ATLANTIS PRESS
Advances in Computer Science Research, volume 44
3rd International Conference on Wireless Communication and Sensor Network (WCSN 2016)

A New Discriminative Tracking Method Applied in Multi-rotor Unmanned Aircraft

Gang Wu
School of Computer Engineering, JinLing Institute of Technology, 211169
Nanjing, China
E-mail: zdhxwg@jit.edu.cn

Xiao-Qin Zeng
College of Computer and Information, Hohai University, 210098
Nanjing, China
E-mail: xzeng@hhu.edu.cn

***Abstract*-Aiming at difficulties for vehicle tracking on the specific scenes such as fast motion, rotation, drastic illumination and scale change, a new discriminative tracking algorithm for moving vehicles is proposed in this paper. We incorporate low-rank sparse representation and dictionary learning with the classical particle filter algorithm. Based on unmanned multi-rotor aircraft, we apply the enhanced algorithm to track selected vehicle in the urban road, demonstrate the performance of our method on the process of vehicle tracking in above scenes. The proposed approach is different from conventional discriminative tracking algorithm. Compared with related methods, experimental results show that the proposed algorithm improves the synthesized efficiency of tracking process, the experiments based on standard testing videos demonstrate that tracking successful rate is significantly improved.**

Keywords-object tracking; online classifier; sparse representation; Particle filter; dictionary learning

I. INTRODUCTION

The main challenge of designing a robust visual tracker exists in the aspect of object appearance variations in sequential images. The design of flexible and reasonable tracking model which can adapt to aforementioned appearance variations is an important subject[1]. Some existing algorithms based on statistic principles were widely used in visual tracking domain for their effective experimental results on the controlled environments. However these statistic algorithms[2] are merely involved with the corresponding parameters based on local characteristic such as texture, edge and color histogram etc.. These algorithms usually fail to track the object accurately over some period of time, due to drastic change in the object appearance.

Sparse representations can provide a compact approximate encoding of a large set of images. These sparse representations denote a subset of the training images. Previous work on sparse representations has addressed on the subject about face recognitions and has rarely concentrated on the aspect of tracking object. Previous sparse approaches[2,3] have commonly tested on fixed image database such as YELU face database in object recognition. Although numerous vision-based recognition and trackers using sparse representation were proposed, sparse representations and online learning methods were only studied and applied in a very small portion of these methods [4,5]. Occlusion is one of the most challenging problems in object tracking process[6]. The tracker handles partial occlusion via $\ell 1$ minimization at the expense of high computational cost. Due to sampling from ambiguous region and the occluded target, the reconstruction errors of image regions are very likely to accumulate and finally cause tracking failure[7].

II. OUR TRACKING METHOD USING LOW-RANK SPARSE AND PARTICLE FILTER

In this paper, we pose object tracking process as binary classification problem where the primary task is to distinguish target image region from the background. Firstly, low-rank sparse representation and ℓ_1/ℓ_2 sparse coding are used to extraction of SIFT features, a spatial pyramid representation for each object image can be obtained. For subsequent efficient object tracking process, low-rank sparsity and constraint are exploited to learn robust linear representations corresponding to candidate particles.

A. Representation by Sparse Coding and Dictionary Learning

At time t, let $\mathbf{x}i$ denote the observation with respect to i-th particle, $X=[X_1,...,X_N]\in R^{m\times n}$ denote the corresponding SIFT descriptor extracted from object image observations, where m and n are the dimensionality of SIFT descriptor. Each observation from a dictionary is represented by $\mathbf{D}t$ = $[d1, d2, \ldots, dm]$, $\boldsymbol{X}0$ can be calculated by [8]:

$$\boldsymbol{X}_0 = \mathbf{D}tZt + \mathbf{E} \tag{1}$$

where the columns of $Z_t=[z_1,z_2,...,z_n]$ are the representations of particle observations with respect to dictionary $\mathbf{D}t$, $\mathbf{E}$ is the error due to noise such as occlusion and illumination. In our work, we solve the following equation efficiently using the learning about dictionary $\mathbf{D}t$..

$$\min_{a_i}\frac{1}{2}\|X_i-DZ_i\|_2^2+\lambda_1\|Z_i\|_1+\lambda_2\|Z_i\|_2^2+\lambda_3\|Z_i\|_* \tag{2}$$

where λ_1 , λ_2 and λ_3 are regularization parameters. We adopt nuclear norm $\|Z_i\|_*$ to minimize its convex envelope. Symbol $\|Z_i\|_*$ represents low-rank sparse representation in Eq.2.

B. Classifier Learning with Sparse Representation and Multi-scale Max Pooling

To initialize the classifier, we collect some target images as positive samples, and background images as negative samples in the preprocessing stage. Compared with raw image features, it is easier to separate the target object from the background with low-rank sparse representation and dictionary. In this paper, we adopt a feature pooling function to better describe object level feature for target samples. The feature pooling function operates on each row of sparse coefficient matrix:

$$b_i = \max\{Z_{i,1},...,Z_{i,n}\} \tag{3}$$

The max pooling[9] is well demonstrated with biophysical evidence in visual cortex, we adopt the following multi-scale max pooling to preserve spatial information and geometric shape:

$$z = [b_1^{\mathrm{T}},...,b_{\mathrm{M}}^{\mathrm{T}}]^{\mathrm{T}} \tag{4}$$

The training image set is composed of positive and negative samples, we draw positive and negative samples around the labeled target location in this paper. Using the above sparse coding of each image patch, positive and negative sample sets are to be computed to form the training data denoted by $\{Z_i, y_i\}_{i=1}^{\mathrm{M}}$, where M is the number of training samples, $Z_i \in \mathrm{R}^{n+2d}$, class label $y_i \in \{+1,-1\}$. After extracting SIFT descriptors, we effectually obtain a set of training data representing target and background images by sparse coding and multi-scale max pooling. With the training data, The linear classifier is learned by minimizing the following cost function[10]:

$$J(w) = \frac{1}{\mathrm{M}}\sum_{i=1}^{\mathrm{M}} \ell(y_i, w, z_i) + \frac{\lambda}{2}\|w\|_2^2 \tag{5}$$

where ℓ is a loss function, λ is the number of training samples, Parameter λ controls the strength of the regularization term, symbol w denotes the classifier parameter set we want to learn. The corresponding classifier score with the learned classifier can be computed by[11]:

$$f(z) = \frac{1}{1+e^{-w^{\mathrm{T}}z}} \tag{6}$$

III. ONLINE TRACKING PROCESS

Generally, we use two main steps to obtain the tracking result. Firstly, we use the online classifier to estimate the most probable tracking location by computing the corresponding classifier score, and form target positive samples and background negative samples. Secondly, we use particle filter with the recent observation data and associated parameters to determine the final tracking result. Fig.1 shows the main parts of our tracking algorithm.

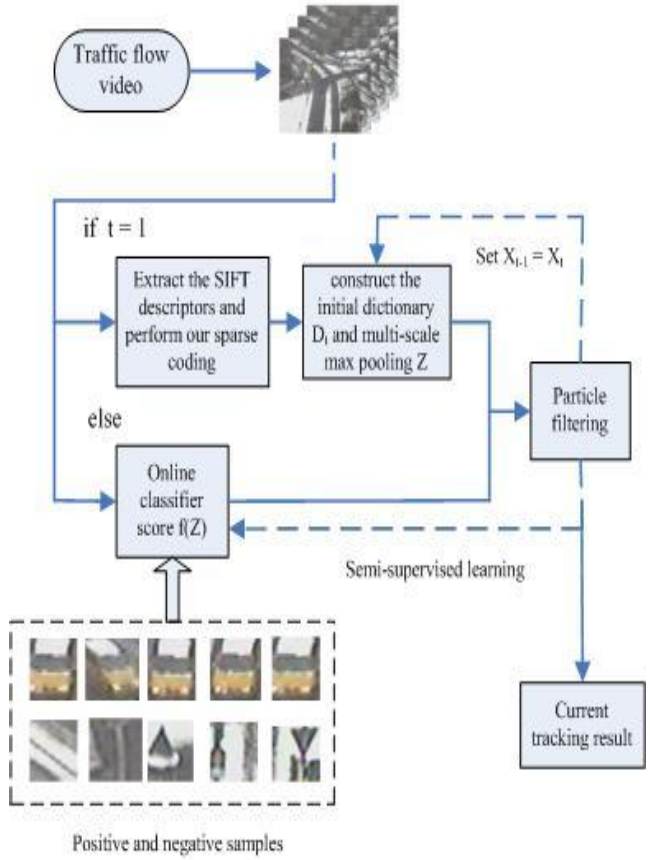


Figure 1. Our tracking algorithm.

Our algorithm is summarized as follows:

for t = 1,…,T do
  if t = 1 then
    Extract the SIFT descriptors from overlapped patches of raw images, then perform sparse coding by Eq.1, and construct the initial dictionary $\mathbf{D}_t$ and multi-scale max pooling Z. Initialize the online classifier with Z.
  else
- Collect target positive samples and background negative samples, form the training data denoted by $\{Z_i, y_i\}_{i=1}^{\mathrm{M}}$. Minimize the cost function $J(w)$, and acquire classifier score $f(Z)$.
- Perform particle filtering to estimate the target state x_t , find the most probable state S_t by using the previous tracking result $y_{1:t-1}$, compute the weights $\{w_i^t\}_{i=1}^{\mathrm{N}}$ by particle filer algorithm. Adaptively update the observation model.
- Set $X_{t-1} = X_t$, Timely update the dictionary $\mathbf{D}_t$ and multi-scale max pooling Z . Plot the tracking result in the current image.

  end if
end for

IV. EXPERIMENTS

A. *Image Set*

We evaluate the proposed tracker on the most challenging image sequences including dtneu_nebel, dtneu_schnee, dtneu_winter, rheinhafen, tas_demo1-2 ,Car4, Car11, etc., which are publicly available on public test database[12]. The challenging factors include heavy occlusion, partial occlusion, illumination change, scale change, rotation, background clutter, fast motion and motion blur (As shown in Table I). On the patch level, small image blocks share structural similarity. We exploit prior and online information from testing data sets and use it for vehicle tracking process. These data sets consist of a large variety of objects which are common in surveillance scenarios. These objects are used for object tracking in many comparative experiments, including person, face, truck, bus and car. We use object classes from above data sets. It should be noted that these object images can also be used to learn a prior for specific tracking tasks.

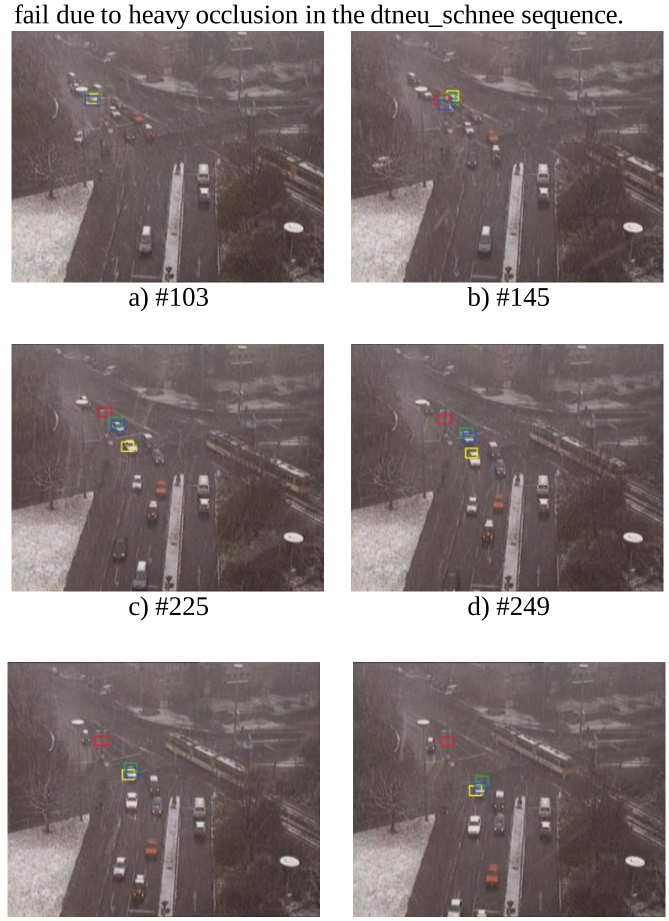
TABLE I. EVALUATED IMAGE SEQUENCES

Sequence	Frames	SEQUENCES
dtneu_nebel	459	heavy occlusion, scale changes, background clutter
dtneu_schnee	398	partial occlusion, background clutter
dtneu_winter	672	partial occlusion, background clutter
rheinhafen	1546	fast motion, scale changes
kwbB	1976	heavy occlusion, scale changes
tas_demo1	1221	illumination variation, scale changes
tas_demo2	1674	motion blur, fast motion
Car4	659	illumination variation, scale changes
Car11	393	illumination variation, background clutter, scale changes
Pet2000	265	rotation, scale changes
test1	1544	fast motion, scale changes
test2	1356	fast motion, background clutter
AVSEQ01	121	fast motion, scale changes

B. *Qualitative Evaluation*

We evaluate the proposed tracker on the most challenging image sequences including dtneu_nebel, dtneu_schnee , rheinhafen , etc., which are publicly available on public test database[13]. With our formation in Eq. 1, the dimensionality of z_t and $\mathbf{E}$ is 16 and 1024 respectively. The parameters of the proposed tracking algorithm are fixed in all experiments. For sparse coding and dictionary learning, the parameters $\lambda 1$, $\lambda 2$ and $\lambda 3$ in Eq. 2 are set to be 0.1, 0.5 and 5. In the offline training phase, we resize the target image patch to 32×32 pixels and extract overlapped 16×16 local patches with 8 pixels as step length, the SIFT descriptors are densely extracted from 16×16 patches from each selected image. For the online classifier, parameter f in Eq.6 is initialized with 50 positive templates and 200 background negative templates in the first frame. The classifier parameter w in Eq.5 is updated every 5 frames. As a trade-off between computational efficiency and effectiveness[14], 600 particles are used in particle filter and the proposed tracker is incrementally updated every 5 frames in all the experiments.

We compare with related tracking algorithms including the ST[15], LRT [1], LRST [8],ℓ1 [5] and IST [16] methods. We present some representative results in this section. The proposed algorithm is implemented in MATLAB which runs at 4 frames per second on PC with Pentium 3.4 GHz i7 CPU and 32GB memory. The tracking results are shown in Fig.2, where the yellow, green, blue and red line box respectively represent the tracking result of ST, LRT, LRST and our tracker. As shown in Fig.2 a), the ST, LRT, LRST trackers and the proposed tracker perform well at frame 103. For sparse-based trackers, simple update without dealing with occluded regions often leads to drifts. When the vehicle is heavily occluded at frame 145, the ST tracker updates the trivial templates with a straightforward scheme, the tracking results are less accurate. The LRT tracker uses a sparse error matrix to handle the occlusions. The LRST tracker is less effective in dealing with heavy occlusion and drifts away at frame 249. At frame 277, the LRST tracker locates the target well but deals with heavy occlusion less effectively. From frame 103 to frame 300, all the trackers except our tracker fail due to heavy occlusion in the dtneu_schnee sequence.



a) #103 b) #145

c) #225 d) #249

e) #277 f) #300

Figure 2. Results of 4 kinds of tracker on standard data source dtneu_schnee

The successful tracking rates and average center location errors of all these trackers are respectively listed in Table II, Overall, the tracking data denote that our tracker achieves favorable results against other methods.

TABLE II. AVERAGE CENTER ERRORS (IN PIXELS) AND SUCCESS RATE

sequences	LRT		ST		LRST	
	error	**rate**	**error**	**rate**	**error**	**rate**
dtneu_nebel	16.6	0.48	53.2	0.22	8.5	0.68
dtneu_schnee	5.2	0.80	8.7	0.58	4.1	0.82
dtneu_winter	15.3	0.53	24.5	0.42	10.5	0.57
rheinhafen	8.3	0.60	9.6	0.58	6.4	0.78
kwbB	15.3	0.50	32.1	0.40	10.2	0.62
tas_demo1	21.2	0.49	87.0	0.30	32.2	0.44
tas_demo2	7.3	0.79	7.5	0.62	6.2	0.77
Car4	10.3	0.62	17.5	0.52	7.4	0.80
Car11	11.4	0.60	21.3	0.44	7.6	0.79
Pet2000	11.9	0.58	17.9	0.47	8.2	0.77
test1	15.1	0.50	40.2	0.22	10.4	0.70
test2	8.1	0.69	9.0	0.58	6.4	0.70
AVSEQ01	7.6	0.80	9.7	0.56	1.6	0.92
sequences	**ℓ1**		**IST**		**Ours**	
	error	**rate**	**error**	**rate**	**error**	**rate**
dtneu_nebel	47.6	0.34	8.2	0.70	4.3	0.76
dtneu_schnee	9.2	0.54	3.1	0.87	1.8	0.90
dtneu_winter	17.3	0.47	5.7	0.77	4.0	0.78
rheinhafen	8.6	0.68	2.7	0.87	1.9	0.90
kwbB	26.2	0.48	6.5	0.70	4.4	0.76
tas_demo1	64.1	0.43	8.9	0.69	5.2	0.70
tas_demo2	5.2	0.80	4.5	0.80	4.1	0.83
Car4	15.6	0.50	5.0	0.84	2.7	0.86
Car11	18.6	0.52	5.1	0.82	2.9	0.84
Pet2000	18.3	0.45	2.6	0.85	6.3	0.75
test1	38.5	0.38	6.5	0.73	3.5	0.81
test2	9.1	0.57	3.2	0.82	1.5	0.92
AVSEQ01	9.5	0.57	2.6	0.86	4.2	0.85

V. CONCLUSION

The contributions of this work are two-fold. Firstly, based on online classifier learning using low-rank sparse representation, we effectively combine the online classifier and particle filter to learn target features in the occlusion scene, and effectively separate the specified moving vehicles from urban traffic flow data. Secondly, the sparse representations of observations are represented jointly rather than independently, and learned jointly by considering all particles, solving one rank minimization problem by a sequence of closed form operations. In this paper, the performance of particle filter is enhanced by integrating low-rank sparse representation, sparse coding and dictionary learning into conventional sequential Monte Carlo framework. Simultaneously, online classification are exploited to handle new observations in the vision tracking process, particle filter tracking process reflects the classifier learning result on low-rank sparse representation.

ACKNOWLEDGMENT

This work was sponsored by JinLing Institute of Technology (High level talent introduction project, No.jit-rcyj-201508). As a postdoctoral, we thank Hohai University Professor Zeng guide the work of this article. We also thank Professor Su and Lu for assisting on the subject (National Natural Science Foundation of China, No. 61375121, No.61305011).

REFERENCES

[1] GS Walia, R Kapoor, “Recent advances on multicue object tracking: a survey [J] ,” Artificial Intelligence Review, vol. 46(1) , 2016, pp. 1-39.

[2] Hsu-Yung Cheng, Jenq-Neng Hwang , “Integrated video object tracking with applications in trajectory-based event detection [J] ,” Journal of Visual Communication and Image Representation, vol. 22(7) , 2011, pp. 673–685.

[3] Yanwen Chong, Wu Chen, Zhilin Li, et al, “Integrated real-time vision-based preceding vehicle detection in urban roads [J],” Neurocomputing, vol. 116, 2013, pp. 144-149.

[4] Zhenjun Han, Jianbin Jiao, Baochang Zhang, et al, “Visual object tracking via sample-based Adaptive Sparse Representation (AdaSR) [J] ,” Pattern Recognition, vol. 44(9) , 2011, pp. 2170-2183.

[5] Mei, X., Ling, H. , “Robust visual tracking and vehicle classification via sparse representation [J] ,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33(11) , 2011, pp. 2259–2272.

[6] J Sun, S Zhang, L Zhang, “Object Tracking With Spatial Context Model [J] ,” IEEE Transactions on Signal Processing Letters, vol. 23(5) , 2016, pp. 727-731.

[7] Zhang, T.,Ghanem, B., Liu, S., Ahuja, N. , “Robust visual tracking via structured multi-task sparse learning[J] ,” International Journal of Computer Vision, vol. 101(2) , 2013, pp. 367–383.

[8] J. Yang, K. Yu, Y. Gong, and T. Huang, “Linear spatial pyramid matching using sparse coding for image classification [C] ,” IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1794–1801.

[9] Y Yin, D Xu, X Wang, M Bai, “Online State-Based Structured SVM Combined With Incremental PCA for Robust Visual Tracking [J].” IEEE Transactions on Cybernetics, vol. 45(9), 2015, pp. 1988-2000.

[10] Salti, S., Cavallaro, A., Stefano, L. D. , “Adaptive appearance modeling for video tracking: Survey and evaluation [J] ,” IEEE Transactions on Image Processing, vol. 21(10), 2012, pp. 4334–4348.

[11] L Cehovin, A Leonardis, M Kristan, “Visual Object Tracking Performance Measures Revisited[J] ,” IEEE Transactions on Image Processing, vol. 25(3) , 2016, pp. 1261-1274.

[12] K Muthuswamy, D Rajan, “Particle filter framework for salient object detection in videos [J] ,” IET Computer Vision, vol. 9(3), 2015, pp. 428-438.

[13] Gang Wu. "A New Resampling Strategy about Particle Filter Algorithm Applied in Monte Carlo Framework[C]," 2009 Second International Conference on Intelligent Computation Technology and Automation. 2009 October 10-11: Changsha, Hunan, China, vol.1, pp.507-510.

[14] Tianzhu zhang, Si Liu, Narendra Ahuja, et al, “Robust visual tracking via consistent low-rank sparse learning[J]”, Int Comput Vis, vol.111(2), 2015, pp.171-190.

[15] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang. "Robust Object Tracking via Sparse Collaborative Appearance Model[J]," IEEE Transactions on Image Processing, vol.23(5), 2014, pp.2356-68.

[16] Qing Wang, Feng Chen, Jimei Yang, Wenli Xu, and Ming-Hsuan Yang. "Transferring Visual Prior for Online Object Tracking[J]", IEEE Transactions on Image Processing, vol.21(7), 2012, pp. 3296-305..