International Journal of Control and Automation
Vol. 10, No. 1 (2017), pp.63-70
http//dx.doi.org/10.14257/ijca.2017.10.1.06

A Kernel Sparse Representation Based Visual Tracking

Xiping Duan[1, 2], Jiafeng Liu[1] and Xianglong Tang[1]

[1]School of Computer Science and Technology
Harbin Institute of Technology, Harbin, China;
[2]College of Computer Science and Information Engineering
Harbin Normal University, Harbin, China
Xpduan_1999@126.com, {jefferyliu,tangxl}@hit.edu.cn

Abstract

Recently, the sparse representation based visual tracking is very popular, which is robust to occlusion and noise, but not satisfactory in the scenarios of fast motion and blur. In order to solve this problem, a new tracking method based on kernel sparse representation is proposed. In this method, each candidate is represented by kernel sparse representation, then the computed reconstruction error is used to obtain the observation probability, and at last the candidate with the maximal observation probability is determined as the target. Aspect to the solving of kernel sparse representation, the accelerated proximal gradient (APG) is adopted. Experiments on several representative image sequences shows that the proposed tracking method performs better than the sparse representation based visual tracking method in the fast motion and blur scenarios.

Keywords: *Computer vision; visual tracking; kernel sparse representation*

1. Introduction

Visual tracking is a hot topic in computer vision and has lots of applications such as intelligence surveillance, vehicle navigation, advanced human-computer interaction. But due to the impact of factors of the change of pose, shape, illumination and viewpoint, cluttered background, noise, occlusion, and so forth, it is still a challenge to achieve robust target tracking.

The visual tracking methods can be categorized as the generative and the discriminative. The generative methods regard the target tracking as a matching problem. For example, the Eigentracker[1], proposed by Black *et al*. and the mean shift tracker[2], proposed by Comaniciu, which have desired real-time performance, but lack the necessary updating of the object templates. IVT[3] proposed by Ross *et al*. which trained an low dimensional sub-domain incrementally to adapt to the change of object appearance. VTD [9] proposed by Kwon *et al*., which used multiple motion models and multiple appearance models to extend the traditional particle filter. The discriminative methods can be regarded as a classification problem for two classes, in which the positive and negative samples are needed to configure and update the classifier. There are some representative methods: Avidan *et al*. [4] proposed to track an object by using the SVM. Collins *et al*. [5] proposed to discriminate the object and background by selecting a group of features online. Grabner *et al*. [10] proposed the online semi-supervised boosting method to improve tracking performance and reduce drift. Babenko *et al*. [6] introduced the multi-instance learning into the visual tracking to avoid the label bias and reduce drift.

Recently, sparse representation has received rapid development in various fields, including face recognition, background subtraction, texture segmentation, image classification, visual tracking and so on. In visual tracking, the target appearance changes over time, with different appearance pattern in different time, therefore the sparsity is obvious and it is a natural way to use sparse coding to represent the object. Sparse representation related visual tracking algorithms have emerged. Mei *et al*. [11] firstly proposed the sparse representation based tracking algorithm, which was motivated by sparse representation based face recognition. Some work [7, 12-14] used sparse representation to generate feature vectors for further searching the target. Zhang *et al*. [15] proposed to solve the sparse representation of the target template with candidate samples as training samples smartly. Li *et al*. [17] proposed to use the compression sensing to extract feature and reducing feature dimensions to improve efficiency. Additionally, some work [16] focused on fusing sparse coding based feature representation and searching.

Although the sparse representation based trackers perform well in many representative sequences, it isn't satisfactory to track in the sequences with the fast motion of the object. Motion blur is caused by the exposure time and relative motion between the camera and object. Usually blur is modeled as a linear convolution of an image with a blurring kernel. To remove the effect of motion blur, it is necessary to estimate the blurring kernel. However the estimate of blurring kernel is a challenge in image processing, which is time consuming and complex, not suitable for visual tracking. Kernel technologies initially used in SVM, could change the distribution of samples and separate linearly inseparable samples linearly in implicit high dimensional Hilbert space. So in this paper, a sparse representation in Hilbert space by using the kernel trick is used for tracking the target.

2. Problem Formulation

In this section, we present the tracking framework and find the problem to be settled.

2.1. Tracking Framework

The basic flow of the sparse representation based tracking method is illustrated in Figure 1 and the procedures of the tracking system can be summarized as follows. Initially, the target location is given manually in the 1st frame, and several samples are cropped within 3 pixels around the target location and used as object templates. Then from the 2nd frame, the following two steps are alternately carried out continuously till the last frame. First, when the next frame comes, some samples are cropped as candidates within a searching radius of r pixels centering at the old object location of last frame. And the observation probability is computed using the chosen observation model for each sample. Then the candidate with maximal observation probability is determined as the object of this frame. Second, the new object is used to update the object template set.

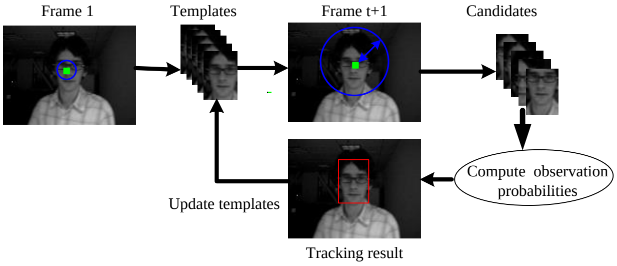


Figure 1. The Tracking Framework

2.2. Kernel Sparse Representation

The previous procedures of visual tracking show that the observation model is critical to the success of the whole tracking system. In the sparse representation based visual tracking, each candidate is represented as the linear combination of object templates, and the reconstruction error is used as its observation probability. Similarly, in the proposed method, each candidate is also represented as the linear combination of object templates. But it is needed that all the candidates and object templates firstly are mapped into a high dimensional space. Specifically, let $\{T_1, T_2, ..., T_M\}$ denote the object template set, which is composed of M object templates. Let x denote a candidate with feature vector y. The corresponding mapped object templates set and candidates are denoted as $T = \{\phi(T_1), \phi(T_2), ..., \phi(T_M)\}$ and $\phi(y)$ respectively using the mapping function $\phi(.)$. The mapped candidate $\phi(y)$ can be sparsely represented as the linear combination of the mapped object templates T with sparse coefficient vector α as follows.

$$\alpha = \arg\min_{\alpha'} \left\|\alpha'\right\|_1 \qquad s.t. \qquad \phi(y) = T\alpha' \tag{1}$$

If the constraint condition in (1) is relaxed to tolerate error to some degree, then the sparse coefficient vector α can be gotten by solving the following object function.

$$\min_{\alpha'} \left\|\phi(y) - T\alpha'\right\|_2 + \lambda\left\|\alpha'\right\|_1 \tag{2}$$

where λ is the sparse parameter. The problem seems to be clear and simple. However, since the mapped object template set T and the mapped candidate $\phi(y)$ is unknown and couldn't obtained directly, the sparse coefficient vector α cannot be obtained by solving (2) directly. Fortunately, according to the theorem in [8](Yin *et al.*), the sparse coefficient vector α can be obtained by modifying (2) as (3).

$$\min_{\alpha'} \left\|T^T\phi(y) - T^T T\alpha'\right\|_2 + \lambda\left\|\alpha'\right\|_1 \tag{3}$$

In (3), it is needed to compute $T^T\phi(y)$ and $T^T T$, where

$$T^T\phi(y) = [\phi(T_1)^T\phi(y), \phi(T_2)^T\phi(y), ..., \phi(T_M)^T\phi(y)]^T \tag{4}$$

and

$$T^T T = \begin{bmatrix} \phi(T_1)^T\phi(T_1) & \phi(T_1)^T\phi(T_2) & \cdots & \phi(T_1)^T\phi(T_M) \\ \phi(T_2)^T\phi(T_1) & \phi(T_2)^T\phi(T_2) & \cdots & \phi(T_2)^T\phi(T_M) \\ \vdots & \vdots & \ddots & \\ \phi(T_M)^T\phi(T_1) & \phi(T_M)^T\phi(T_2) & & \phi(T_M)^T\phi(T_M) \end{bmatrix} \tag{5}$$

Thus the inner product of two mapped samples $\phi(y_1)$ and $\phi(y_2)$ can be computed as $\phi(y_1)^T\phi(y_2) = K(y_1, y_2)$ by choosing an appropriate kernel function $K(.,.)$

Equation (3) can be extended further with the error item as follows:

$$\min_{\begin{bmatrix}\alpha' \\ E'\end{bmatrix}} \left\|T^T\phi(y) - [T^T T, I]\begin{bmatrix}\alpha' \\ E'\end{bmatrix}\right\|_2 + \lambda\left\|\begin{bmatrix}\alpha' \\ E'\end{bmatrix}\right\|_1 \tag{6}$$

where I is the identity matrix and E is the corresponding sparse coefficient. By the added error item, the greater error is allowed.

After the sparse coefficient α has been obtained, it can be used to compute the reconstruction error re of the responding candidate.

\

$$re = \left\|\phi(y) - T^T\alpha\right\|_2 \tag{7}$$

The reconstruction error can further be used to compute the observation probability of candidate x.

$$P(\mathbf{y}|o) \propto c \cdot \exp(-re) \tag{8}$$

here o and c are the real object state and the normalization constant respectively.

2.3. APG for Problem Solving

Equation (6) can be solved by APG (Accelerated Proximal Gradient) algorithm, which is composed of two continuously alternate steps. The first step is the proximal gradient descent and attenuation, and the second step is the aggregation between two iterations. The details are shown as follows.

(1) Initialization: $\mathbf{z}_1 = \mathbf{x}_0 = 0$, $t_1 = 1$;

(2) Let $L = 2\lambda_{\max}(\mathbf{A}^t \mathbf{A})$;

(3) Do while loop till the convergence:

1) $\mathbf{x}_k = \left[\left|\mathbf{z}_k - 2/L \cdot \mathbf{A}^T(\mathbf{A}\cdot\mathbf{z}_k - y)\right| - \mu/L\right]_+ \cdot \mathrm{sgn}(\mathbf{z}_k - 2/L\cdot\mathbf{A}^T(\mathbf{A}\cdot\mathbf{z}_k - y))$;

2) $t_{k+1} = (1+\sqrt{1+4t_k^2})/2$;

3) $\mathbf{z}_{k+1} = \mathbf{x}_k + (t_k - 1)/t_{k+1}\cdot(\mathbf{x}_k - \mathbf{x}_{k-1})$.

where $\mathbf{A} = [T^T T, I]$ and $[\mathbf{x}]_+ = \max(0, \mathbf{x})$.

2.4. Template Updating

To reduce the tracking drift and improve tracking accuracy, it is necessary to update the templates with time. But how to update is difficult to answer. Here, a simple but effective method is used, in which the result of tracking is used to update the templates. Specifically, after the tracking result is obtained, the error ratio er is computed as follows.

$$er = \|E\|_1 / (\|\alpha\|_1 + \|E\|_1) \tag{9}$$

If the er is not greater than a threshold τ, the result is used to update templates.

3. Tracking Algorithm

3.1. Feature Vector

Before the sparse representation and observation probability of each candidate are computed, the feature vectors of object templates and candidates should be obtained. In this paper, the feature vector of an object template or a candidate is obtained by stacking the corresponding image columns to form a 1D vector.

3.2. Details of Algorithm

The details of the algorithm are given in what follows:

(1) Initialization: The object location in the first frame, the object template set $\{T_1, T_2, ..., T_M\}$ and the kernel function $K(.,.)$.

(2) From the 2nd frame, the following steps are alternately carried out continuously till the last frame.

1) Crop N samples $\{x_1, x_2, ..., x_N\}$ with feature vectors $\{y_1, y_2, ..., y_N\}$ as candidates within a searching radius r pixels centering at the old object location of last frame.

2) Calculate the sparse coefficient vector α_i for each sample x_i by solving (6).

3) Compute the observation probability $P(\mathbf{y}_i|o)$ for each sample x_i using (7) and (8).

4) Determine x_j s.t. $j = \arg\max_i P(y_i|o)$ as the object state of the current frame.

5) The tracking result is used to update the template set as section 2.4.

4. Experiments

In this section, the effectiveness of our tracking algorithm is evaluated by the following experiments. Since the proposed algorithm is developed to improve the performance of the sparse representation based tracking in the fast motion and blur scenarios, thus comparison is conducted between them. Three representative image sequences with fast motion of the object, which are Jumping, Deer and Lemming, are chosen. In experiments, all the images are converted to gray-level, and the sparse parameter λ, the searching radius r and the updating threshold τ are set as 0.2, 6, and 0.125 respectively. The chosen kernel function is polynomial kernel $K(x,y)=(1+x^T y)^d$ with the parameter $d=2$.

Jumping sequence is an example of abrupt motion. In this sequence, a man is jumping up and down in front of a building. But due to the limitation of the camera and the fast motion of the object, the blur occurs. In such case, the traditional sparse representation based tracking system begins to drift away after the 210th frame, and loses the object finally, while the proposed algorithm can track the object robustly using a more accurate observation model as shown in Figure 2 and Figure 5 (a).

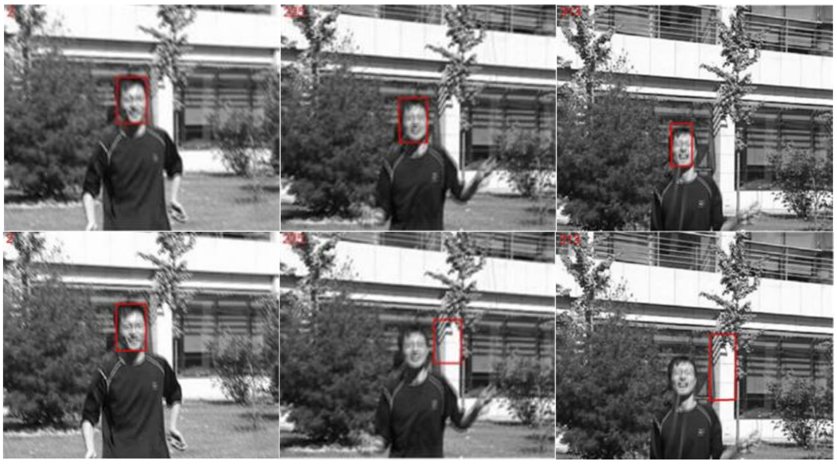


Figure 2. The Tracking Result Comparison between the Proposed Method (The 1st Row) and the Sparse Representation Based Tracking Method (the 2nd Row) In Frames 2, 225 and 313 of Jumping Sequence

Deer sequence is another example. In this sequence, several deer are running forwardly, among which a deer is the object of tracking. Similar to the previous jumping sequence, the fast motion causes the blur and becomes the barrier to robust tracking. The sparse representation based tracking begins to drift just after several frames, while the proposed method can stably and robustly track during the whole period, as shown in Figure 3 and Figure 5 (b).

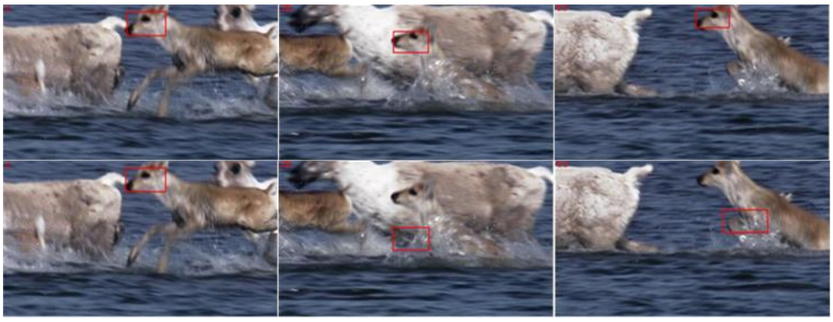


Figure 3. The Tracking Result Comparison between the Proposed Method (The 1st Row) and the Sparse Representation Based Tracking Method (The 2nd Row) In Frames 2, 26 and 63 of Deer Sequence

In Lemming sequence, a lemming toy is moved in a complex scenario. In the first frame, due to the slow speed, the two algorithms can track the object dependently. But later the sparse representation based algorithm can not track from the 232nd frame due to the fast motion, and lost the object finally, while the proposed method can still track dependently and accurately as shown in Figure 4 and Figure 5 (c).

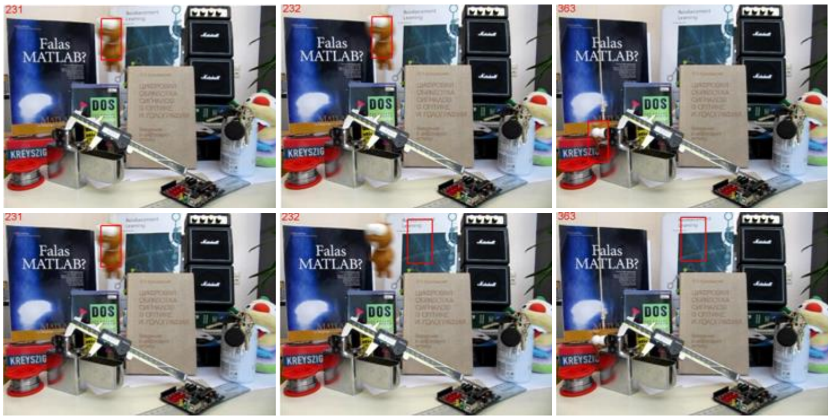
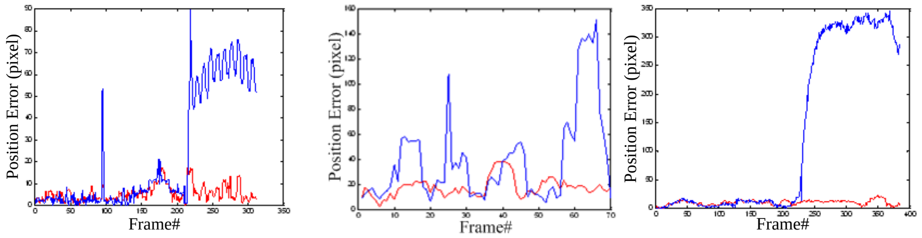


Figure 4. The Tracking Result Comparison between the Proposed Method (The 1st Row) and the Sparse Representation Based Tracking Method (The 2nd Row) In Frames 231, 232 and 363 of Lemming Sequence

The quantitative comparison between the previous two algorithms is to compare the center location error in pixels as in Figure 5.



a. Jumping b. Deer c. Lemming

Figure 5. The Quantitative Comparison of the Proposed Method (Red Line) With the Sparse Representation Based Tracking Method (Blue Line) On Jumping (A), Deer (B) and Lemming (C) Respectively

5. Conclusions

In the scenario of fast motion and blur, the performance of the sparse representation based visual tracking is not satisfactory. To improve performance, a kernel sparse representation based visual tracking is proposed, which has the following three characteristics: (1) The kernel trick is used to represent the non-linear relation between the candidate and templates in the scenario of fast motion and blur. (2) The APG algorithm is used to optimize the target function. (3) A simple strategy is used to update the target templates with the error rate. Experiments on three representative image sequences showed the effectiveness of the proposed algorithm comparing with the traditional sparse representation based visual tracking method in the scenarios of fast motion and blur. In the future, we will further improve the time efficiency of the kernel sparse representation based tracker.

Acknowledgements

This work is supported by Scientific Research Fund of Heilongjiang Provincial Education Department (NO: 12541238), Dr. Scientific Research Foundation of Harbin Normal University (KGB201216), the National Science Foundation of China (61173087), Heilongjiang Provincial University Engineering R&D Center of Machine Vision and Intelligent Detection, and Heilongjiang Provincial Education Department Key Laboratory of Intelligent Education and Information Engineering.

References

[1] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation", International Journal of Computer Vision, vol. 26, no. 1, **(1998)**, pp. 63-84.

[2] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-based object tracking", Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 25, no. 5, **(2003)**, pp. 564-577.

[3] 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. 1-3, **(2008)**, pp. 125-141.

[4] S. Avidan, "Support vector tracking", Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 26, no. 8, **(2004)**, pp. 1064-1072.

[5] R. T. Collins, Y. Liu and M. Leordeanu, "Online selection of discriminative tracking features", Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 27, no. 10, **(2005)**, pp. 1631-1643.

[6] B. Babenko, M. H. Yang and S. Belongie, "Robust object tracking with online multiple instance learning", Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 33, no. 8, **(2011)**, pp. 1619-1632.

[7] Q. Wang, F. Chen, J. Yang and W. Xu, "Transferring visual prior for online object tracking", Image Processing, IEEE Transactions on, vol. 21, no. 7, **(2012)**, pp. 3296-3305.

[8] J. Yin, Z. Liu, Z. Jin and W. Yang, "Kernel sparse representation based classification", Neurocomputing, vol. 77, no. 1, **(2012)**, pp. 120-128.

[9] J. Kwon and K. M. Lee, "Visual tracking decomposition", Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, **(2010)**, pp. 1269-1276.

[10] H. Grabner, C. Leistner and H. Bischof, "Semi-supervised on-line boosting for robust tracking", Computer Vision–ECCV 2008. Springer Berlin Heidelberg, **(2008)**, pp. 234-247.

[11] X. Mei and H. Ling, "Robust visual tracking using ℓ 1 minimization", Computer Vision, 2009 IEEE 12th International Conference on. IEEE, **(2009)**, pp. 1436-1443.

[12] Q. Wang, F. Chen, W. Xu and M. Yang, "Online discriminative object tracking with local sparse representation", Applications of Computer Vision (WACV), 2012 IEEE Workshop on. IEEE, **(2012)**, pp. 425-432.

[13] S. Zhang, H. Yao and S. Liu, "Robust visual tracking using feature-based visual attention", Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on. IEEE, **(2010)**, pp. 1150-1153.

[14] B. Liu, J. Huang, L. Yang and C. Kulikowsk, "Robust tracking using local sparse appearance model and k-selection", Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, **(2011)**, pp. 1313-1320.

[15] S. Zhang, H. Yao, X. Sun and S. Liu, "Robust object tracking based on sparse representation", Proceedings of the SPIE International Conference on Visual Communications and Image Processing. **(2010)**, pp. 77441N-77441N.

[16] H. Li, C. Shen and Q. Shi, "Real-time visual tracking using compressive sensing", Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, **(2011)**, pp. 1305-1312.

[17] W. Zhong, H. Lu and M. H. Yang, "Robust object tracking via sparsity-based collaborative model", Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, **(2012)**, pp. 1838-1845.

Authors

Xiping Duan, she was born in 1980. She received the M.S. degree from Harbin University of Science and Technology, Harbin, China, in 2006. She is now a Ph.D. candidate at School of Computer Science in Harbin Institute of Technology. Her current research interests include computer vision, pattern recognition and image processing.

Jiafeng Liu, he was born in 1968. He received the Ph.D. degree from Harbin Institute of Technology in 1996. He is currently an associate professor at Harbin Institute of Technology. His research interests include pattern recognition, image processing and computer vision.

Xianglong Tang, he was born in 1960. Professor of Harbin Institute of Technology, director of pattern recognition research center of HIT, senior member of China Computer Foundation. He received B.S. degree at HIT in 1982, M.S. degree at HIT in1986 and Ph.D. degree at HIT in 1995. Dr. Tang has published more than 80 papers in the fields of image processing, pattern recognition and artificial intelligence. His recent research interests include: medical imaging, intelligent motion analysis of human beings, handwriting character recognition and pattern recognition.