Ph.D. degree in Control Science & Engineering from the South China University of Technology in 2008. He is currently an assistant professor of the South China University of Technology. His research interests include artificial intelligence, wireless sensor networks, internet of things, etc.

Viacheslav Voronin received the diploma in Candidate of Science in Radio engineering, Southern Federal University. He is currently a leader of Computer Vision Group at research institute of Digital signal processing and computer vision with Don State Technical University, Rostov-on-Don, Russia. His research interests include machine vision, image processing, etc.