Visual tracking via temporally smooth sparse coding

Liu, Ting; Wang, Gang; Wang, Li; Chan, Kap Luk

2014

Liu, T., Wang, G., Wang, L., & Chan, K. L. (2015). Visual tracking via temporally smooth sparse coding. IEEE signal processing letters, 22(9), 1452-1456.

https://hdl.handle.net/10356/107462

https://doi.org/10.1109/LSP.2014.2365363

Visual tracking via temporally smooth sparse coding

Ting Liu, Gang Wang, Li Wang and Kap Luk Chan

***Abstract*—Sparse representation has been popular in visual tracking recently for its robustness and accuracy. However, for most conventional sparse coding based trackers, the target candidates are considered independently between consecutive frames. This paper shows that the temporal correlation of these frames can be exploited to improve the performance of tracking and makes the tracker more robust to noise. Furthermore, to improve the tracking speed, we revisit a more efficient method for ℓ_1 norm problem, marginal regression, which can solve the sparse coding problem more efficiently. Consequently we can realize real-time tracking based on the temporal smooth sparse representation. Extensive experiments have been done to demonstrate the effectiveness and efficiency of our method.**

***Index Terms*—Sparse representation, marginal regression, temporal smoothness, visual tracking.**

I. Introduction

Visual tracking is an important technique in computer vision with various applications such as security and surveillance, human computer interaction and auto-control systems [1–3]. With the development of single object tracking method, most of the tracking tasks in simple environment with slow motion and slight occlusion can be solved well by current algorithms. However, in more complicated situations more robust and faster tracking methods are required to realize real-time and accurate tracking.

Recently, the sparse representation has been approved to be robust against partial occlusions [4–13], which leads to improved tracking performance. However, the conventional sparse coding based trackers treat the candidates independently between consecutive frames. Apparently, the information of neighbouring frames should be helpful for the stability of tracking results, because the targets usually change slightly between adjacent frames in the tracking sequences. We should take advantage of the temporal correlation to enhance the trackers. In this paper, we propose a temporally smooth sparse coding method to model temporal correlation for tracking, in the sparse coding framework. Furthermore, sparse coding based trackers need to perform ℓ_1 minimization. Currently most sparse coding based tracker [14–20] are using the Lasso [21] to solve the ℓ_1 norm related problems which is computationally expensive, especially for tracking applications. Hence, developing an efficient solver for the ℓ_1 minimization has been a key to make the sparse coding based tracker useful.

Motivated by [18] and recent progress of marginal regression which is a much older and computationally simpler method, we propose a novel tracking algorithm, called Temporal smooth Tracker via Marginal Regression (TMRT), which explicitly considers the relationship of neighbouring frames and take advantage of marginal regression to dramatically improve the tracking speed. The experimental results show that our tracker can achieve high speed and high accuracy.

The authors are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. Gang Wang is also with Advanced Digital Sciences Center, University of Illinois, Singapore 138632 (e-mail: liut0016@e.ntu.edu.sg; wanggang@ntu.edu.sg; wa0002li@e.ntu.edu.sg; eklchan@ntu.edu.sg).

II. Temporally smooth tracking based on sparse representation

A. The basic sequential inference model

The visual tracking problem is usually carried out as an inference task in a Markov model with hidden state variables. Let x_t denote the state variable describing the parameters of an object at the time t (e.g. location or motion parameters) and define $Y_t = [y_1, y_2, \cdots, y_t]$ as a set of observed video frames. The optimal state $\hat{x}_t$ is computed by the maximum a posterior (MAP) estimation

$$\hat{x}_t = \arg\max_{x_t^i} p(x_t^i | Y_t) \tag{1}$$

where x_t^i is the state of the i-th frame. Using Bayes' theorem, we have

$$p(x_t|Y_t) \propto p(y_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|Y_{t-1}) dx_{t-1} \tag{2}$$

The $p(y_t|x_t)$ is the observation model.

The dynamics between states in this space is usually modelled by the Brownian motion. Each parameter in x_t is modelled independently by a Gaussian distribution given its counterpart in x_{t-1}.

$$p(x_t|x_{t-1}) = \mathcal{N}(x_t, x_{1:t-1}, \Psi) \tag{3}$$

where Ψ is a diagonal covariance matrix whose elements are the corresponding variances of affine parameters. The observation model $p(y_t|x_t)$ denotes the likelihood of the observation y_t at state x_t. $p(y_t|x_t)$ is proportional to the similarity between the candidate and the target.

B. Sparse representation in tracking

Our method is developed based on the sparse representation method [18], hence we first give it a brief review. For the dictionary construction, we have N target templates $T = [T_1, T_2, \cdots, T_N]$. Then M overlapped local image patches are extracted from the target region of each template based on a spacial layout. These local patches consist the dictionary, $D = [d_1, d_2, \cdots, d_{(N\times M)}] \in \mathbb{R}^{X\times(N\times M)}$. Here X means the dimension of the image patch vector. Each local patch represents one fixed part of the target object. The target candidates are also decomposed into a collection of local patches, $Y = [y_1, y_2, \cdots, y_M] \in \mathbb{R}^{X\times M}$. The dictionary captures the commonality of different templates and is able to represent various forms of these parts. The target candidates

are represented by the patches from templates using sparse codes. With the sparsity assumption, the local patches can be represented as the linear combination of only a few basis elements of the dictionary by solving

$$\min_{b_i} \|y_i - Db_i\|_2^2 + \lambda\|b_i\|_1, \quad s.t. \quad b_i \geqslant 0 \tag{4}$$

where $b_i \in \mathbb{R}^{(N\times M)\times 1}$ is the corresponding sparse code of that local patch y_i, and all the elements of b_i are nonnegative. Note $\beta = [b_1, b_2, \cdots, b_M]$ represents the sparse codes of one candidate. The sparse coefficients of each local patch are divided into several segments, according to the template that each element of the vector corresponds to.

After obtaining the coefficients b_i from Eq. 4, these segmented coefficients are summed to obtain v_i for the i-th patch through $\sum_{k=1}^{N} b_i^{(k)}$. To improve the robustness of our tracker, we take advantage of the structural relationship of the local patches as in [18]. For the vector v_i, each local patch of the candidate is represented by the patches at the same positions of the templates. Hence the feature is represented as $f = diag(V)$. Based on II-A, the observation model $p(y_t\,|x_t)$ is proportioned to $\sum_{k=1}^{M} f_k$. Refer to [18] for more details.

C. Temporally smooth tracking

As described above, the conventional sparse coding based tracking methods infer b_i independently. Obviously, in tracking videos, neighbouring frames are presumably more related to each other than frames that are farther apart. Hence, we propose a mechanism to incorporate such feature similarity and temporal information into the framework of sparse coding tracking, leading to a sparse representation with an improved tracking accuracy. We propose the formulation based on the framework of sparse coding tracking methods.

$$\min_{b_{ri}^t} \left\|y_{ri}^t - Db_{ri}^t\right\|_2^2 + \lambda\left\|b_{ri}^t\right\|_1 + \gamma\left\|b_{ri}^t - b_{r^*i}^{t-1}\right\|_2^2, \tag{5}$$

where y_{ri}^t denotes the i-th local patch of the r-th candidate at the frame t; b_{ri}^t is the corresponding sparse code at the frame t; $b_{r^*i}^{t-1}$ is the selected result from the frame $t-1$. The scalar γ controls the tradeoff between the temporal smoothness and sparse constraint. Traditional sparse coding trackers minimizes the reconstruction error of the encoded samples. Our proposed method TMRT, on the other hand, minimizes the reconstruction of encoded samples as well as the difference between neighbouring frames. In the t-th frame, we actually select a candidate whose sparse code is similar to that of the previous frame's result and has low reconstruction error. As a result, our tracker can be more robust to noise at individual frames.

Compared with traditional sparse based tracking methods, we have an extra pairwise constraint term. Thus, it is difficult to minimize directly. We adopt a two-step algorithm to solve the optimization problem.

We first utilize marginal regression to solve the sparse coding for each frame (II-D); then based on the obtained sparse codes, we optimize the smooth part of the objective function. The sparse codes b_{ri}^t for the t-th frame are calculated by comparing the similarity with previous tracking results. In the second step, in addition to the smooth constraint term $\left\|b_{ri}^t - b_{r^*i}^{t-1}\right\|_2^2$, we also consider the reconstruction error obtained from the first step. The reason is that in many situations, such as heavy occlusion, if we ignore the reconstruction error, the tracking results may drift to the occluded object which can optimize $\left\|b_{ri}^t - b_{r^*i}^{t-1}\right\|_2^2$. Hence, we also consider the impact of $\|y_{ri}^t - Db_{ri}^t\|_2^2$. In our experiments, when the value of γ is between 10 to 50, we can get similar results. In our experiments, we just simply set γ as 10.

Our method is illustrated with an example as shown in Fig. 1. There are two candidates (expressed as red and blue boxes) at frame 167 and 168. The feature codes of the candidates are shown on the right with corresponding colors. Without temporal smoothness constraint, the regions in blue will be considered as the target according to ASLA tracking method. However, it will result in "drifting" of the target. In fact, when we consider the temporal smoothness between consecutive frames, regions in red will be considered as the targets based on the Eq. 5.

Because the computation of step two is simple, the main time consuming part is solving sparse coding. Next we introduce marginal regression to speedup the sparse coding optimization.

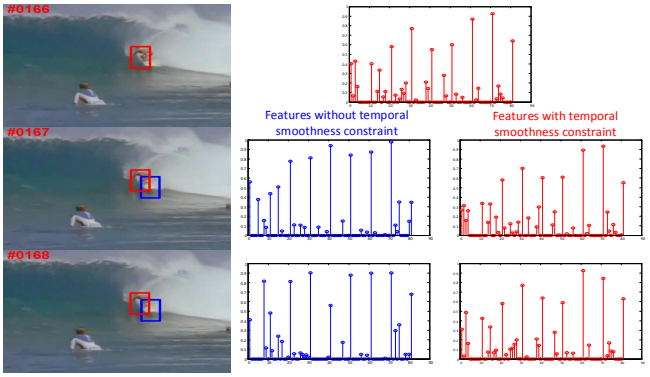

Fig. 1: Comparison of tracking performance obtained by trackers without and with temporal smoothness constraint. Because of the sparse coding noise, there is an apparent drift of the blue boxes that are yielded from method without temporal smoothness constraint. The red boxes are obtained from the proposed method. By considering the temporal correlation, the red features are more similar to neighbouring ones and red boxes track the target more accurately.

D. Marginal regression for sparse solution

The speed of sparse coding based trackers is usually slow. In the past, several methods have been proposed to improve the speed, however it is still too slow for practical applications. Recently the Lasso [21, 22] is a popular tool for ℓ_1 optimization. Similar to other ℓ_1 optimization solutions, Lasso has complicated operations. The gradient descent algorithms

and LARS algorithm [23] for Lasso require $O(p^3 + np^2)$ operations. To overcome this limitation, we show how marginal regression can be used to obtain sparse codes faster.

Consider a regression model, $Y = D\beta + z$, where $Y = (Y_1, Y_2, \cdots, Y_n)^T$, D is a $n \times p$ design matrix, coefficients $\beta = (\beta_1, \beta_2, \cdots, \beta_p)^T$ and $z = (z_1, z_2, \cdots, z_n)^T$ is the noise variable. For sparse coding, the main problem is variable selection: determining which components of β are non-zero.

Marginal regression [24] (also called correlation learning, simple thresholding [25], and sure screening [26]) is an efficient method for variable selection in which the outcome variable is regressed on each covariate separately and the resulting coefficient estimates are screened. To compute the marginal regression estimates for sparse coding, we begin by computing the marginal regression coefficients, assuming D has been standardized, we calculate:

$$\hat{\alpha} = D^T Y \tag{6}$$

Then, we threshold $\hat{\alpha}$ using a parameter $\tau > 0$:

$$\hat{\beta}_i = \begin{cases} \hat{\alpha}_i & |\hat{\alpha}_i| \geqslant \tau \\ 0 & otherwise \end{cases} \tag{7}$$

We sort these coefficient in terms of their absolute values, and select the top largest coefficients whose ℓ_1 norm is bounded by τ. The thresholding parameter τ is selected by cross validation. In our experiments, we set τ as 0.8.

This procedure requires just $O(np)$ operations. When p is much larger than n, marginal regression provides two orders of magnitude speedup over Lasso. This is a significant advantage of marginal regression because it is now tractable for much larger problems. [26] and [27] introduce more details about the comparison of marginal regression and Lasso. Hence, the marginal regression method improve the speed of the tracker significantly by replacing the Lasso ℓ_1 optimization. In addition, a template update scheme is adopted in [28] to overcome pose and illumination changes.

III. Experiments

To evaluate our proposed trackers, we compile a set of 22 challenging tracking sequences. Four of the sequences are shown in Fig. 2. These videos are recorded in indoor and outdoor environments and have variations of rotation, illumination change, occlusion, etc. Besides the baseline tracker ASLA [18], we also test the MRT tracker, which uses marginal regression instead of Lasso for the baseline method. We compared the proposed algorithms TMRT with nineteen state-of-the-art visual trackers: Frag[29], BSBT [30], LOT [31], CT [32], SMS [33], KMS [34], CPF [35], DFT [36], ORIA [37], IVT [28], CSK [38], CXT [39], TLD [40], VTD [41],ℓ_1APG [19], MTT [15], SCM [14], LSST [42], ASLA [18]. All our experiments are performed using MATLAB R2012b on a 3.2 GHZ Intel Core i5 PC with 16 GB RAM. We resize the target image patch to 32×32 pixels and extract 3×3 overlapped local patches within the target region. For all experiments, we set the number of particles as 600, the total number of target templates as 10. For fair comparison, we use the source codes provided by the authors. They were initialized using their default parameters. For the IVT, ℓ_1APG, MTT, SCM, LSST, ASLA and our proposed methods, we used the same affine parameters (x, y translation, rotation angle, scale, etc.) for each sequence in candidates sampling.

To assess the performance of the proposed tracker, two criteria, the center location error as well as the overlap rate, are employed in our paper. A smaller average error or a bigger overlap rate means a more accurate result. Given the tracking result of each frame R_T and the corresponding ground truth R_G, we can get the overlap rate by the PASCAL VOC [43] criterion, $score = \frac{area(R_T \cap R_G)}{area(R_T \cup R_G)}$. Table I and II report the quantitative comparison results respectively.

We first compare the tracking accuracy of the proposed TMRT tracker to that of the same tracker without smoothing constraint (MRT). From Table I and II, we can see that our tracker with temporal smoothing performs better than MRT and ASLA. It proves that the marginal regression can obtain similar convergence result for ℓ_1 norm problem compared with Lasso; and the temporal smoothing gains extra accuracy compared to conventional sparse coding based tracking methods. As shown in the tables, the proposed tracker yields favorable performance against other state-of-the-art methods.

We compare our method with other sparse coding based trackers on speed. The computational cost of these methods is huge due to the Lasso method or APG for ℓ_1 norm, even though they employ the C language to improve the solution speed as toolbox. Due to the inherent similarity between these sparse coding based L1 tracker and the proposed tracker (TMRT), we compare their average runtimes in Table III. Based on the results, it is clear that our trackers TMRT is much more efficient than the other sparse coding based L1 trackers. Besides the mentioned sparse coding based L1 trackers, we also test the speed of IVT, which is the baseline of most particle filter based tracking methods. IVT can run at 16fps in our computer. Our proposed method is only 1fps slower than IVT, while getting much better performance with sparse feature and temporal smoothness constraint.

TABLE III: Running speed comparison of several popular sparse coding based L1 trackers

L1	$\ell_1$apg	MTT	SCM	ASLA	LSST	TMRT
0.1fps	1fps	2fps	0.5fps	1.5fps	5fps	15fps

We also plot the results of IVT, TLD, L1APG, CXT, CSK, MTT, VTD, SCM, ASLA, LSST and TMRT trackers in visualization comparison. Due to the pages limitation, we only select four of 22 sequences for visualization and analysis.

In the david sequence, a moving face is tracked. The tracking results are shown in Fig. 2(a). LSST fails at frames 402. L1APG, MTT and CSK start to drift around frame 467. The other trackers track the moving face accurately.

As shown in Fig. 2(b), a stuffed animal is being moved around on the Sylvser sequence. LSST, L1APG and IVT trackers fail around frame 450, 597 and 928 respectively. Our method track the target throughout the sequence.

In the Jumping sequence, the tracked object is subject to fast location changes when the man is skipping rope. Most methods fail to track the object when the man jump quickly.

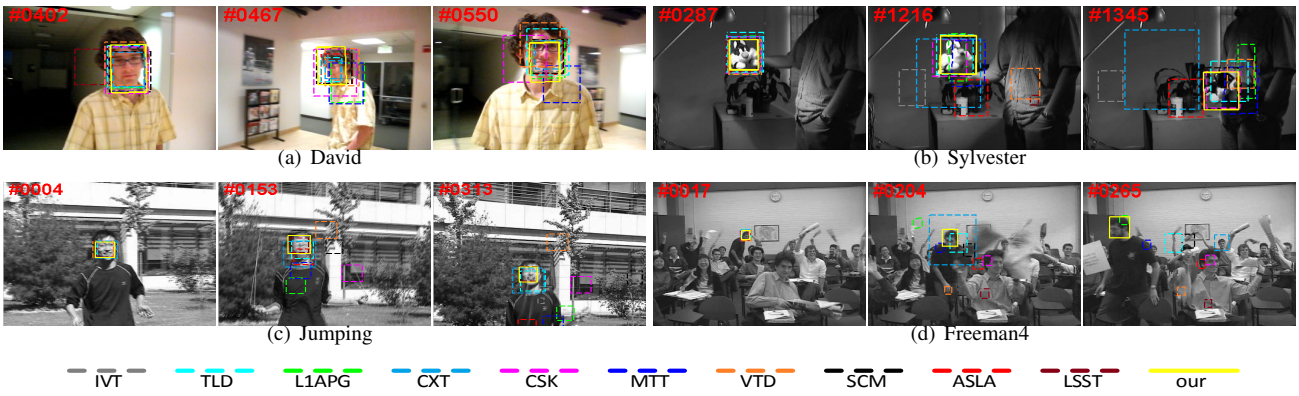

Fig. 2: Comparison of our approach with state-of-the-art trackers in challenging situations.

TABLE I: Average center location error (in pixels). The best three results are shown in red, blue, and green fonts.

Sequence	FragT	BSBT	LOT	CT	SMS	KMS	CPF	DFT	ORIA	IVT	CSK	CXT	TLD	VTD	$\ell_1$apg	MTT	SCM	LSST	ASLA	MRT	TMRT
Car4	179.8	57.7	165.6	229.7	140.2	52.9	38.7	61.9	237.4	2.9	19.1	49.5	18.8	12.3	16.4	37.2	3.8	2.9	4.3	4.5	2.7
Car11	63.9	8.1	30.8	78.0	88.9	30.3	43.8	21.6	26.8	2.1	3.2	15.9	25.1	27.1	1.7	1.8	1.8	1.6	2.0	2.1	1.3
David	76.7	20.9	23.9	12.8	25.1	20.3	26.7	27.5	23.3	3.6	17.6	6.0	9.7	13.6	10.8	13.4	3.4	4.3	3.6	3.7	3.4
Occ1	4.6	14.5	34.7	19.9	23.1	19.2	28.8	22.6	21.3	16.3	11.9	22.6	17.6	11.1	6.8	14.1	3.2	5.3	10.8	12.8	4.2
Occ2	15.5	15.6	14.9	12.8	29.3	28.9	21.0	7.9	5.5	10.2	5.9	6.3	18.6	10.4	6.3	9.2	4.8	3.1	3.8	4.3	4.0
Sylvester	15.0	14.7	11.3	8.6	17.8	18.1	12.8	44.9	9.3	34.2	9.9	14.8	10.5	19.6	26.2	7.5	8.0	69.6	14.6	13.1	7.2
Singer1	22.0	76.4	127.4	13.7	8.7	53.1	7.7	18.8	8.1	8.5	14.0	11.3	32.7	4.1	3.1	41.2	3.8	3.5	4.8	5.2	2.7
Skating1	144.2	57.9	110.5	150.4	230.1	89.7	118.3	168.3	69.6	19.3	7.8	129.7	96.0	9.3	101.5	219.2	16.3	17.8	6.0	5.9	5.6
Woman	111.9	33.2	117.1	113.6	97.6	13.9	83.8	9.5	212.2	167.5	207.3	72.5	47.9	118.5	128.9	137.3	7.9	118.5	2.8	3.6	2.5
Subway	15.8	81.0	4.7	11.1	139.1	117.0	80.1	13.5	120.1	130.8	159.5	129.1	79.9	141.3	145.2	157.1	3.5	117.9	63.5	68.9	3.5
Walking2	54.3	41.5	58.8	58.5	78.6	43.9	49.9	29.1	20.0	2.5	17.9	30.4	24.3	44.0	5.1	4.0	1.6	53.2	21.4	22.5	1.9
Caviar	94.2	79.6	43.9	65.5	11.1	19.3	20.9	89.1	76.6	66.2	69.1	72.6	53.0	60.9	68.6	67.5	2.2	3.1	2.3	2.2	2.0
Freeman4	47.3	46.7	38.6	93.0	105.9	36.0	62.0	57.5	54.5	43.0	64.7	65.2	39.2	60.8	22.1	23.5	37.7	72.3	45.5	46.7	3.5
Tiger1	74.0	64.0	111.4	29.9	45.8	69.6	37.3	19.0	87.0	93.2	50.4	45.4	49.5	102.3	58.4	59.1	75.4	86.3	55.9	52.2	18.0
Deer	50.4	27.5	65.2	10.5	78.6	43.8	79.6	98.7	149.2	127.5	5.0	6.7	25.7	11.9	38.4	9.2	36.8	10.0	8.0	7.9	4.2
Motorbike	196.7	41.8	24.9	187.1	136.1	54.4	211.0	20.2	7.1	7.7	6.5	159.6	195.2	9.8	8.3	7.1	10.6	132.2	9.7	10.2	4.9
Biker	76.4	96.6	65.2	21.1	14.1	95.2	74.4	122.6	83.5	17.6	27.3	83.0	78.3	13.2	71.8	29.2	99.8	47.3	19.0	20.1	13.1
Football	16.9	32.6	18.3	11.6	190.2	87.1	15.8	10.3	14.2	42.5	16.2	12.8	11.8	4.1	12.4	6.5	10.4	7.6	18.0	16.5	7.8
Jumping	58.4	33.5	5.6	53.0	37.4	84.2	56.1	65.6	76.1	36.8	49.1	7.0	3.6	63.0	8.8	19.2	3.9	4.8	39.1	42.3	4.5
Board	18.3	197.1	162.2	52.2	20.9	20.5	35.8	123.0	185.6	84.4	18.0	154.5	127.9	52.8	183.3	139.8	32.2	20.1	7.3	7.9	7.2
Surfer	22.8	40.0	14.8	17.2	15.5	19.8	25.2	42.2	48.7	84.3	157.5	14.9	25.5	33.9	56.8	30.2	62.1	146.9	51.9	50.6	9.2
Shaking	192.1	66.6	82.6	80.0	38.7	59.2	80.7	26.3	28.4	85.7	17.2	29.2	37.1	9.5	79.8	9.2	11.1	102.6	22.1	21.0	8.5

TABLE II: Average overlap rate(%). The best three results are shown in red, blue, and green fonts.

Sequence	FragT	BSBT	LOT	CT	SMS	KMS	CPF	DFT	ORIA	IVT	CSK	CXT	TLD	VTD	$\ell_1$apg	MTT	SCM	LSST	ASLA	MRT	TMRT
Car4	22	21	4	28	5	27	19	25	23	92	47	31	64	73	70	53	89	92	89	89	92
Car11	9	43	42	23	2	36	8	38	38	81	76	57	38	43	83	58	79	84	81	82	85
David	19	39	27	56	24	38	14	30	43	72	41	65	60	53	63	53	75	75	79	79	82
Occ1	90	77	41	74	58	72	53	69	64	85	79	63	65	77	87	79	93	89	83	82	92
Occ2	60	64	46	68	8	46	42	77	72	59	78	74	49	59	70	72	82	86	82	81	83
Sylvester	58	57	57	68	7	47	56	38	65	52	63	60	67	62	40	65	69	28	59	60	65
Singer1	34	21	19	34	54	32	45	36	65	66	36	49	41	79	83	32	85	80	81	83	87
Skating1	13	16	26	9	4	31	19	14	22	34	50	14	19	53	10	10	47	34	42	45	57
Woman	15	19	9	13	6	57	7	76	15	19	19	20	13	14	16	16	66	78	78	79	82
Subway	46	17	56	57	18	15	12	73	17	17	19	18	18	16	16	7	72	15	19	21	67
Walking2	27	26	34	27	31	27	32	40	45	79	46	37	31	33	76	79	82	34	37	36	81
Caviar	19	14	25	33	56	42	32	14	19	21	19	19	21	19	13	14	87	85	84	85	89
Freeman4	14	15	16	0	2	3	5	17	20	15	13	17	22	16	34	22	26	13	13	36	62
Tiger	26	22	14	41	39	39	39	53	13	10	26	32	38	12	31	26	16	27	29	35	60
Deer	8	40	21	60	10	41	12	25	4	22	73	70	41	58	45	60	46	58	62	63	74
Motorbike	13	30	58	14	11	49	11	30	65	73	71	23	20	70	73	74	67	26	72	71	74
Biker	21	26	44	46	46	25	36	25	38	64	50	42	23	63	34	42	35	36	55	56	67
Football	57	29	65	46	2	8	64	65	51	55	55	54	56	81	68	71	69	69	57	51	70
Jumping	14	15	58	7	9	10	10	11	9	28	5	52	69	8	59	30	73	65	24	56	71
Board	67	10	19	45	52	65	55	31	13	33	67	17	20	47	16	19	63	40	74	73	82
Surfer	39	30	49	52	50	49	35	35	21	23	25	49	45	38	18	40	25	19	37	38	60
Shaking	8	11	13	20	25	23	12	63	44	30	57	12	39	70	28	69	68	42	46	50	75

LSST and TMRT methods can track the object quite well.

In the Freeman4 sequence, the target person move around a classroom while the other students are waving papers ahead. Initially, SCM, TLD, L1APG, VTD and our tracker succeed to recover from the heavy occlusion. After frame 204, only our TMRT tracker tracks the man successfully.

IV. Conclusion

In summary, based on the framework of the sparse based trackers [16–18], we developed a real time temporal smooth visual tracker with improved tracking performance. The proposed tracker utilizes information among consecutive frames to improve the tracking accuracy and employs marginal regression to speedup the tracking speed significantly. Extensive experiments on real-world video sequences have been done to validate the high computational efficiency and better accuracy of the TMRT method.

References

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm computing surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.

[2] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 2411–2418.

[3] B. Wang, G. Wang, K. L. Chan, and L. Wang, "Tracklet association with online target-specific metric learning," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1234–1241.

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

[5] D. Wang, H. Lu, and M.-H. Yang, "Online object tracking with sparse prototypes," *Image Processing, IEEE Transactions on*, vol. 22, no. 1, pp. 314–325, 2013.

[6] Y. Bai and M. Tang, "Object tracking via robust multitask sparse representation," *Signal Processing Letters, IEEE*, vol. 21, no. 8, pp. 909–913, 2014.

[7] 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.

[8] B. Zhuang, H. Lu, Z. Xiao, and D. Wang, "Visual tracking via discriminative sparse similarity map," 2014.

[9] N. Wang, J. Wang, and D.-Y. Yeung, "Online robust non-negative dictionary learning for visual tracking," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 657–664.

[10] J. Lu, Y.-P. Tan, G. Wang, and G. Yang, "Image-to-set face recognition using locality repulsion projections and sparse reconstruction-based similarity measure," *IEEE transactions on circuits and systems for video technology*, vol. 23, no. 6, pp. 1070–1080, 2013.

[11] J. Lu, G. Wang, and P. Moulin, "Human identity and gender recognition from gait sequences with arbitrary walking directions," *Information Forensics and Security, IEEE Transactions on*, vol. 9, no. 1, pp. 51–61, 2014.

[12] F. Zeng, X. Liu, Z. Huang, and Y. Ji, "Kernel based multiple cue adaptive appearance model for robust real-time visual tracking," *Signal Processing Letters, IEEE*, vol. 20, no. 11, pp. 1094–1097, 2013.

[13] D. Wang, H. Lu, and C. Bo, "Online visual tracking via two view sparse representation," *Signal Processing Letters, IEEE*, vol. 21, no. 9, pp. 1031–1034, 2014.

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

[15] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via structured multi-task sparse learning," *International journal of computer vision*, vol. 101, no. 2, pp. 367–383, 2013.

[16] X. Mei and H. Ling, "Robust visual tracking using ℓ_1 minimization," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1436–1443.

[17] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient l1 tracker with occlusion detection," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1257–1264.

[18] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1822–1829.

[19] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1830–1837.

[20] J. Xing, J. Gao, B. Li, W. Hu, and S. Yan, "Robust object tracking with online multi-lifespan dictionary learning," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 665–672.

[21] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[22] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.

[23] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani *et al.*, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[24] K. Balasubramanian, K. Yu, and G. Lebanon, "Smooth sparse coding via marginal regression for learning sparse representations," in *ICML (3)*, 2013, pp. 289–297.

[25] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution," *Communications on pure and applied mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[26] J. Fan and J. Lv, "Sure independence screening for ultrahigh dimensional feature space," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 5, pp. 849–911, 2008.

[27] C. R. Genovese, J. Jin, L. Wasserman, and Z. Yao, "A comparison of the lasso and marginal regression," *The Journal of Machine Learning Research*, vol. 98888, no. 1, pp. 2107–2143, 2012.

[28] 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, pp. 125–141, 2008.

[29] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 798–805.

[30] S. Stalder, H. Grabner, and L. Van Gool, "Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1409–1416.

[31] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1940–1947.

[32] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 864–877.

[33] R. T. Collins, "Mean-shift blob tracking through scale space," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2. IEEE, 2003, pp. II–234.

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

[35] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Computer vision–ECCV 2002*. Springer, 2002, pp. 661–675.

[36] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1910–1917.

[37] Y. Wu, B. Shen, and H. Ling, "Online robust image alignment via iterative convex optimization," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1808–1814.

[38] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 702–715.

[39] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1177–1184.

[40] Z. Kalal, J. Matas, and K. Mikolajczyk, "Pn learning: Bootstrapping binary classifiers by structural constraints," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 49–56.

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

[42] D. Wang, H. Lu, and M.-H. Yang, "Least soft-threshold squares tracking," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 2371–2378.

[43] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.