ROBUST VISUAL TRACKING VIA INVERSE NONNEGATIVE MATRIX FACTORIZATION

Fanghui Liu[1], *Tao Zhou*[1], *Keren Fu*[1,2], *Irene Y.H. Gu*[2] *and Jie Yang*[*1]

[1]Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, China
[2]Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden

ABSTRACT

The establishment of robust target appearance model over time is an overriding concern in visual tracking. In this paper, we propose an inverse nonnegative matrix factorization (NMF) method for robust appearance modeling. Rather than using a linear combination of nonnegative basis vectors for each target image patch in conventional NMF, the proposed method is a reverse thought to conventional NMF tracker. It utilizes both the foreground and background information, and imposes a local coordinate constraint, where the basis matrix is sparse matrix from the linear combination of candidates with corresponding nonnegative coefficient vectors. Inverse NMF is used as a feature encoder, where the resulting coefficient vectors are fed into a SVM classifier for separating the target from the background. The proposed method is tested on several videos and compared with seven state-of-the-art methods. Our results have provided further support to the effectiveness and robustness of the proposed method.

Index Terms— inverse NMF, local coordinate constraint, incremental NMF, visual tracking

1. INTRODUCTION

Visual tracking has been consolidated its important research status in computer vision with wide applications ranging from video surveillance to vehicle navigation [1]. One essential aspect in visual tracking is to model the appearance of objects. Such modeling methods can be either generative [2–4] or discriminative [5,6] with pros and cons. Generative methods focus on searching the most similar candidate to the target with minimizing reconstruction error; Discriminative methods cast the tracking problem as a binary classification, separating the target from the background.

Nonnegative Matrix Factorization (NMF) has recently been applied to visual tracking with variety works including Orthogonal Projective NMF Tracker [7], Constraint Online NMF Tracker [8] and Constrained Incremental NMF Tracker [9]. In NMF, a nonnegative data matrix $\mathbf{X}$ is decomposed into two non-negative matrices $\mathbf{U}$ and $\mathbf{V}$ ($\mathbf{X} \approx \mathbf{UV}$), where $\mathbf{U}$ is the basis matrix and the columns of $\mathbf{V}$ are coefficients vectors. In these methods, as generative trackers, NMF with different constraints (e.g., sparsity constraint, graph-based regularization) are adopted into appearance modelling. Different form previous work, NMF in [10] serves as an approach of feature extraction. After solving nonnegative coefficient vectors $\mathbf{v}_i$ of different corresponding candidates based on $\mathbf{U}$, a Naive Bayes classifier is trained to distinguish between the target and the background.

It might be tempting to agree that these methods have shown good performance for a range of scenarios. However, further exploiting discriminative information could improve the robustness of tracking. For example, in existing generative NMF trackers, background information are not taken into consideration, leading to lack of discriminative ability in these NMF's variants. In [10], only basis matrix $\mathbf{U}$ is adopted to represent the target, whereas the corresponding encoding vectors $\mathbf{v}_i$ are ignored.

Motivated by the above issues, this paper proposes a novel tracking method, the inverse NMF tracker, that is a reverse thought to conventional NMF tracker. The main novelties of the proposed method include: (a) an inverse NMF representation formulation is proposed to represent the basis matrix by disparate candidates with corresponding coefficient vectors, which combines both the foreground and background information; (b) a local coordinate constraint is imposed on encoding vectors for local similarity and sparsity; (c) incremental learning is introduced to the proposed tracker for online updating appearance models.

2. THE BIG PICTURE OF PROPOSED METHOD

As shown in the block diagram of Fig.1, the proposed method can be briefly described as follows. First, a simple tracker (e.g. IVT [2]) is used as the initialization process on the first m frames to collect target patches in each frame. This forms the positive template set $\mathbf{T}_{pos} = [\mathbf{T}_1^p, \mathbf{T}_2^p, \cdots, \mathbf{T}_N^p]$ (or called the initial data matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$), and the negative template set, $\mathbf{T}_{neg} = [\mathbf{T}_1^n, \mathbf{T}_2^n, \cdots, \mathbf{T}_r^n] \in \mathbb{R}^{M \times r}$ from the background. The positive template set is then decomposed into the basis matrix $\mathbf{U} \in \mathbb{R}^{M \times K}$ and coefficient matrix $\mathbf{V} \in \mathbb{R}^{K \times N}$ by using Graph-based NMF [11]. After that (frames $> m$), new candidate object patches are sampled using a particle filter, forming $\mathbf{Y}_{1:S} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_S\} \in \mathbb{R}^{M \times S}$. The proposed inverse NMF is then applied for estimating the coefficient vectors $\mathbf{C}_{pos}$, $\mathbf{C}_{neg}$ and $\mathbf{C}$ from the positive samples $\mathbf{T}_{pos}$, negative samples $\mathbf{T}_{neg}$ and candidates $\mathbf{Y}$, and fed into a SVM

*Corresponding author: jieyang@sjtu.edu.cn (Jie Yang). This research is partly supported by NSFC, China (No: 61273258) and 863 Plan, China (No. 2015AA042308).

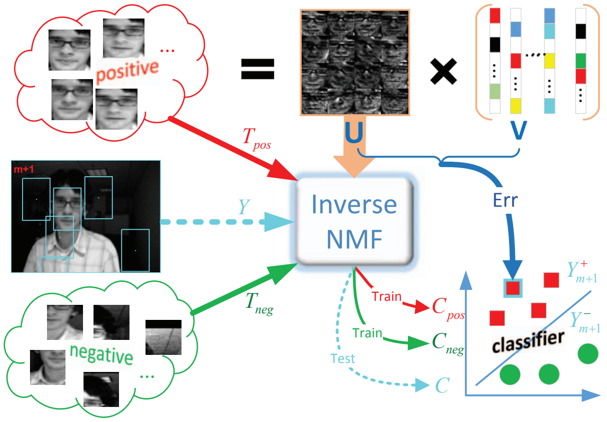

Fig. 1. Illustration of the proposed inverse NMF framework

classifier for training, and subsequently employed for assigning the encoding vector $\mathbf{c}^{(i)}$ to the target or the background.

3. PROPOSED INVERSE NMF TRACKER

3.1. Review: Conventional NMF and Its Variants

For the sake of mathematical convenience and easy to use in the subsequent description, methods and formula in some conventional NMFs are briefly summarized. It provides a justification for NMF's widespread application such as face recognition [12], data clustering [13]. Basis vectors in $\mathbf{U}$ represents latent semantic information of original data in a subspace, each basis vector of which reflects the centroid of a cluster.

Based on NMF, to preserve the similarity between the coefficient vectors and the data points, a Laplacian regularization term is introduced into NMF [11]:

$$\mathcal{O} = \|\mathbf{X} - \mathbf{UV}\|_F^2 + \lambda Tr(\mathbf{VLV}^T) \tag{1}$$

where λ is graph-based regularization parameter. The graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$ and $\mathbf{W}$ is the weight matrix,

$$W_{ij} = \begin{cases} \mathrm{e}^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}} & \text{if } \mathbf{x}_\mathrm{i} \in \mathcal{N}_\mathrm{k}(\mathbf{x}_\mathrm{j}), \text{ or } \mathbf{x}_\mathrm{j} \in \mathcal{N}_\mathrm{k}(\mathbf{x}_\mathrm{i}) \\ 0 & \text{otherwise} \end{cases}$$

An alternative way, that simultaneously takes into account the similarity and sparsity, is to use a local coordinate coding constraint [14]. The columns of the basis matrix $\mathbf{U}$ can be considered as a set of anchor points, and each data point in the original space can be approximated by a linear combination of only a few anchor points [15].

$$\mathcal{Q} = \sum_{i=1}^{N} (\mu \sum_{k=1}^{K} |v_{ki}| \cdot \|\mathbf{u}_k - \mathbf{x}_i\|^2) = \mu \sum_{i=1}^{N} \|(\mathbf{x}_i \mathbf{1}^T - \mathbf{U})\Lambda_i^{1/2}\|^2 \tag{2}$$

where μ is the regularization parameter, and $\mathbf{1} \in \mathbb{R}^K$ denotes the column vector whose entries are all ones and $\Lambda_i = diag(\mathbf{v}_i) \in \mathbb{R}^{K \times K}$. This term in (2) is the local coordinate constraint, that imposes penalty if $\mathbf{u}_k$ is far away from $\mathbf{x}_i$ when the new coordinate v_{ki} is large.

3.2. Estimate Coefficient Matrices by the Inverse NMFs

The implicit rationale behind inverse NMF is also based on a perspective of clustering representation in computer vision applications. As a reverse thought to conventional NMF, the basis matrix $\mathbf{U}$ is spanned by candidates $\mathbf{Y}$ ($\mathbf{U} \approx \mathbf{YC}$). Each row $\mathbf{c}^{(i)}$ in $\mathbf{C}$ corresponds to the responses of one candidate on basis matrix $\mathbf{U}$, which can be regarded as discriminative feature for classification in visual tracking. Compared with the reverse sparsity theory in [16], the coefficient vector $\mathbf{c}^{(i)}$ is natural sparse due to distinct meanings in NMF. If $\mathbf{Y}$ contains a set of good candidates (i.e., similar to the target), these good candidates will spread among basis vectors. A few nonzero coefficients in $\mathbf{c}^{(i)}$ are needed, as basis vectors can be easily represented their neighbouring good candidates. For bad candidates, they seems to be incoherent in a subspace spanned by basis vectors. And there is not definite link between the background and the target representation $\mathbf{U}$. If $\mathbf{Y}$ contains a set of bad candidates (from the background), it is difficult for these bad candidates to represent basis vectors accurately and sparsely. These corresponding coefficient vectors do not hold the sparsity property as that in the positive sample case.

By exploiting this difference between the candidates in the target and the background, the basis matrix $\mathbf{U}$ can be mapped into associated coefficient vectors. Since only coefficient vectors from good candidates are associated with physical meanings, coefficient vectors are used as discriminative features to separate the target from the background.

Incorporating the local coordinate constraint into our inverse NMF method to preserve the similarity of coefficient vectors for the similar candidate features and sparsity in these vectors simultaneously. We estimate the coefficient matrix $\mathbf{C}$ by employing the following objective function using the constrained optimization,

$$\min_{\mathbf{C}} \|\mathbf{U} - \mathbf{YC}\|_F^2 + \mu \sum_{k=1}^{K} \|(\mathbf{u}_k \mathbf{1}^T - \mathbf{Y})\Omega_k^{1/2}\|^2 \tag{3}$$
$$s.t. \;\; \mathbf{C} \geq 0$$

where $\mathbf{1} \in \mathbb{R}^S$ and $\Omega_k = diag(\mathbf{c}_k) \in \mathbb{R}^{S \times K}$. We estimate the positive coefficient vector $\mathbf{C}_{pos}$ using the target patches $\mathbf{T}_{pos}$ ($\mathbf{X}$), by the following formula:

$$\min_{\mathbf{C}_{pos}} \|\mathbf{U} - \mathbf{XC}_{pos}\|_F^2 + \mu \sum_{k=1}^{K} \|(\mathbf{u}_k \mathbf{1}^T - \mathbf{X})\Gamma_k^{1/2}\|^2 \tag{4}$$
$$s.t. \;\; \mathbf{C}_{pos} \geq 0$$

where $\mathbf{1} \in \mathbb{R}^N$ and $\Gamma_i = diag(\mathbf{c}_{pos})_i \in \mathbb{R}^{N \times K}$. We can easily derive the formula for estimating $\mathbf{C}_{neg}$ from the negative templates $\mathbf{T}_{neg}$ in a similar fashion.

It is worth noting that the objective functions in (3) and (4) for estimating $\mathbf{C}_{pos}$, $\mathbf{C}_{neg}$ and $\mathbf{C}$ are differentiable convex functions, and the nonnegative constraints are non-smooth convex functions. Hence, their solutions can be obtained by minimizing the cost functions with respect to $\mathbf{C}_{pos}$, $\mathbf{C}_{neg}$ and $\mathbf{C}$, by using the accelerated proximal gradient (APG) algorithm in [4].

3.3. Identify Candidates

The estimated $\mathbf{C}_{pos}^T$ and $\mathbf{C}_{neg}^T$ from inverse NMFs are used as the features for the positive and negative samples, for training a SVM classifier. Once $\mathbf{C}$ is obtained, the SVM classifier is employed to assign the encoding vector $\mathbf{c}^{(i)}$ to the target or the background. The corresponding candidates $\mathbf{Y}_{1:S}$ is divided into positive candidates $\mathbf{Y}^+$ and negative candidates $\mathbf{Y}^-$.

To make our algorithm more robust, a coarse-to-fine searching scheme for the optimal candidate is proposed. After obtaining $\mathbf{Y}^+$ and $\mathbf{Y}^-$, we do not exactly choose the positive candidate with the highest confidence value as our tracking result [1]. The observation likelihood can be measured by the reconstruction error of positive candidates $\mathbf{Y}^+$ as shown in Fig.1 (noting that the time index is omitted for simplicity):

$$p(\mathbf{y}_i^+|\mathbf{x}_i) = \operatorname*{argmax}_j \exp(-\|\mathbf{y}_i^+ - \mathbf{U}\mathbf{v}_j\|_2^2) \quad \forall j \qquad (5)$$

where $\mathbf{y}_i^+$ represents the i-th positive candidate from $\mathbf{Y}^+$, and $\mathbf{v}_j$ denotes the j-th column of coefficient matrix $\mathbf{V}$. This searching scheme incorporates the merit of generative methods into the classification problem. The optimal state $\mathbf{x}^*$ from the positive samples $\mathbf{Y}^+$ with the minimal reconstruct error is chosen as the tracking result.

3.4. Incremental Learning for Online Updating

Incremental learning is applied for maintaining timely target and background appearance models. For the negative template set, the model is updated in each short time interval (e.g., 5 frames in our tests) as the tradeoff between the computation and the model fitness. With the new optimal candidate added into the positive template set $\mathbf{X}$, our appearance model need to be update promptly. It is impossible to recalculate $\mathbf{U}$ and $\mathbf{V}$ totally just because of time-consuming. Although incremental learning is studied in both NMF [17] and GNMF [18], we adopt the incremental learning in a similar spirit to that in GNMF, however, fits to updating $\mathbf{U}$ and $\mathbf{V}$ in the proposed inverse NMF. This can be described as follows. Let $\mathbf{X}^{t+1}$, $\mathbf{U}^{t+1}$, $\mathbf{V}^{t+1}$, $\mathbf{E}^{t+1}$, $\mathbf{W}^{t+1}$, and $\mathbf{D}^{t+1}$ be the corresponding matrices when the $(t+1)$-th sample $\mathbf{x}$ arrives. Noting that $\mathbf{X}^{t+1} = [\mathbf{X}^t, \mathbf{x}] \in \mathbb{R}^{M\times(t+1)}$, $\mathbf{V}^{t+1} = [\mathbf{V}^t, \mathbf{v}] \in \mathbb{R}^{K\times(t+1)}$, the relation $[\mathbf{X}^t, \mathbf{x}] \approx \mathbf{U}^{t+1}\ [\mathbf{V}^t, \mathbf{v}]$ holds. The incremental learning on each element u_{ik} in $\mathbf{U}$ and v_i in $\mathbf{v}$ may then be written by the following updating equations:

$$\begin{aligned} u_{ik} &\leftarrow u_{ik}\frac{[\mathbf{X}^{t+1}(\mathbf{V}^{t+1})^T]_{ik}}{[\mathbf{U}^{t+1}\mathbf{V}^{t+1}(\mathbf{V}^{t+1})^T]_{ik}} \\ v_i &\leftarrow v_i\frac{[(\mathbf{U}^{t+1})^T\mathbf{x} + \lambda\mathbf{V}^t(\mathbf{W}^{t+1})_{:,t+1} + \lambda\mathbf{v}w_{end}]_i}{[(\mathbf{U}^{t+1})^T\mathbf{U}^{t+1}\mathbf{v} + \lambda\mathbf{V}^t(\mathbf{D}^{t+1})_{:,t+1} + \lambda\mathbf{v}d_{end}]_i} \end{aligned} \qquad (6)$$

where $(\mathbf{W}^{t+1})_{:,t+1}$ is the $(t+1)$-th column of the Laplacian matrix $\mathbf{W}$, and $w_{end} = (\mathbf{W}^{t+1})_{t+1,t+1}$ is the element from the last row and last column of $\mathbf{W}$. Similar definition holds for d_{end} in the matrix $(\mathbf{D}^{t+1})_{:,t+1}$.

The proposed inverse NMF tracker is a combination of using the inverse NMF method for finding sparse encoding vectors and using the particle filter for finding the best target candidate. The flowchart of the inverse NMF tracking algorithm is summarized in **Algorithm 1**.

```
Algorithm 1: Algorithm for Inverse NMF Tracker
Initialization: Extract templates T in the first $m$ frame.
Construct the weight matrix W by using (3.1) and the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$.
Obtain X, U, V by (1).
for $t = m + 1$ to the end of the sequence do
    $S$ particles are sampled;
    Inverse NMF: obtain encoding vector: $\mathbf{C}_{pos}$, $\mathbf{C}_{neg}$ and C by the APG approach;
    Train a SVM classifier by $\mathbf{C}_{pos}^T$, $\mathbf{C}_{neg}^T$, and then classifies encoding vector $\mathbf{c}^{(i)}$ of C;
    for each positive particle $\mathbf{y}_t^+$ do
        Compute their likelihood by (5);
    end
    Choose $\mathbf{x}_t^*$ with the minimal reconstruct error;
    Update: for each $5$ frames do
        Update positive and negative template sets;
        Recalculate W and D in (3.1);
        Update U and v by (6);
    end
end
```

4. EXPERIMENTS

Setup: The proposed tracker was implemented in MATLAB on a PC with Intel Xeon E5506 CPU (2.13 GHz) and 24 GB memory. The following parameters were used for our tests: each observation (i.e. patch of image) was normalized to 32×32 pixels; the graph-regularized parameter was set to $\lambda = 1$; kNN was fixed to 10 nearest neighbors; the spread $\sigma = 2$ was used in the Gaussian Kernel; the number of initial positive templates and the negative templates were $N = 140$ and $r = 280$ respectively from the first 5 frames; the number of basis vectors was $K = 16$; the local coordinate regularization parameters was $\mu = 0.5$; the iteration number was set to 5, and the Lipschitz constant was 1/0.00018 in the APG.

Methods comparison: The proposed method is also compared with seven state-of-the-art methods, including: DSSM [16], ASLA [19], L1 Tracker [3], L1-APG [4], MTT [20], IVT [2], and MIL [21].

Results: Fig.2 shows screen shots of tracking results from different trackers. Tab.1 shows the performance of these methods based on the center location error (CLE), where a small CLE value indicates more accurate hence better tracking. Tab.2 shows the performance of different methods based on the overlap rate between the tracked bounding box and the

[1] In our experiments, statistical results show that the number of $\mathbf{Y}^+$ accounts for about 10% of the whole $\mathbf{Y}$.

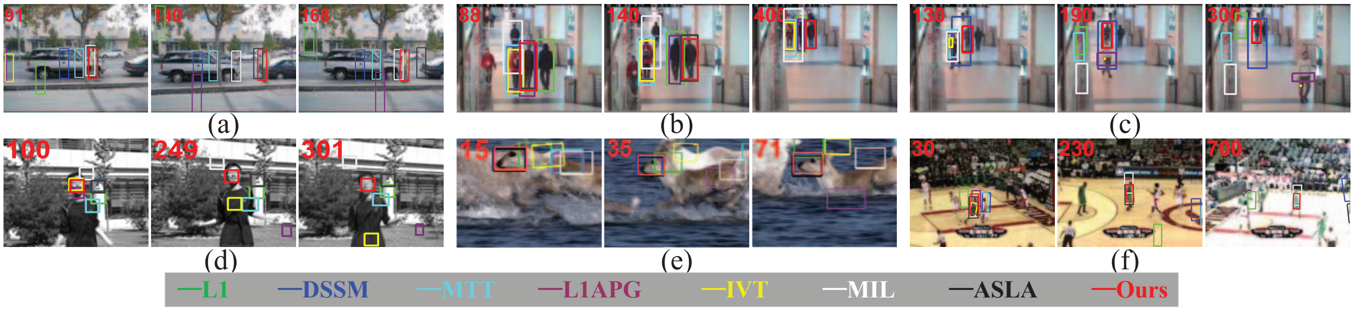

Fig. 2. Representative frames of some sampled tracking results. And subfigures from top to bottom, left to right: (a) - (f), from video David3, Caviar1, Caviar3, Jumping, Deer and Basketball.

Table 1. Performance in terms of ”center location error” (CLE) in pixels. Red and blue colors indicate the best and 2nd best performance, respectively.

Sequence	DSSM	ASLA	L1	MTT	IVT	MIL	L1APG	Ours
Human7	80.4	2.9	103.7	17.0	54.1	21.8	27.8	7.3
David3	99.2	87.4	100.4	363.3	100.2	38.4	204.4	5.0
Jumping	115.8	45.4	51.3	58.7	27.9	9.7	97.5	5.2
Deer	10.0	10.7	97.9	15.2	123.8	225.8	197.9	8.1
Caviar1	22.1	1.5	34.6	55.2	98.5	48.5	95.0	2.5
Caviar3	61.5	2.2	65.9	64.8	66.2	57.8	26.6	5.5
Carscale	17.8	48.8	66.8	83.7	11.7	27.3	81.2	14.9
Faceocc1	23.5	7.7	6.5	8.9	17.3	11.7	32.3	6.3
Basketball	242.1	21.0	126.1	117.2	310.4	139.7	97.7	11.4
Car4	55.0	3.5	4.1	223.0	96.1	2.6	60.1	4.4
avg.	72.8	23.1	65.7	107.9	108.2	63.6	61.9	7.3

Table 2. Performance in terms of ”overlap rate” e. Red and blue colors indicate the best and 2nd best performance, respectively.

Sequence	DSSM	ASLA	L1	MTT	IVT	MIL	L1APG	Ours
Human7	0.31	0.81	0.12	0.48	0.11	0.29	0.27	0.71
David3	0.33	0.46	0.35	0.09	0.31	0.41	0.14	0.77
Jumping	0.24	0.10	0.30	0.24	0.49	0.53	0.12	0.67
Deer	0.61	0.60	0.07	0.55	0.04	0.04	0.05	0.61
Caviar1	0.42	0.89	0.28	0.28	0.27	0.25	0.28	0.83
Caviar3	0.14	0.84	0.20	0.14	0.13	0.11	0.19	0.69
Carscale	0.75	0.45	0.36	0.49	0.62	0.42	0.55	0.65
Faceocc1	0.69	0.87	0.87	0.84	0.72	0.82	0.59	0.88
Basketball	0.038	0.56	0.01	0.03	0.02	0.03	0.25	0.63
Car4	0.52	0.91	0.84	0.15	0.45	0.91	0.34	0.88
avg.	0.40	0.65	0.33	0.26	0.35	0.37	0.32	0.74
fps.	0.86	0.66	0.23	1.02	20.41	18.86	4.41	1.23

ground truth box from these methods on several videos. The overlap rate is defined as $e = \frac{area(R_T \cap R_G)}{area(R_T \cup R_G)}$, where R_T and R_G are the area of tracked and ground truth box, respectively.

DavidOutdoor: The challenging issues in this video shown in Fig.2(a) are mainly due to *heavy occlusions* and *pose changes*. L1, L1-APG, IVT, MTT and DSSM completely fail at frames #36, #34, #57, #64, #75 and #140. When David passes through the tree, MIL suffers from severe drifts. When pose change occurs, the proposed tracker performs well without drifting whereas ASLA does not.

Caviar1 and Caviar3: The two videos in Fig.2(b)(c) are typical ones with *heavy occlusion*. Without heavy occlusions, all methods achieve favorable performance. However, the existing methods either fails to track or tracks with degraded accuracy after the target is heavily occluded. From the tracking results in these videos, ASLA and our proposed tracker are shown to be more robust against severe occlusions. The remaining trackers, L1, L1APG, IVT, MTT and DSSM, show difficult in capturing appearance changes after occlusions, where the appearance becomes much dissimilar to their initial one.

Jumping and Deer: Tracking in these 2 videos, as shown in Fig.2(d),(e), are affected by *fast motion* and *motion blur*. It is difficult to accurately predict the location of the target when abrupt motion occurs, and the appearance changes due to motion blur posed some challenges for accurately locating the target. In the *Jumping* video, before #249 frame, most trackers have poor tracking accuracy except IVT and the proposed tracker. After #249 frame, only the proposed tracker is able to track the target object. In the *Deer* video, the proposed tracker and DSSM shows promising results as compared with the proposed tracker.

Basketball: The video, shown in Fig.2(f), contains drastic *heavy occlusions*, *pose variation* and *illumination variation*. ASLA and our proposed method accurately track the basketball player when the player is occluded by others, and MIL loses tracking accuracy to some extent when the player suffers pose variation. The other methods are not adapt to the appearance changes. When comes to illumination variation, ASLA cannot locate the target to obtain accurate appearance model.

In summary, our test results on these videos have shown that ASLA and the proposed tracker are effective and robust to heavy occlusions. Only our method adapts the scale variation, pose changes and illumination variation of the target.

5. CONCLUSION

This paper proposes an inverse NMF method for visual tracking. It combines the merit of generative tracking methods with the discriminative methods. The proposed inverse NMF method not only leverages the minimal reconstruction error to search the optimal candidate but also takes the background information into consideration. Using separately estimated coefficient vectors that are served as encoding features from the foreground and background in inverse NMF, the proposed tracker shows enhanced discriminant ability during the tracking. Quantitative and qualitative comparisons with seven state-of-the-art trackers on ten videos have demonstrated the effectiveness and robustness of the proposed tracker.

6. REFERENCES

[1] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Object Tracking Benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 1–1, 2015.

[2] David a. Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang, "Incremental Learning for Robust Visual Tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[3] Xue Mei and Haibin 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.

[4] Chenglong Bao, Yi Wu, Haibin Ling, and Hui Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1830–1837.

[5] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie, "Robust Object Tracking with Online Multiple Instance Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.

[6] Sam Hare, Amir Saffari, and Philip H S Torr, "Struck: Structured output tracking with kernels," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 263–270.

[7] Dong Wang and Huchuan Lu, "On-line learning parts-based representation via incremental orthogonal projective non-negative matrix factorization," *Signal Processing*, vol. 93, no. 6, pp. 1608–1623, 2013.

[8] Yi Wu, Bin Shen, and Haibin Ling, "Visual tracking via online nonnegative matrix factorization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 3, pp. 374–383, 2014.

[9] Huanlong Zhang, Shiqing Hu, Xiaoyu Zhang, and Lingkun Luo, "Visual Tracking via Constrained Incremental Non-negative Matrix Factorization," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1350–1353, 2015.

[10] Cheng Qian, Yanbin Zhuang, and Zezhong Xu, "Visual tracking with structural appearance model based on extended incremental non-negative matrix factorization," *Neurocomputing*, vol. 136, pp. 327–336, 2014.

[11] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.

[12] David Guillamet and Jordi Vitria, "Non-negative matrix factorization for face recognition," in *Topics in Artificial Intelligence*, pp. 336–344. Springer, 2002.

[13] Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han, "Multi-view clustering via joint nonnegative matrix factorization," in *Proc. of SDM*. SIAM, 2013, vol. 13, pp. 252–260.

[14] Kai Yu, Tong Zhang, and Yihong Gong, "Nonlinear learning using local coordinate coding," in *Advances in neural information processing systems*, 2009, pp. 2223–2231.

[15] Yan Chen, Jiemi Zhang, Deng Cai, Wei Liu, and Xiaofei He, "Nonnegative local coordinate factorization for image representation," *IEEE Transactions on Image Processing*, vol. 22, no. 3, pp. 969–979, 2013.

[16] Bohan Zhuang, Huchuan Lu, Ziyang Xiao, and Dong Wang, "Visual tracking via discriminative sparse similarity map," *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1872–1881, 2014.

[17] Serhat S. Bucak and Bilge Gunsel, "Incremental subspace learning via non-negative matrix factorization," *Pattern Recognition*, vol. 42, no. 5, pp. 788–797, 2009.

[18] Zhe-Zhou Yu, Yu-Hao Liu, Bin Li, Shu-Chao Pang, and Cheng-Cheng Jia, "Incremental Graph Regulated Nonnegative Matrix Factorization for Face Recognition," *Journal of Applied Mathematics*, vol. 2014, no. 11, pp. 1–10, 2014.

[19] Xu Jia, Huchuan Lu, and Ming Hsuan Yang, "Visual tracking via adaptive structural local sparse appearance model," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1822–1829, 2012.

[20] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja, "Robust visual tracking via structured multi-task sparse learning," *International Journal of Computer Vision*, vol. 101, no. 2, pp. 367–383, 2013.

[21] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie, "Robust Object Tracking with Online Multiple Instance Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.