

Implementation of Small Target Tracking based on C-STC with multi-core DSP

Yanyan Lei^{1, a}, Rui Wang^{1, b}, Jinzhong Wu^{2, c}

¹School of Instrumentation Science and Opto-electronics Engineering, Beihang University, Beijing 100191, China;

²The Third Research Institute of China Electronics Technology Group Corporation, Beijing 100015, China.

^abhleiyan@163.com, ^bwangr@buaa.edu.cn, ^cwujinzhong@cetc3.cn

Keywords: C-STC, Small target tracking, Bilinear interpolation, TMS320C6678.

Abstract. A novel method by integrating the color information and bilinear interpolation with the traditional fast tracking via spatio-temporal context learning (STC) called as C-STC is presented. Experimental results with variant videos show that the improved algorithm can track the object accurately compared with the traditional STC, Particle Filter (PF) and Multiple Instance Learning (MIL). Moreover, multi-core emulation experiments have been conducted based on TMDXEVM6678L. Synchronization and communication between cores are realized based on the MessageQ. Two modes of data transmission from PC to DSP are achieved. The results show that the transmission via network promotes the real-time performance, which laying the foundation for communication between the DSP and the gigabit ethernet camera.

1. Introduction

Small target tracking plays an important role in a wide range of application such as reconnaissance and remote monitoring. At present, most of the proposed target tracking algorithms are not suitable for small targets of the size from 7×7 pixels to 25×25 pixels, because the small targets are more complex than dot targets and their image information is much less than that of the normal targets [1, 2] which are larger than 25×25 pixels. Still, there are numerous work been carried out in this field. Zhou [3] realized small target tracking combining gaussian background modeling with Particle Filter (PF), but the algorithm is not suitable for the scene of changed background. Zhao [4] proposed a tracking algorithm using improved Expectation Maximization (EM) method under illumination changes or occlusions. However, the problem of time-consuming is unresolved. In addition, the above algorithms focus on the target itself which lacks the shape, structure and texture information. Zhang [5] proposed STC algorithm, which makes full use of spatio information around the target. But STC is failed to track target stably when the illumination changes.

In this paper, to address the problem of the poor robustness for small target tracking, we improve the performance of traditional STC from two aspects by integrating the bilinear interpolation and color information. Moreover, to promote the embedded application, multi-core emulation experiments are conducted based on TMDXEVM6678L [6] instead of TMS320DM642 [7] which is a single-core processor with poor real-time performance. In the following, section 2 states the traditional STC principle. Section 3 detailed the C-STC. Section 4 shows the implement of improved algorithm with multi-core DSP. Section 5 gives the experimental results and the analysis.

2. Traditional STC Principle

The key to STC is calculating a confidence map $c(x)$ estimating the object location likelihood.

$$c(x) = P(x | o) = \sum_{c(z) \in X^c} P(x, c(z) | o) = \sum_{c(z) \in X^c} P(x | c(z), o) P(c(z) | o) c(x) = b \cdot \exp(-|x - x^*| / \alpha^\beta) \quad (1)$$

The algorithm is implemented in four steps. First, a spatial context model is learned which expresses the spatio relationship between the target and its surrounding background by solving a

deconvolution problem in (3), where context prior probability $p(c(z)|o)$ is calculated in (2). Second, the learned spatial context model is used to update a spatio-temporal context model for the next frame in (4). Third, Tracking in the next frame is formulated by calculating a confidence map as a convolution problem, and we can estimate the target location by maximizing the confidence map in (5) finally.

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

$$h^{sc}(x) = \mathcal{F}^{-1} \left(\mathcal{F}(c(x)) / \mathcal{F}(p(c(x)|o)) \right) \quad (3)$$

$$h_{t+1}^{sc} = (1 - \rho)h_t^{sc} + \rho h_t^{sc} \quad (4)$$

$$x_{t+1}^* = \arg \max_{x \in \Omega_c(x_t^*)} \left(\mathcal{F}^{-1}(\mathcal{F}(h_{t+1}^{sc}(x)) \odot \mathcal{F}(p_{t+1}(c(z)|o))) \right) \quad (5)$$

3. C-STC Algorithm

3.1 C-STC Framework

Our whole C-STC tracking approach is summarized in Fig 1. It inherits the framework of STC and the improvements are shown in red box. The target location in the first frame is designated manually. In the (t) -th frame, the improved context prior probability $p_t(c(z)|o)$ is learned in (7) and the spatial context model is updated in (5). In the frame $(t+1)$ -th frame, based on x_t^* , the interpolated probable target region R'_{t+1} can be determined. Then we compute confidence map in R'_{t+1} , and detect target in (5). At last, to get the actually target position, coordinate system transformation from $X'O'Y'$ to XOY must be done. We will narrate our improvements in two respects.

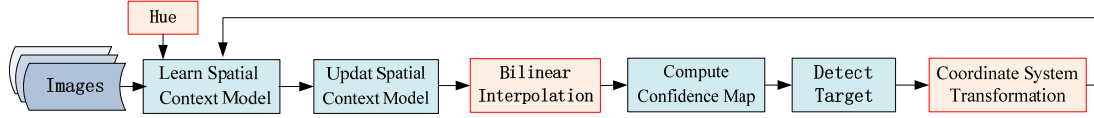


Fig.1 Flow chart of C-STC

3.2 Enlarge the Target and Surrounding Region

Traditional STC is failed to track small targets which lack of information. To solve the problem, interpolation technology [8] is used to construct new data points for the target and surrounding region. In this study, we enlarge the probable target region (see the red box in Fig.2) instead of the whole image for less calculation

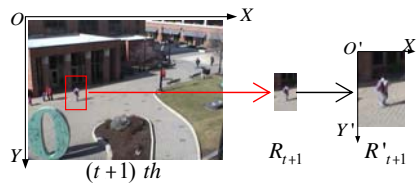


Fig.2 Interpolation of possible target area

The probable target region must be determined firstly. There is a short time interval between neighboring two frames, so we can assume that the target moves with a constant velocity in this period. That is, the deviation of the target location between the (t) -th and the $(t+1)$ -th frame equals the one between the $(t-1)$ -th and the (t) -th frame. Deviations in horizon and vertical directions are set to Δx and Δy respectively. In the (t) -th frame, The target area is the rectangle area R_t , whose width, height and the center are w_t , h_t and x_t^* separately. Then the length and breadth of the probable target region R_{t+1} can be computed by $w_{t+1} = w_t + 2\Delta x + \nu$ and $h_{t+1} = h_t + 2\Delta y + \nu$. Where $\nu=5$ is error compensation value. According to R_{t+1} we can get the corresponding interpolation map R'_{t+1} by calculating the R, G, B values of interpolated points respectively. The interpolation formula of R can be represented as (6), likewise for that of B and G.

$$r(m, n) = r(i, j)(1 - \eta)(1 - \xi) + r(i + 1, j)\eta(1 - \xi) + r(i, j + 1)(1 - \eta)\xi + r(i + 1, j + 1) \times \eta \times \xi \quad (6)$$

Where $\eta = (x_m - x_i) / (x_{i+1} - x_i)$, and $\xi = (y_n - y_j) / (y_{j+1} - y_j)$.

3.3 Integrating Color Information

Compared with the traditional STC which uses the grey intensity information only, we add the color information which plays an important role in tracking. Considering that the R, G, B are easily influenced by illumination in RGB color space and three color channels is used to represent the color information, hue(H) in HIS color space is exploited, which is't seriously affected by illumination.

Firstly, color space is conversed from RGB to HIS and the hue is extracted. Secondly, the hue is used to calculate the context of prior model by (7).

$$p(c(z) | o) = [(1-\lambda)I(z) + \lambda H(z)] \omega_\sigma(z - x^*) \quad (7)$$

Where λ denotes the weight for H and other parameters are the same to (2).

4. The C6678 Implementation of C-STC

Our C-STC is successfully carried out using a TMDXEVM6678L[6] evaluation module (EVM) that includes an on-board C6678 processor. SYS/BIOS which is a lightweight real time native operating system is employed to perform task scheduling and coordination between cores. Synchronization and communication between cores is realized based on the MessageQ combining shared memory. As show in Fig.3, data are transmitted from PC to DSP in two ways.

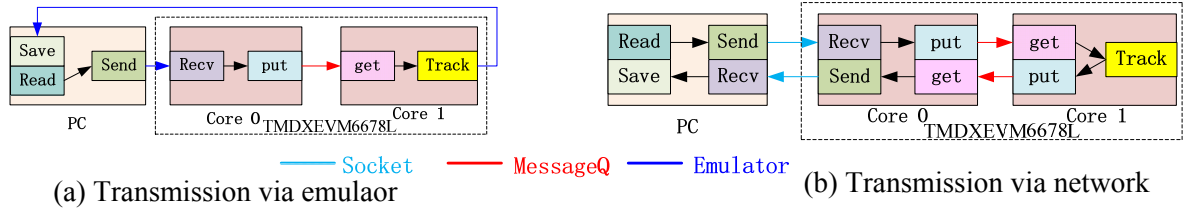


Fig.3 Flowchart of tracking system

The transmission via emulator is illustrated in Fig.3(a). Images are transferred from PC to DSP by emulator firstly. Core 0 performs some initializations and receives images from PC. Object tracking is carried on with core 1. At last, images which are annotated with tracked targets are sent to PC for visualization. As shown in fig.3(b). The framework shown in (b) is the same except the communication between PC and cores 0, which is realized based on socket. PC which is set as the client communicates with DSP which is decided as the server using TCP to ensure rapid and stable network. The key point in the two realization modes is communication between core 0 and core 1. MessageQ which has higher stability and reliability is used for synchronization and communication. Combining with the Shared memory, we only need to pass a pointer without moving data actually, reducing the amount of data copying.

5. Experiments and Evaluation

5.1 Accuracy Analysis

These Experiments are implemented in PC with an Intel Dual-Core 2.83GHz CPU.

Three fully-annotated video sequences are used to validate the performance of our C-STC compared with three algorithms included PF [3], MIL [2] and STC [5]. Partial results of the target tracking in three image sequences are given in Fig.4. In all experiments, the learning parameter in Eqs.(4) is set to $\rho = 0.075$, the normalization constant in Eqs.(2) is set to $a = 2.25$, the shape parameter in Eqs.(1) is decided as $\beta = 1$ and the color coefficient in Eqs.(6) is selected as $\lambda = 0.25$.

Two evaluation metric are used for tracking precision[9]. One is the center location error defined as the average euclidean distance between the center locations of the tracked targets and the manually labeled ground truths. The other is overlap ratio that is $OS = \text{Area}(r_i \cap r_a) / \text{Area}(r_i \cup r_a)$, where r_i is the tracked target box and r_a denotes the ground truth box. The center error and overlap curves are illustrated in Fig.5. As the illumination changes in the sequence of *Crowds*, MIL interrupts tracking at

frame 180 and the performance of PF is unstable. C-STC has better tracking performance after the frame 180 compared with the traditional STC. Fig. 5 (b) shows that, STC and MIL gradually deviate from the actual target location and PF loses the target around the frame 150 due to the low resolution of the target. While C-STC can tracking steadily profiting from color information. For the sequence of *copter*, which has the attribute of motion blur, C-STC achieves outstanding performance owing to more target information by bilinear interpolation. In short, C-STC possesses robustness against variances of illumination.

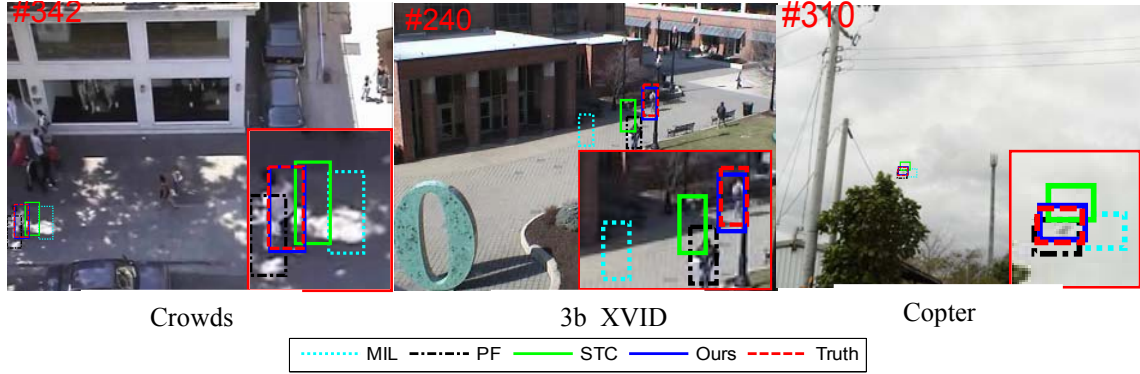


Fig.4 Partial tracking results

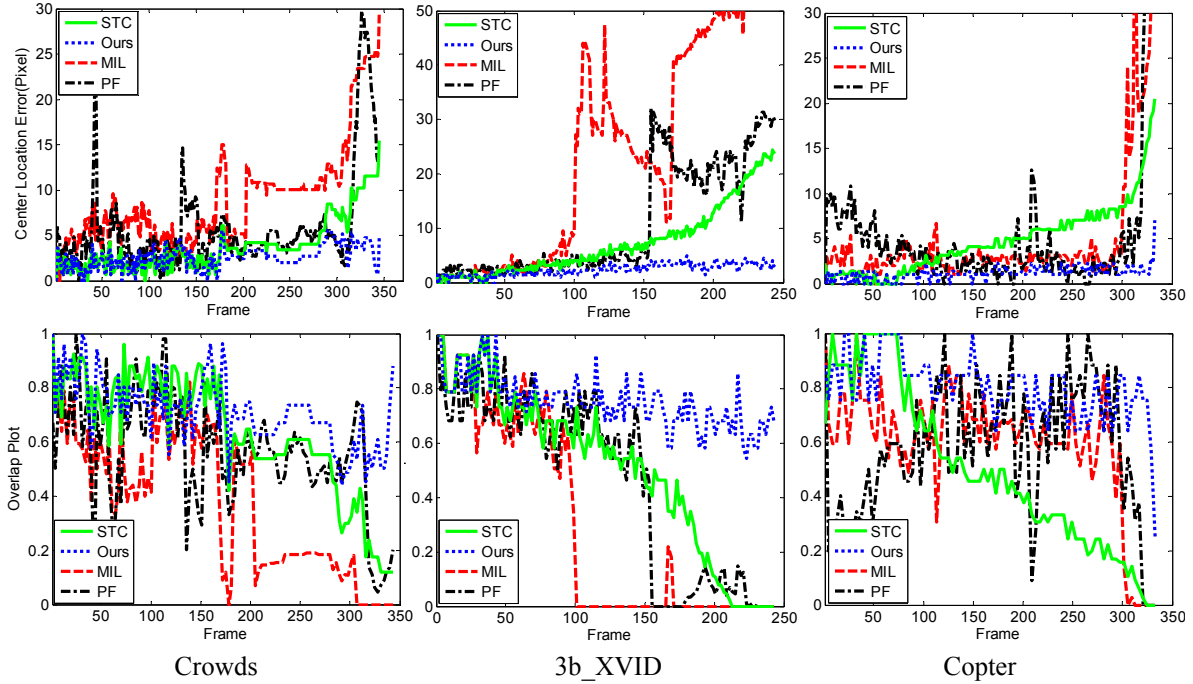


Fig.5 Plots of center location error and overlap

5.2 Time-Consuming Analysis

Table1 Time-consuming of each model(ms/frame)

	Image Resolution (pixels)	Target Resolution (pixels)	Size (KB)	Tracking	Emulator Transmission	Network Transmission	Speedup
Crowds	11×25	320×240	225	8.6	3102	316	90%
3b_XVID	12×25	300×240	211	9.2	2945	311	89%
Copter	16×12	640×424	795	4.6	9130	405	95%

Our experimental evaluation is carried out in the above two systems that include PC and a TMDXEV6678L EVM with an on-board C6678 processor running at 1 GHz. Time-consuming results of each transfer model on 3 video sequences with different sizes are given in Table 1. The experimental results show that images annotated with tracked targets are sent to PC steadily and reliably without any data loss , which means that our data transmission is feasible. Table 1 shows that tracking section can meet the real-time requirements and the transmission over network has an about

90 percent reduction in time-consuming than that of emulator, which laying the foundation for communication between the DSP and the gigabit ethernet camera.

6. Conclusion

In this paper, we have proposed a novel tracking method based on STC combining bilinear interpolation and color information. The proposed method performs well under illumination changes. Moreover, multi-core emulation experiments are conducted based on TMDXEVM6678L. Synchronization and communication between cores and processors have been realized which demonstrates that proposed schemes are feasible

Acknowledgement

This work was partially supported by National Natural Science Foundation of China(60974108) and Research Grant of the CETC-3 (VG201508).

References

- [1] Yilmaz A, Javed O, Shah M.Object tracking: A survey[J]. Acm computing surveys (CSUR), Vol.38 (2006) No. 4, p.1-35.
- [2] Babenko B, Yang M-H, Belongie S.Robust object tracking with online multiple instance learning[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on. Vol. 33 (2011) No. 8, p. 1619-1632.
- [3] ZHOU S X,ZHOU J,SONG L,Tracking Algorithm for Small Objects[J]. Computer Engineering. Vol. 36 (2010) No. 16, p. 186-191.
- [4] Wang R, Zhao Y, Liu Y, et al. A target tracking algorithm based on the continuous wavelet transform of spatio-temporal processing block[C]. Electronics and Optoelectronics (ICEOE), 2011 International Conference on. ShanXi, 2011, p. 95-98.
- [5] Zhang K H,Lei Z ,Liu Q-S,et al. Fast visual tracking via dense spatio-temporal context learning[M]. Germany:Springer International Publishing, 2014, p.127 - 141
- [6] NIU J H.Examples and introduction to TMS320C66X KeyStone architecture multi-core DSP[M]. Shanghai:Shanghai Jiaotong University Press, 2014, p.184-198.
- [7] Wang R, Wang L, Jiang Z-W. Active Visual System for Moving Object Intelligent Tracking Based on DSP[J]. Opto-Electronic Engineering. Vol. 36 (2009) No. 2, p. 6-10.
- [8] Kang Y, Wang G, Hu J.A bilinear interpolation mean shift small target tracking algorithm[C].Signal Processing and Communication Systems (ICSPCS), 2011 5th International Conference on. Honolulu, 2011, p.1-6.
- [9] Wu Y, Lim J, Yang M-H.Online object tracking: A benchmark[C].Computer vision and pattern recognition (CVPR), 2013 IEEE Conference on. Portland, 2013, p. 2411-2418.