

## STC Tracking Algorithm Based on Kalman Filter

Panqiao Chen<sup>1,a</sup>, Mengzhao Yang<sup>2,b</sup>

<sup>1</sup>School of Automation, Beijing Institute of Technology, Beijing, China

<sup>2</sup>School of Automation, Beijing Institute of Technology, Beijing, China

<sup>a</sup>chenpanqiao89@163.com, <sup>b</sup>rreal34@126.com

**Keywords:** objecting tracking; occlusion; STC; Kalman Filter

**Abstract.** During object tracking, Fast tracking via Spatio-Temporal Context Learning which combines temporal correlation among sequential frames and spatial correlation between object and background can solve the problem of semi-occlusion, but not full-occlusion. Kalman Filter makes use of the predictive value and measurement to calculate the optimal state. This paper aims at solving the full-occlusion problems by combining the algorithm and Kalman Filter together. Experiments show that the improved STC can solve occlusion problems effectively.

### Introduction

As for image tracking, the feature and position of target could not change so much between two successive frames, so temporal correlation can be considered when the tracking algorithm is designed. There is a certain relationship between the target and surrounding background. When the object's appearance changes, algorithms can still be valid with the relationship as mentioned. The tracking algorithm including temporal and spatial correlations is called fast tracking via Spatial-Temporal Context Learning, STC for short<sup>[1-3]</sup>. Occlusion can't be avoided when the object is moving, which may cause the useful information less than normal. Depending on how much information is captured by the camera, the occlusion can be divided into semi-occlusion and full-occlusion<sup>[4]</sup>. STC can solve the problem of semi-occlusion but not the latter, so it is necessary to improve it by adding Kalman Filter<sup>[5]</sup>.

### Algorithms and Criteria

**STC.** Tracking can be treated as calculating the possibility that target appears at a certain location, so it may be solved by the knowledge of probabilistic model. For the uncertain  $x$ , let  $C(x)$  be the confidence map, meaning that the  $x$  maxing  $C(x)$  is the target location. The  $C(x)$  can be expressed as Eq.1.

$$C(x) = P(x|o) = \sum_{c(z) \in x^c} P(x, c(z)|o). \quad (1)$$

In Eq.1,  $c(z)$  means the feature set of the context. By use of the joint distribution probability formula, Eq.1 can be expressed as Eq.2.

$$C(x) = \sum_{c(z) \in x^c} P(x|c(z), o)P(c(z)|o). \quad (2)$$

In Eq.2,  $P(x|c(z), o)$  is the spatial context model, which represents the relationship between target and surrounding points in the context. When the object is sheltered, the spatial context model can be utilized to continue tracking.  $P(c(z)|o)$ , the context prior model, is a probability, expressing the exterior feature of points in the context. The basic idea of STC is to locate the target by designing the prior probability  $P(c(z)|o)$  and updating the spatial context model  $P(x|c(z), o)$ .

Spatial context model expresses the relationship between the target  $x$  and other points  $z$  in the context, while the positional relationship in space contains distance and direction. Build the spatial context model as Eq.3. The formula Eq.3 means that different  $z$  has different relationships with  $x$ , so the algorithm can use context to mark the target.

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

Because grayscale in the picture could be measured, so we use this feature to build the context prior model, as shown in Eq.4, in which  $I(z)$  means the grayscale of  $z$ . Considering the focus of attention, add a weighting function to express the different impacts on target caused by different  $z$ .

$$P(c(z)|o) = I(z)\omega_\sigma(z - x^*) = I(z) \cdot a \cdot \exp\left(-\frac{|z - x^*|^2}{\sigma^2}\right). \quad (4)$$

Build the confidence map in the shape of normal distribution, see Eq.5.

$$C(x) = P(x|o) = b \exp\left(-\left|\frac{x - x^*}{\alpha}\right|^\beta\right). \quad (5)$$

Associate Eq.3 with Eq.4 to calculate  $C(x)$ . In order to reduce the amount of computation, change the time domain into the frequency, as Eq.6. Learn spatial context model as shown in Eq.7.

$$\mathcal{F}\left(b \exp\left(-\left|\frac{x - x^*}{\alpha}\right|^\beta\right)\right) = \mathcal{F}(h^{sc}(x)) \odot \mathcal{F}(I(z)\omega_\sigma(z - x^*)). \quad (6)$$

$$h^{sc}(x) = \mathcal{F}^{-1}\left(\frac{\mathcal{F}\left(b \exp\left(-\left|\frac{x - x^*}{\alpha}\right|^\beta\right)\right)}{\mathcal{F}(I(z)\omega_\sigma(z - x^*))}\right). \quad (7)$$

For frame  $t-1$ , use Eq.7 to learn  $h_{t-1}^{sc}(x)$ , then update the spatio-temporal context as shown in Eq.8. Calculate the confidence map  $C(x)$  as Eq.9. The  $x$  making  $C_t(x)$  maximum is the target position in frame  $t$ .

$$H_t^{stc} = (1 - \rho)H_{t-1}^{stc} + \rho h_{t-1}^{sc}. \quad (8)$$

$$C_t(x) = \mathcal{F}^{-1}\left(\mathcal{F}(H_t^{stc}(x)) \odot \mathcal{F}(I_t(x)\omega_\sigma(x - x^*))\right). \quad (9)$$

**KSTC.**As known in the world, Kalman Filter is used to calculate the optimal value of next state by combining the estimated and measurement<sup>[6]</sup>. As STC can track object effectively without occlusion, take the output of it as Kalman Filter's measurement<sup>[7]</sup>. Using the displacement and speed to estimate position, establish the state vector as Eq.10. The block diagram of STC based on Kalman Filter (the new algorithm is called KSTC for short) is shown in Fig.1. Kalman Filter uses the output of STC as the measurement and calculates the optimal position of target. For next frame, the output of Kalman Filter helps STC to predict spatial context. The new KSTC is used to solve full-occlusion.

$$X_k = \begin{bmatrix} x_k, y_k, v_{(x)k}, v_{(y)k} \end{bmatrix}^T. \quad (10)$$

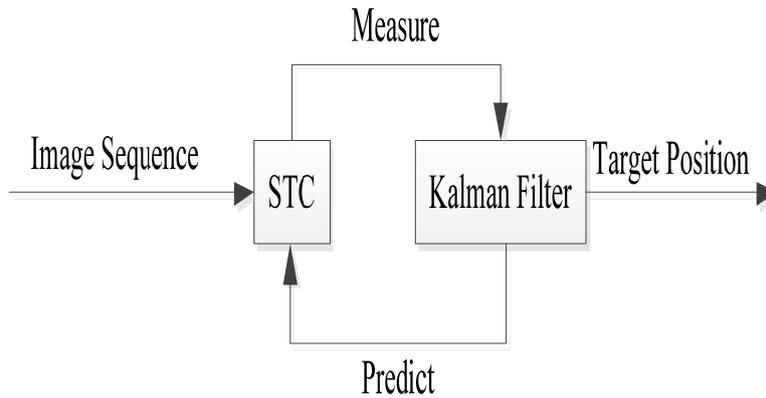


Fig.1 The block diagram of KSTC

**Criteria.** The paper uses three criteria to weigh the performance of algorithms, namely FPS, CLE and OR<sup>[8]</sup>. Frames per second, short for FPS, means the number of tracked frames in each second and reflects the data processing speed. CLE is referred to as center location error, in the name of distance between the centers of tracking rectangle in red and ground-truth rectangle in green. For the sake of simplicity, let  $R_t$  be the symbol of tracking rectangle and  $R_g$  be the other one. OR is the abbreviated form of overlap rate, expressing the ratio of intersection and union between  $R_t$  and  $R_g$ . For FPS and OR, the larger the better but CLE is opposite.

### Algorithm verification

Select "SUV" as the tracked image sequence. The video is about a car driving on the road with a certain speed, experiencing occlusion three times during the tracking. There are some results in Fig.2 showing that STC can track the SUV effectively during frames from 514<sup>th</sup> to 560<sup>th</sup>, which is the first and partial occlusion. After that, the algorithm loses the object and can't capture it again, which can be seen from frames 570<sup>th</sup> and 680<sup>th</sup>. Some results of KSTC are shown in Fig.3. Different



Fig.2 Some results of STC

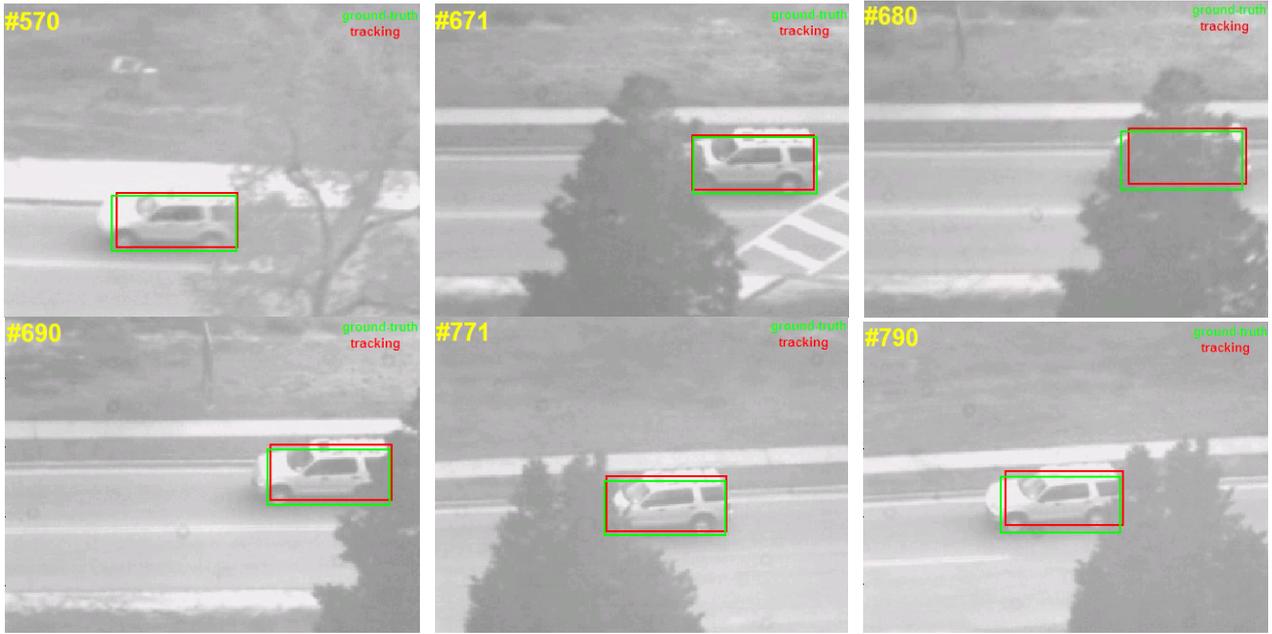


Fig.3 Some results of KSTC

from STC, after the semi-occlusion, KSTC keeps on tracking the object efficiently proved by frame 570<sup>th</sup>. Frames from 671<sup>th</sup> to 690<sup>th</sup> are the second occlusion whose area much larger than the first one and can be treated as full-occlusion. The experiment shows KSTC can catch the object perfectly. There's also another shelter needed to be analyzed, that is frames from 771<sup>th</sup> to 790<sup>th</sup>. The improved algorithm has the ability to track the driving car all the time.

### Summary

With the criteria as mentioned above, the performance of STC and KSTC is compared. The curves of CLE and OR about STC are shown in Fig.4 and Fig.5 while KSTC's are shown in Fig.6 and Fig. 7. The data analysis can be seen in Table.1.

Firstly, comparing Fig.4 and Fig.6 leads to the conclusion that the CLE after frame 560<sup>th</sup> is much larger than before in Fig.4, because STC hasn't tracked the object any longer since then. While the curve in Fig.6 is flat, that is KSTC can solve occlusion so that the distance between centers of  $R_t$  and  $R_g$  doesn't change so much.

Besides, the curves of Fig.5 and Fig.7 are different after frame 560<sup>th</sup> because  $R_t$  and  $R_g$  doesn't have intersection in STC and the value of OR becomes zero. While the value of OR is larger than 0.5 in

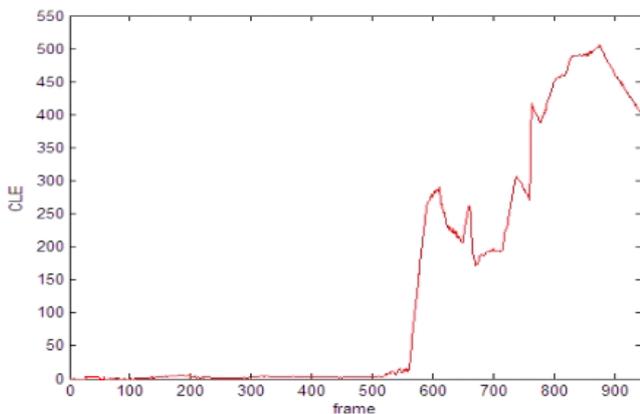


Fig.4 CLE of STC

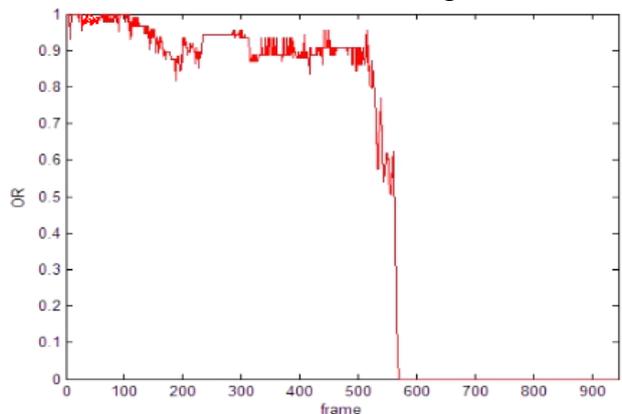


Fig.5 OR of STC

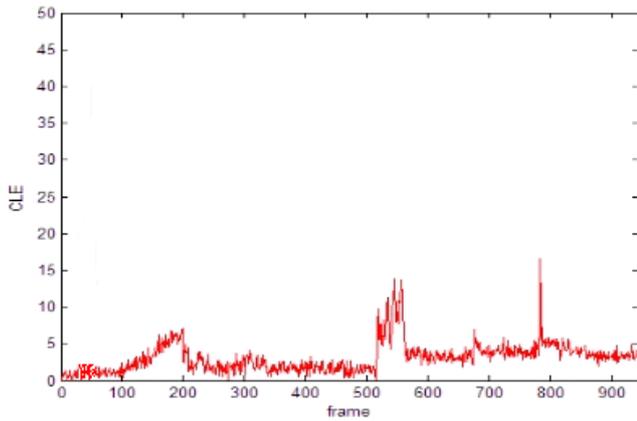


Fig.6 CLE of KSTC

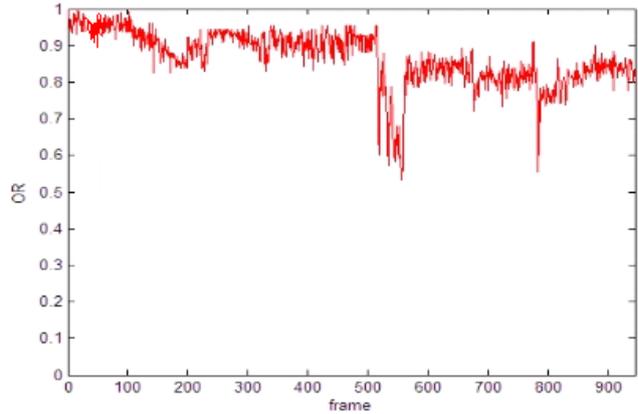


Fig.7 OR of KSTC

Table.1 Performance comparison between STC and KSTC

| Algorithm | FPS    | Average CLE | Max CLE | Average OR | Min OR |
|-----------|--------|-------------|---------|------------|--------|
| STC       | 107.80 | 138.19      | 505.59  | 0.53       | 0      |
| KSTC      | 106.98 | 3.91        | 16.55   | 0.86       | 0.56   |

KSTC, which proves the new algorithm can improve the tracking effect during the occlusion.

Lastly, see the data in Table.1. FPS of STC is 107.8, which means STC can track 107.8 frames per second. While KSTC's is 106.98 and the number is smaller than the former, proving the speed of KSTC is slower than that of STC a little. But the average and maximal CLEs of KSTC are much smaller than STC, showing the distance between  $R_t$  and  $R_g$  is short, just 3.91 pixels medially. The average and minimal ORs of KSTC are larger than STC, proving the tracking precision is higher.

In conclusion, although the speed of KSTC is a little slower than STC, KSTC can solve problems of full-occlusion and its tracking performance is better.

## Reference

- [1] HT Nguyen, J Qiang, AWM Smeulders. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2007, 29(1):52-64.
- [2] Liu, X. ,Lin, L. ,Yan, S. et al. IEEE Transactions on Circuits and Systems for Video Technology, 2011, 21(4):393-407.
- [3] Zhang K H, Zhang L, Liu Q S, Zhang D, Yang M H. Fast visual tracking via dense spatio-temporal context learning. In: Proceedings of the 2014 European Conference on Computer Vision (ECCV). Czech Republic: Springer, 2014. 127-141.
- [4] Ablavsky, Vitaly, Sclaroff, Stan. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(9):1758-1775.
- [5] Baojun Qi, Tao Wu, Hangen He et al. Optical Engineering, 2012, 51(2):160-168.
- [6] Jian-Nan Chi, Chenfei Qian, Pengyun Zhang et al. Neurocomputing, 2014, 128(Mar.27):42-49.
- [7] Youngrock Yoon, Kosaka A., Kak A.C. et al. IEEE Transactions on Robotics, 2008, 24(5): 1238-1251.
- [8] Wu Y, Lim J, Yang M H. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 37(9):1834-1848.