

## Research Article

# Target Tracking Algorithm Using Angular Point Matching Combined with Compressive Tracking

Jing Luo,<sup>1,2</sup> Tingting Dong,<sup>1,2</sup> Chunyuan Zi,<sup>1,2</sup> Chunbo Xiu,<sup>1,2</sup>  
Huixin Tian,<sup>1,2</sup> and Jiantao Xi<sup>3</sup>

<sup>1</sup>Key Laboratory of Advanced Electrical Engineering and Energy Technology, Tianjin Polytechnic University, Tianjin 300387, China

<sup>2</sup>School of Electrical Engineering and Automation, Tianjin Polytechnic University, Tianjin 300387, China

<sup>3</sup>School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Keiraville, NSW 2500, Australia

Correspondence should be addressed to Tingting Dong; [dong1331047@163.com](mailto:dong1331047@163.com)

Received 30 June 2015; Revised 20 October 2015; Accepted 25 October 2015

Academic Editor: Samir K. Mondal

Copyright © 2015 Jing Luo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To solve the problems of tracking errors such as target missing that emerged in compressive tracking (CT) algorithm due to factors such as pose variation, illumination change, and occlusion, a novel tracking algorithm combined angular point matching with compressive tracking (APMCCT) was proposed. A sparse measurement matrix was adopted to extract the Haar-like features. The offset of the predicted target position was integrated into the angular point matching, and the new target position was calculated. Furthermore, the updating mechanism of the template was optimized. Experiments on different video sequences have shown that the proposed APMCCT performs better than CT algorithm in terms of accuracy and robustness and adaptability to pose variation, illumination change, and occlusion.

## 1. Introduction

Target tracking is very important in the field of computer vision, involved in intelligent transportation, monitoring security, vision navigation, and other civil and military fields. In recent years, it has been widely studied by relevant scholars in the field, so it has important practical significance [1–3]. Moving target tracking is involved in the motion parameters of object, such as location, velocity, acceleration, and pose in the consecutive image sequences. To study moving target tracking, it would be classified into two categories: (a) directly detect and identify the target in image sequences independent on the prior knowledge, and then find the location of interested target; (b) first of all, build the model according to the prior knowledge, and then find the target accurately in real time in the subsequent frames [4]. Based on these two ideas, a variety of effective target tracking algorithms were derived. But current tracking algorithms remain problems such as tracking drift and tracking error, so it still needs further research and exploration to improve the accuracy, real time performance, and robustness [5–7].

In recent years, the tracking algorithm based on compression perception has caused widely public concern [8, 9]. The concept of compression perception was proposed by Candes and Donoho in 2006; its main idea was to randomly sample from the signal which can be compressed under the condition of sampling intervals far lower than that of the Nyquist sampling and then to reconstruct the original signal within a certain error range through nonlinear reconstruction algorithm [10, 11]. In literature [12], Mei and Ling proposed a robust  $L_1$  tracking algorithm in the particle filter framework. They regarded the target tracking problem as a sparse approximation problem and regarded the candidate which has minimum projection error as tracking target through a  $\ell_1$  norm least squares solution. This method would deal with occlusion and noise effects well; however it is hard to meet the real time requirements due to the large amount of calculation. Zhang et al. [13] put forward a simple and effective compression perception tracking method. A sparse random measurement matrix was generated through certain liberal principles. Zhang et al. compressed the original feature according to the random measurement matrix and

then obtained the model based on sparse representation and distinguished target with background by the naive Bayesian classifier. However, the compressive tracking algorithm will produce tracking errors even missing target due to factors such as target deformation, illumination change, and occlusion.

Aiming at the problem in literature [13], this paper proposed an algorithm combined angular point matching with compressive tracking (APMCCT). Firstly, a sparse measurement matrix was adopted to extract the Haar-like features and the classification results of each sample were calculated. Secondly, the target position offset of the sample which corresponds to maximum value was fused to angular point matching. Finally, to make the APMCCT algorithm much stronger and robust, the updating mechanism of the template was optimized.

## 2. Compression Tracking Algorithm

*2.1. Feature Extraction Based on Compression Perception.* The literature [13] used a sparse matrix to satisfy the condition of constraints isometric sex (RIP) [14], the original feature space was projected to a low-dimensional subspace, and the compressed low-dimensional subspace can keep the information of characteristics in original high-dimensional space. The specific formula is expressed by

$$v = \Psi x, \quad (1)$$

where  $x \in R^{m \times 1}$  is original high-dimensional feature,  $v \in R^{n \times 1}$  is compressed low-dimensional characteristics,  $\Psi \in R^{n \times m}$  is a random measurement matrix, and  $n \ll m$ . Ideally, the low-dimensional characteristics will fully retain information of the high-dimensional signal or the distance relationship of samples in the original space.

A set of rectangle filters  $\{h_{1,1}, \dots, h_{w,h}\}$  were used to convolute with each sample  $z \in R^{w \times h}$  to generate high-dimensional feature:  $\{h_{1,1}, \dots, h_{w,h}\}$ , which was defined as

$$h_{i,j}(x, y) = \begin{cases} 1, & 1 \leq x \leq i, 1 \leq y \leq j \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where  $i$  and  $j$  are the width and height of a rectangle filter, respectively. Then each filtered sample was represented as a column vector in  $R^{wh}$ . Afterwards, these vectors were concatenated as a very high-dimensional feature vector  $X = (x_1, \dots, x_m)^T \in R^m$ , where  $m = (wh)^2$ . The dimension of  $m$  is very high, typically in the order of  $10^6$  to  $10^{10}$ . The literature [13] used a very sparse random measurement matrix; each element of the matrix is defined as

$$\psi_{ij} = \sqrt{s} \times \begin{cases} 1 & \text{with probability } \frac{1}{2s} \\ 0 & \text{with probability } 1 - \frac{1}{s} \\ -1 & \text{with probability } \frac{1}{2s}. \end{cases} \quad (3)$$

Achlioptas [15] proved that this type of matrix with  $s = 2$  or  $3$  satisfied the Johnson-Lindenstrauss lemma. This

sparse matrix form very easily which requires only a uniform random generator. Assume that the probabilities of 1 and  $-1$  are the same and the probabilities were set to be  $1/s$ . Li et al. [16] showed that for  $s = O(m)$  ( $x \in R^m$ ), this matrix is asymptotically normal. Therefore, the new features were the weighted sum of original characteristics with  $\psi_{ij}$  as the weights; namely, each region character was the weighted sum of all characteristics of rectangular area in different size.

*2.2. Classifier Construction and Update.* In current frame, candidate samples were searched in the field of previous target location with radius  $r$ , and then the sparse characteristics of candidate samples were extracted through the random samples measurement matrix. Finally the sample which corresponds to maximum value was selected by naive Bayes classifier. All elements in  $v$  were assumed independently distributed and a model was established with a naive Bayes classifier [17]

$$\begin{aligned} H(v) &= \log \frac{\prod_{i=1}^n p(y=1 | v_i)}{\prod p(y=0 | v_i)} \\ &= \log \frac{\prod_{i=1}^n p(v_i | y=1) p(y=1)}{\prod_{i=1}^n p(v_i | y=0) p(y=0)}, \end{aligned} \quad (4)$$

where  $y = 1$  represents positive sample and  $y = 0$  represents negative sample. The two prior probabilities are equal; all are 50%. In the literature [18], Diaconis and Freedman proved almost all projection of high-dimensional vector met Gaussian distribution, so the conditional probability also belongs to the Gaussian distribution. Therefore the probability density function of normal function distribution could be indicated by the mean and variance value. Namely,

$$\begin{aligned} p(v_i | y=1) &\sim N(\mu_i^1, \sigma_i^1), \\ p(v_i | y=0) &\sim N(\mu_i^0, \sigma_i^0). \end{aligned} \quad (5)$$

In this type,  $\mu_i^1$  and  $\sigma_i^1$  are the mean and variance value of positive sample, respectively.  $\mu_i^0$  and  $\sigma_i^0$  are the mean and variance value of negative sample, respectively.

The formula of updating algorithm is as follows:

$$\mu_i \leftarrow \lambda \mu_i + (1 - \lambda) \mu, \quad (6)$$

$$\sigma_i \leftarrow \sqrt{\lambda (\sigma_i)^2 + (1 - \lambda) \sigma^2 + \lambda (1 - \lambda) (\mu_i - \mu)^2}, \quad (7)$$

where  $\lambda$  is learning factor and  $\lambda > 0$ ; update rate is faster when  $\lambda$  is smaller.

## 3. Harris Corner Detection Principle

Angular point is an important local characteristic of image. These pixels contain rich two-dimensional structure information which gray-scale transformation value high enough in all directions [19, 20]. Harris corner detection algorithm is more effective in feature extraction and it only used the first-order difference of gray and filtering. So it is relatively stable and robust in factors such as rotation, noise, and visual

transform [21]. This experiment adopts Harris operator as the corner extraction operator.

The basic idea of Harris corner detection is to design a local window in the image and then calculate the energy of the window in all directions. The formula is as follows:

$$E(u, v) = \sum_{x,y} w(x, y) [I(x+u, y+v) - I(x, y)]^2, \quad (8)$$

where  $(x, y)$  represents all points in the local window,  $E(u, v)$  represents the energy of image in the local window along one direction, and  $w(x, y)$  is the window which was weighted. Usually, the values of weight which was in the center of the window are higher than that near the window border. The Gaussian function was often used to filter the noise.

Formula (7) in matrix form is

$$E(u, v) = [u \ v] \mathbf{M} \begin{bmatrix} u \\ v \end{bmatrix}, \quad (9)$$

where

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C} & \mathbf{B} \end{bmatrix} = w(x, y) \otimes \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}. \quad (10)$$

$I_x, I_y$  are the derivative of pixels along  $x$ - and  $y$ -axis, respectively. The two eigenvalues of the matrix  $\mathbf{M}$  are larger in the area where corner points exist. The corner points in the image will be detected through the corner response function in type (10) after the correlation function was calculated by (8) and (9)

$$\mathbf{R} = \det(\mathbf{M}) - k \cdot \text{tr}^2(\mathbf{M}), \quad (11)$$

where  $\text{tr}(\mathbf{M})$  and  $\det(\mathbf{M})$  are the trace and determinant value of matrix  $\mathbf{M}$ , respectively. The value of  $k$  was 0.04. Setting a threshold  $T$ , the corner points satisfy the conditions of  $\mathbf{R} \geq T$ .

## 4. Angular Point Matching Combined with Compressive Tracking

**4.1. Angular Point Matching.** Local matching methods based on region mainly contain Sum of Absolute Differences (SAD), Sum of Squared Differences (SSD), and normalized cross correlation (NCC). The minimum value of similarity measure function was selected in SAD and SSD algorithm; however, the maximum value was selected in NCC algorithm. The NCC algorithm has best antinoise ability and matching precision, so this paper adopts the method of NCC as angular point matching:

$$\text{NCC} = \frac{\sum (W_1(x_1, y_1) - u_1)(W_2(x_1, y_1) - u_2)}{\sqrt{\sum (W_1(x_1, y_1) - u_1)^2 \sum (W_2(x_1, y_1) - u_2)^2}}, \quad (12)$$

where  $W_1$  and  $W_2$  are two windows with same size which center on angular point  $p_1$  in image 1 and angular point  $q_1$  in image 2.  $u_1$  and  $u_2$  are the mean of pixel in the windows  $W_1$  and  $W_2$ , respectively, and the value of  $(x_1, y_1)$  in the range of the size of window.

**4.2. Ransac Fine Matching.** There are mismatches no matter what feature descriptor and similarity measure method was used. This paper adopts Ransac [22] to remove the mismatch in the candidate matching points. Ransac algorithm calculates the mathematical model of the data based on sample data sets which contain abnormal data and then gets effective samples.

The basic idea of Ransac algorithm used in this paper is described as follows.

(1) A model and a sample set  $P$  is considered; the potential of the model's minimum sample sets is  $n$  ( $n$  is the minimum of the sample size). The size of the sample is larger than  $n$ . The subset  $S$  contains  $n$  samples randomly extracted from  $P$  and the initial size of  $S$  is  $M$ .

(2)  $B = P/S$  complementary set. The samples from  $B$  whose error on the basis of  $M$  is less than the value  $t$  constituted set of  $W$ .  $W$  and subset  $S$  composed  $S^*$ .  $S^*$  is the set of inliers, and these inliers form consensus set of  $S$ .

(3) The correct parameters of the model are obtained if  $(S^*) \geq N$ . A new model  $M^*$  is recalculated by least squares and  $S^*$ ; then a new  $S$  is extracted, and the above process is repeated.

(4) If the consensus set is not found after certain times of sampling, the algorithm fails. Otherwise, the biggest consensus set is chosen and judge inliers and outliers are judged. And the algorithm is over.

**4.3. Template Update Mechanism.** In literature [13], Gaussian distribution was used to update the template; the mean and variance of sample were updated after tracking. However, it will produce tracking errors due to factors such as occlusion, illumination change, and pose variation. Similarity between template and target would be low; if the template was updated at this time, and it will result in greater deviation. So, in this paper, the position of target was compared with the previous template through NCC, and the template which has larger similarity was regarded as the template of current frame.

This paper set up a threshold  $\eta$ . If normalized cross correlation is greater than the threshold, the template was updated; otherwise the template was not updated.

**4.4. Process of APMCCT.** Steps of angular point matching combined with compressive tracking (APMCCT) were as follows.

(1) Initialization: select the target manually and set the initial parameters; create the projection matrix; set the number of initial characteristics and initial parameters of classifier.

(2) Calculation: read the image of current frame, and then collect several positive samples within the region of  $\|q(x) - q_0\| < \alpha$  and several negative samples within the region of  $\beta < \|q(x) - q_0\| < \gamma$ . Calculate compressed domain feature vector  $v$  of each sample in the candidate area. Extract the angular point of current image by Harris operator. In addition, as the template of the first frame image is calculated within the scope of the target area, the rest of the image given the target motion characteristics is calculated in an expanded 10% area.

(3) Tracking: calculate the response value of each candidate sample by the type (4) and the sample which corresponds



FIGURE 1: APMCCT.

to maximum value was selected to calculate the target position offset; calculate the angular point offset and then the two offsets are used to calculate target position.

(4) Update classifier: calculate normalized cross correlation between current location and the initial template; if the normalized cross correlation is greater than the threshold, the template was updated according to the type ((6), (7)).

(5) Return to step (2) to process the next frame image.

## 5. Experiment Results and Analysis

This experiment was implemented in MATLAB R2012a, which runs at 35 frames per second (FPS) on a AMD 2.10 GHz CPU with 4 GB RAM. The simulation software MATLAB R2012a was downloaded on the Internet. In order to verify the robustness of the new algorithm, the performance of APM-CCT algorithm was compared with the original compression tracking. Some tracking results were shown in Figures 1, 2, 4, 5, 7, and 8 about Girl, SUV, and Fish. Furthermore,

the tracking error curve was mapped as shown in Figures 3, 6, and 9. These challenging test sequences are from [1] and they are available on the website <http://visultracking.net>.

*5.1. Experiment Parameter Settings.* According to the target location at the current frame, the search radius of positive samples set  $X^\alpha$  was set to  $\alpha = 4$  which generates 45 positive samples. The inner and outer radii to generating negative samples set  $X^{\beta,\gamma}$  were set to  $\beta = 8$  and  $\gamma = 30$ , respectively. The number of negative samples was set to be 50 which is selected randomly from set  $X^{\beta,\gamma}$ . The search radius of candidate samples set  $X^\zeta$  was set to  $\zeta = 20$  and 1100 candidate samples were generated. The dimensions of the projection space was set to  $n = 50$ , and the learning parameter was set to  $\lambda = 0.85$ .

*5.2. Results and Analysis.* Center Location Error (CLE) was used to evaluate APMCCT and CT algorithm. CLE is

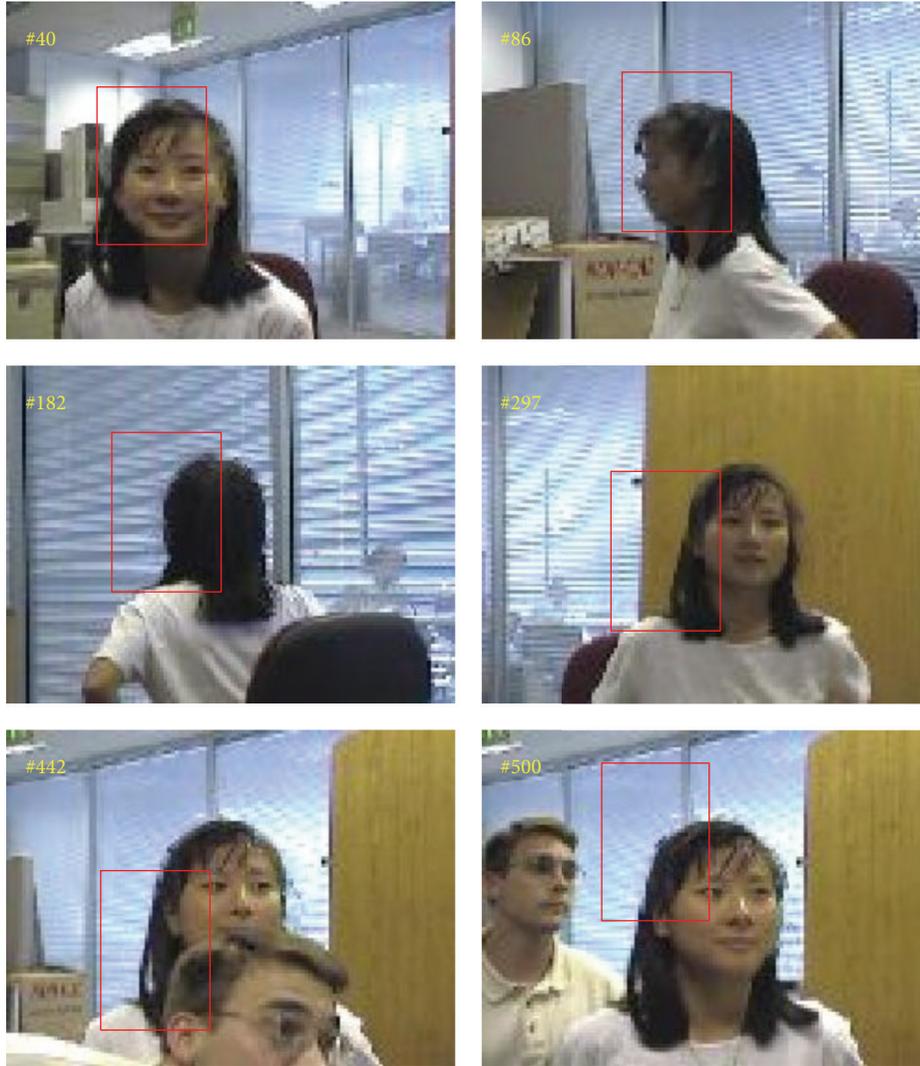


FIGURE 2: CT.

the Euclidean distance of the real location center with the target tracking results. The specific formula is expressed by

$$\text{CLE} = \sqrt{(x_{\text{real}} - x_{\text{track}})^2 + (y_{\text{real}} - y_{\text{track}})^2}, \quad (13)$$

where  $x_{\text{real}}$  and  $x_{\text{track}}$  are the  $x$ -coordinate of real location center and tracking location center, respectively. Likewise,  $y_{\text{real}}$  and  $y_{\text{track}}$  are the  $y$ -coordinate of real location center and tracking location center, respectively.

Girl video sequence is constituted of 500 frame color images whose size is  $128 \times 96$ . The background of these sequence images is relatively simple. In the process of tracking, the head and the body of the target rotated 360 degrees, and the people who are similar to the tracking target appeared. In the experiments, the representative sequence images such as the 40th frame, the 86th frame, the 182nd frame, the 297th frame, the 442nd frame, and the 500th frame were selected to show the performance of APMCCT and compression tracking (CT) algorithm, as shown in Figures 1

and 2. When the target rotated in the process of tracking, the deviation of CT was becoming increasingly big as errors accumulate continuously, so the tracking effect is not good. APMCCT algorithm is oppositely stable. When the target rotated and changed in appearance in the 86th frame, 182nd frame, and 297th frame, APMCCT algorithm still tracked accurately. When the people who are similar to the tracking target appeared in the 442nd frame and the 500th frame, APMCCT algorithm performed relatively accurately. What is more, APMCCT algorithm would quickly find the tracking target; on the contrary, CT algorithm failed to find the target correctly while the target is missing. But because APMCCT algorithm did not make improvement on scale, the performance of APMCCT algorithm was slightly down in some frames with range from 90th frame to 110th frame and range from 180th frame to 240th frame. The change of scale leads to changing the area of corner detection, so angular point matching did not focus on the area of the target. We will research and make improvement in this respect next. In order

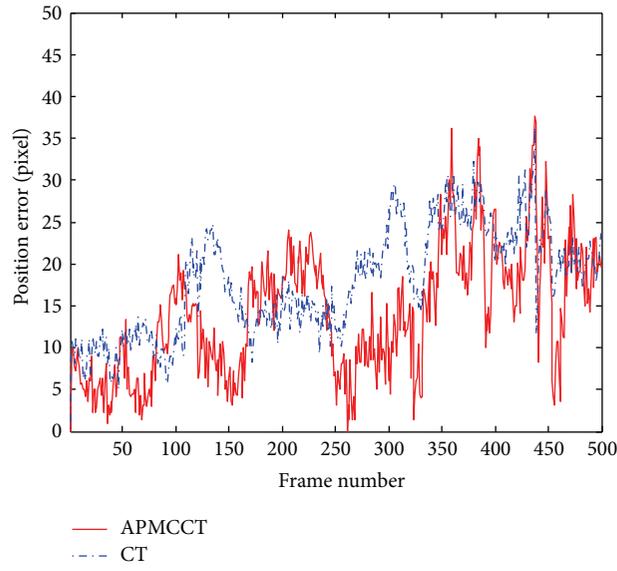


FIGURE 3: Girl tracking error curve.

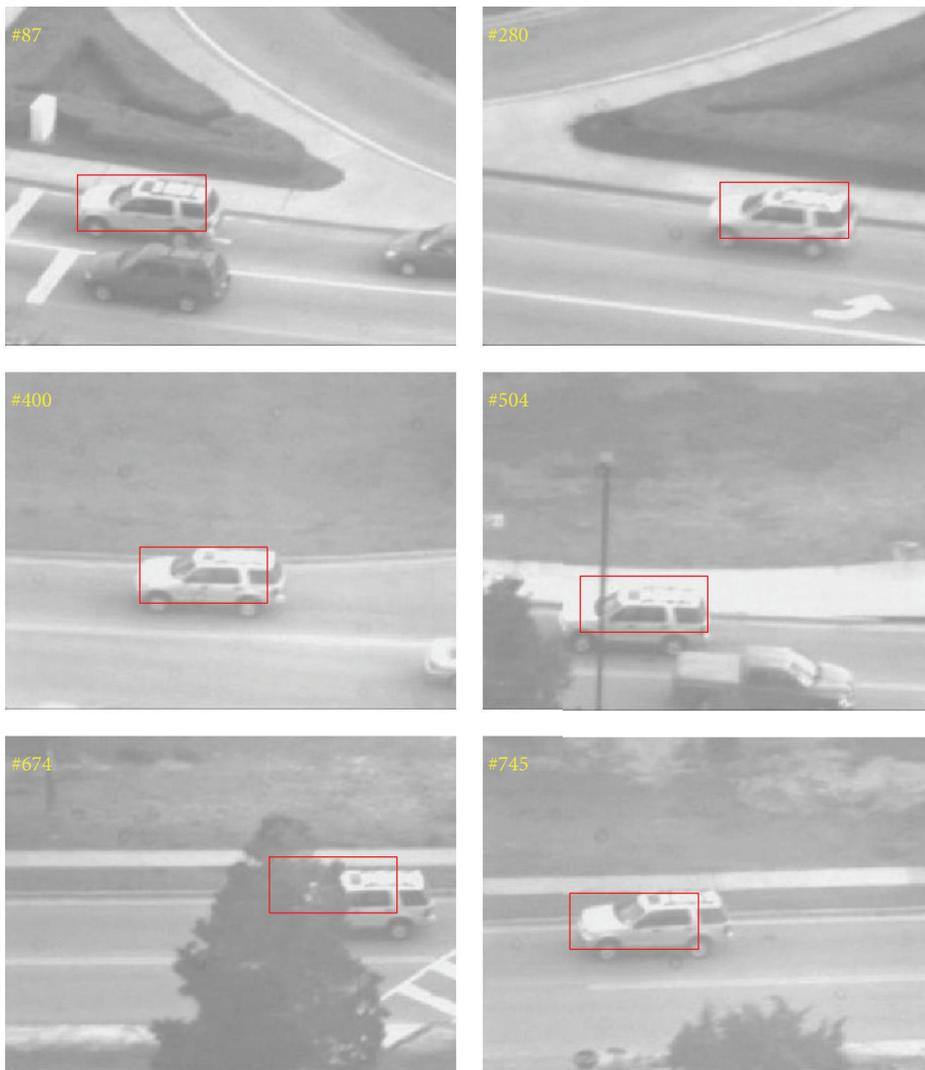


FIGURE 4: APMCCT.

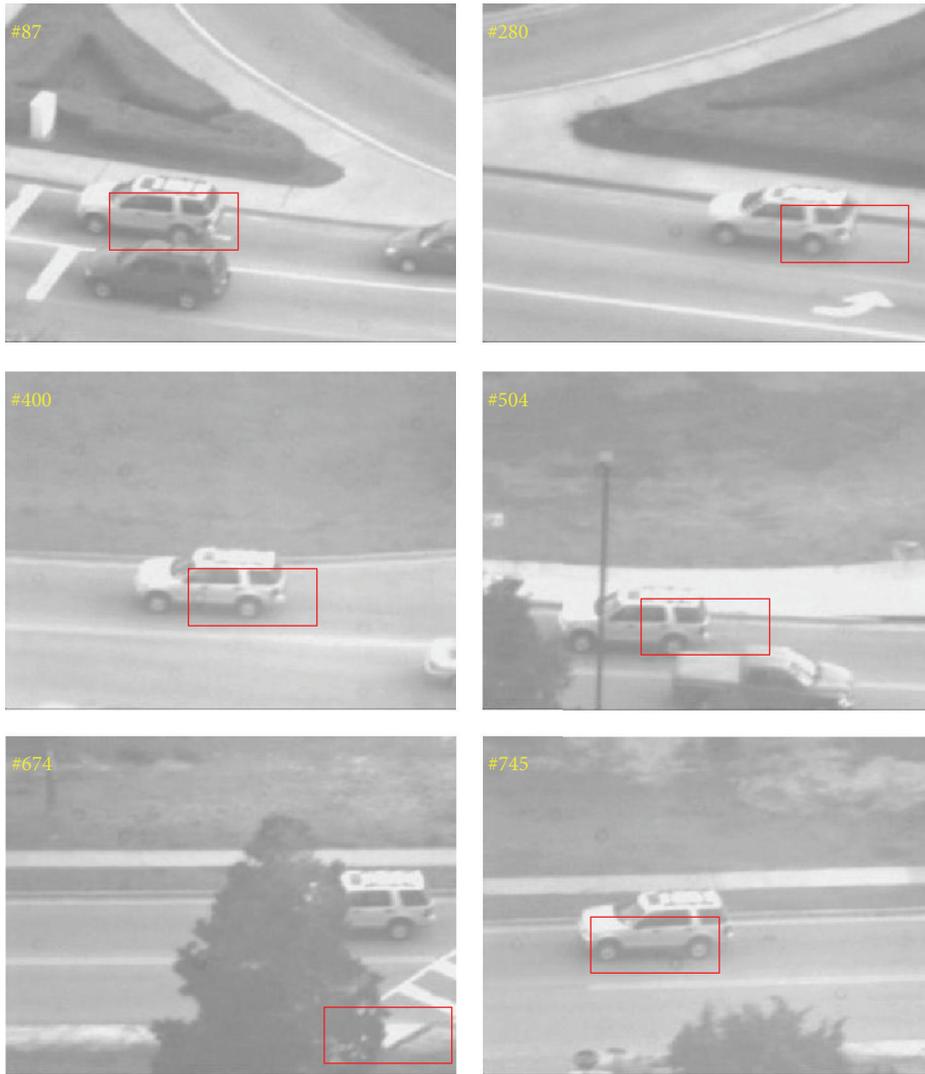


FIGURE 5: CT.

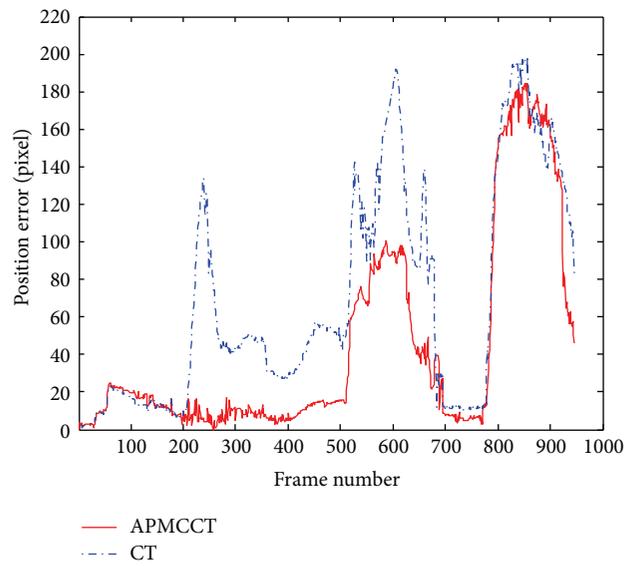


FIGURE 6: SUV tracking error curve.

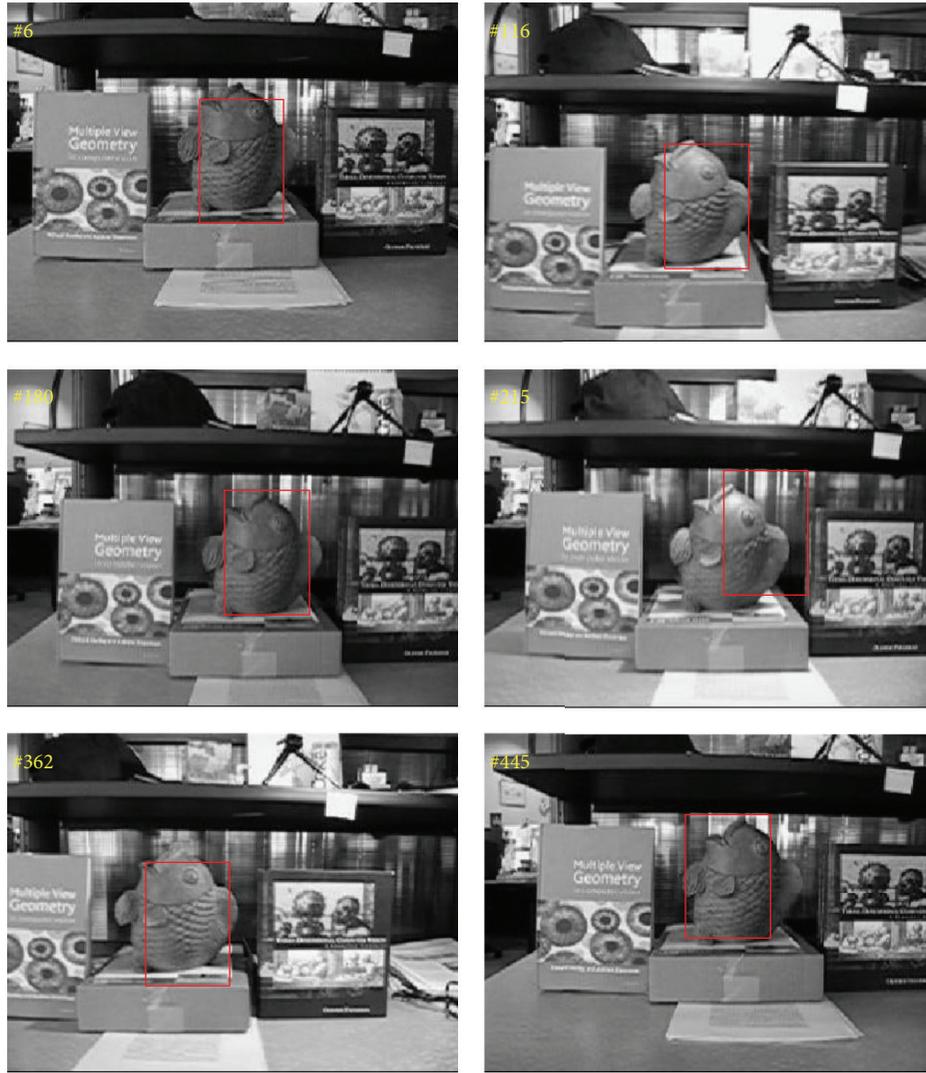


FIGURE 7: APMCCT.

to compare the performance of APMCCT and CT algorithm in each frame of the video sequence, the tracking error curve of Girl video sequence was mapped as shown in Figure 3.

SUV video sequence is composed of 945 frame images whose size is  $320 \times 240$ . In the process of tracking, the scale of the target was changed, and different obstructions appeared. In the experiments, the representative sequence images such as the 87th frame, the 280th frame, the 400th frame, the 504th frame, the 674th frame, and the 745th frame were selected to show the performance of APMCCT and compression tracking (CT) algorithm, as shown in Figures 4 and 5. APMCCT and CT algorithm tracked accurately until the obstructions appeared. When the obstructions appeared in the process of tracking, the deviation of CT was becoming increasingly big as errors accumulate continuously, even leading to loss of the target. Instead, APMCCT algorithm still tracked accurately as shown in the 504th frame, 674th frame, and 745th frame. APMCCT algorithm would quickly find the tracking target after the tracking is drifting or missing. In order to compare

the performance of APMCCT and CT algorithm in each frame of the video sequence, the tracking error curve of SUV video sequence was mapped as shown in Figure 6.

Fish video sequence is composed of 476 frame images whose size is  $320 \times 240$ . In the experiments, the representative sequence images such as the 6th frame, the 116th frame, the 180th frame, the 215th frame, the 362nd frame, and the 445th frame were selected to show the performance of APMCCT and compression tracking (CT) algorithm, as shown in Figures 7 and 8. The deviation of CT was becoming increasingly big as errors accumulate continuously, even leading to loss of the target because illumination changed and camera dithered. Instead, the tracking error of APMCCT algorithm was relatively small. When illumination changed in the 362nd frame and 445th frame, APMCCT algorithm still tracked accurately. In order to compare the performance of APMCCT and CT algorithm in each frame of the video sequence, the tracking error curve of Fish video sequence was mapped as shown in Figure 9.

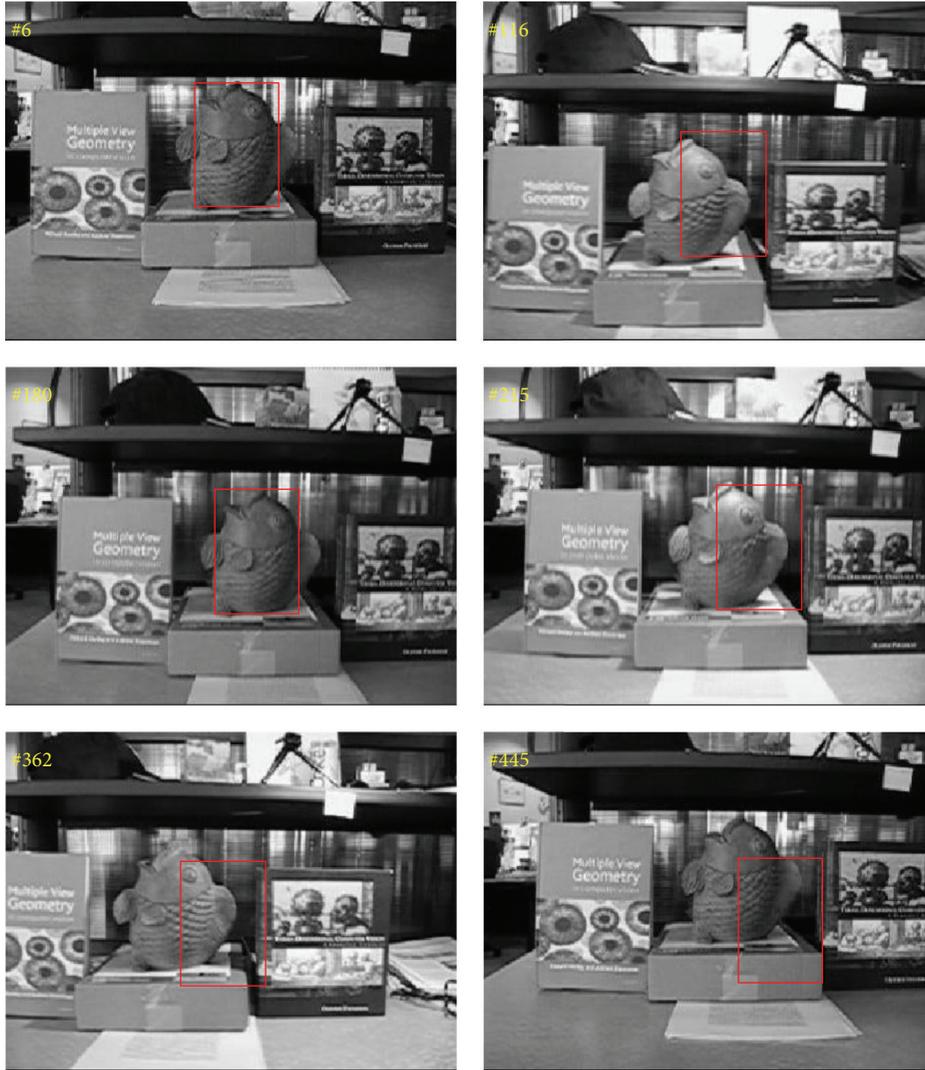


FIGURE 8: CT.

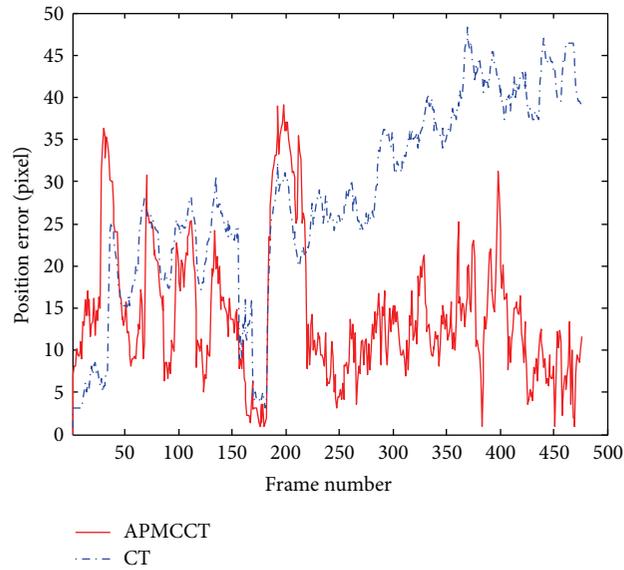


FIGURE 9: Fish tracking error curve.

## 6. Conclusion

Aiming at the problem of object tracking, a novel algorithm combined angular point matching with compressive tracking (APMCCT) was proposed. And the updating mechanism of the template was optimized. The experiments for different video sequences show that APMCCT algorithm can adapt to the changes of the target and the background and track target accurately. The accuracy and robustness have obvious improvement compared with CT algorithm. But APMCCT algorithm is still unable to adaptively adjust the scale of the tracking box, which will be studied in the future.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (no. 61203302 and no. 61403277) and the Tianjin Research Program of Application Foundation and Advanced Technology (14JCYBJC18900).

## References

- [1] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: a benchmark," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2411–2418, Portland, Ore, USA, June 2013.
- [2] J. C. Sanmiguel, A. Cavallaro, and J. M. Martínez, "Adaptive online performance evaluation of video trackers," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2812–2823, 2012.
- [3] J. Luo, D. Song, C. Xiu, S. Geng, and T. Dong, "Fingerprint classification combining curvelet transform and gray-level cooccurrence matrix," *Mathematical Problems in Engineering*, vol. 2014, Article ID 592928, 15 pages, 2014.
- [4] J. Zhang, X. B. Mao, and T. J. Chen, "Survey of moving object tracking algorithm," *Application Research of Computers*, vol. 26, no. 12, pp. 4407–4410, 2009.
- [5] S. Avidan, "Support vector tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. I-184–I-191, IEEE, Kauai, Hawaii, USA, December 2001.
- [6] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1838–1845, IEEE, Providence, RI, USA, June 2012.
- [7] J. Luo, S. Geng, C. Xiu, D. Song, and T. Dong, "A curvelet-SC recognition method for maize disease," *Journal of Electrical and Computer Engineering*, vol. 2015, Article ID 164547, 8 pages, 2015.
- [8] S. Zhang, H. Yao, X. Sun, and X. Lu, "Sparse coding based visual tracking: review and experimental comparison," *Pattern Recognition*, vol. 46, no. 7, pp. 1772–1788, 2013.
- [9] X. Mei and H. Ling, "Robust visual tracking using  $\ell_1$  minimization," in *Proceedings of the 12th IEEE International Conference on Computer Vision (ICCV '09)*, pp. 1436–1443, IEEE, Kyoto, Japan, September–October 2009.
- [10] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [11] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [12] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2259–2272, 2011.
- [13] K. H. Zhang, L. Zhang, and M. H. Yang, "Real time compressive tracking," in *Proceedings of the European Conference on Computer Vision (ECCV '12)*, pp. 864–877, Florence, Italy, October 2012.
- [14] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.
- [15] D. Achlioptas, "Database-friendly random projections. Johnson-Lindenstrauss with binary coins," *Journal of Computer and System Sciences*, vol. 66, no. 4, pp. 671–687, 2003.
- [16] P. Li, T. Hastie, and K. W. Church, "Very sparse random projections," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06)*, pp. 287–296, Philadelphia, Pa, USA, August 2006.
- [17] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifier: a comparison of logistic regression and naive bayes," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS '02)*, Vancouver, Canada, November 2002.
- [18] P. Diaconis and D. Freedman, "Asymptotics of graphical projection pursuit," *The Annals of Statistics*, vol. 12, no. 3, pp. 793–815, 1984.
- [19] J. Gao, Y. Wu, K. Wu, and J. Sun, "Image matching method based on corner detection," *Chinese Journal of Scientific Instrument*, vol. 34, no. 8, pp. 1717–1725, 2013.
- [20] Z. Weichuan, C. Dong, and Z. Lei, "Multi-scale corner detection based on anisotropic Gaussian kernels," *Journal of Electronic Measurement and Instrument*, vol. 26, no. 1, pp. 37–42, 2012.
- [21] C. G. Harris and M. J. Stephens, "A combined corner and edge detector," in *Proceedings of the Alvey Vision Conference*, pp. 147–151, 1998.
- [22] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

