

# Visual Tracking via Online Non-negative Matrix Factorization

Yi Wu\*, *Member, IEEE*, Bin Shen\*, and Haibin Ling, *Member, IEEE*

**Abstract**—In visual tracking, holistic and part-based representations are both popular choices to model target appearance. The former is known for great efficiency and convenience while the latter for robustness against local appearance or shape variations. Based on non-negative matrix factorization (NMF), we propose a novel visual tracker that takes advantage of both groups. The idea is to model the target appearance by a non-negative combination of non-negative components learned from examples observed in previous frames. To adjust NMF to the tracking context, we include sparsity and smoothness constraints in addition to the non-negativity one. Furthermore, an online iterative learning algorithm, together with a proof of convergence, is proposed for efficient model updating. Putting these ingredients together with a particle filter framework, the proposed tracker, *Constrained Online Non-negative Matrix Factorization* (CONMF), achieves robustness to challenging appearance variations and non-trivial deformations while runs in real time. We evaluate the proposed tracker on various benchmark sequences containing targets undergoing large variations in scale, pose or illumination. The robustness and efficiency of CONMF is validated in comparison with several state-of-the-art trackers.

**Index Terms**—Visual tracking, NMF, Non-negative Matrix Factorization, particle filter.



## 1 INTRODUCTION

Visual tracking has been attracting research efforts in computer vision community for decades. It plays an important role in various applications such as security surveillance, human computer interaction, media management, etc [45].

A key challenge in visual tracking is to model the target appearance over time, since it often undergoes non-trivial variations (e.g., illumination and pose changes) and non-linear corruptions (e.g., occlusions and motion blur [44]). A huge amount of previous work has been devoted to visual tracking with many different appearance models [25], [42]. Appearance features such as color, edge, texture are often used as object representation [33], [11]. While extremely efficient, color distribution-based models discard important geometric information between parts of the target. This issue makes these trackers vulnerable to variations from the environment (e.g., occlusion) or the target itself (e.g., pose change). Alternatively, holistic or part-based methods simultaneously encode intensity information and geometric information, usually with a more complex model and more expensive computational cost. A typical and popular holistic method is template-based representation [13], [43], which treats a target as a vector in a space spanned by a template set. A natural extension is

to use low-dimensional subspace representations to enhance efficiency and robustness (e.g., [34]). Part-based methods (e.g., [20], [37], [27], [1]), on the other hand, emphasize on the semantic structures of target appearance and address naturally local appearance changes and non-linear deformation.

To handle temporal variation, dynamical update or learning schemes have been used in many tracking systems. For example, template update is widely used in template-based trackers (e.g., [28], [4], [10]). Recently, online learning approaches have been introduced for visual tracking, which dynamically update trackers with newly available information (e.g., [2], [24]). Manifold-based representations are used (e.g., [21]) for modeling pose and view changes.

In this paper, we propose a novel target representation, based on the *Non-negative Matrix Factorization* (NMF) [22], to implicitly combine holistic and part-based methods. The idea is to model target appearances as non-negative linear combinations of a set of non-negative basis that implicitly captures structure information. To encode the characteristics in the tracking process, we introduce sparsity and smoothness constraints, which reflect the underlying assumption that target appearances across frames are drawn from the same manifold. As a result, our representation inherits the implicit part decomposition and other merits of nonnegativity constraint from NMF, the conciseness from the sparse representation, and the rich description power from underlying manifold representation. As shown in our experiments, our approach is robust to environmental variation (e.g., illumination change or noises), target deformation (e.g., shape or pose change), and corruptions (e.g., partial or full occlusion). The proposed tracker achieves the state-of-art performances while runs very fast.

To apply the constrained NMF for visual tracking, we propose an online algorithm, named *Constrained Online Non-negative Matrix Factorization* (CONMF), to capture the dy-

\*These authors contributed equally to this work. This work was supported in part by NSFC (Grants 61005027, 61273259, and 61272223) and NSF of Jiangsu Province (Grants BK2012045, and BK2011825). Ling is supported in part by NSF (Grant IIS-1218156).

- Yi Wu is with the School of Information and Control Engineering, Nanjing University of Information Science and Technology, Nanjing, China, 210044.
- Bin Shen is with the Department of Computer Science, Purdue University, West Lafayette, IN 47907 USA.
- Haibin Ling is with the Department of Computer and Information Science, Temple University, Philadelphia, PA 19122 USA.
- Copyright (c) 2013 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to [pubs-permissions@ieee.org](mailto:pubs-permissions@ieee.org).

dynamic information. CONMF incrementally updates the current appearance model, represented by the basis matrix  $U$ , using information in the new frame. An illustration of the idea is shown in Figure 1. Specifically, given a new sample  $\mathbf{y}$  in a new frame, the algorithm iteratively and alternatively updates basis matrix  $U$  and the approximation of  $\mathbf{y}$  with respect to  $U$ . The update is very efficient since only a few iterations are needed in practice and the convergence is guaranteed in theory. The representation is then combined with the particle filter framework [15] for tracking. At each frame, the likelihood of each particle is derived from its sparse reconstruction error using the learned basis  $U$ . Then, tracking is driven by the standard sequential Bayesian inference in particle filter.

In summary, there are two main contributions of our work: First, we propose to use constrained NMF for visual tracking. To the best of our knowledge, this is the first time constrained NMF has been used for tracking. Second, we develop a novel online learning algorithm for the constrained NMF with guaranteed low computational cost and low memory cost. The algorithm takes into account the smoothness of target appearance across frames and efficiently integrates new information into the learned appearance model. Benefiting from the combination of NMF with the sparsity, smoothness, and temporal continuity constraints, the two contributions not only make the proposed NMF-Tracker robust under different tracking environments, but also bring great run-time efficiency. We tested the proposed tracker on several sequences involving different challenges such as pose change, illumination variation and occlusion. In the experiments, the NMF-Tracker demonstrated excellent performances in comparison with eight alternative state-of-the-art trackers.

Although the proposed approach adopts the sparsity constraint, it is based on the NMF representation [22] and different from the sparse tracking approaches [29], [31], [26], [38], [32] which adopt the sparse representation [39]. In this paper the target is modeled as an additive combination of nonnegative components which are learned by a novel online NMF algorithm to enable our tracker to adapt to new scenario. However, in [29] the original intensity templates are used as the basis images and a heuristic approach is adopted to update the template set. Further, the smoothness constraint adopted in our approach accelerates the convergence of our tracker which is much faster than [29].

In the rest of this paper, we first describe in §2 target representation based on the constrained non-negative matrix factorization for visual tracking. After that, the proposed tracker is presented in §3, which focuses on the proposed online learning algorithm for the evolutionary appearance model. The experimental validation is then given in §4, followed by the conclusion in §5.

## 2 APPEARANCE MODELING

### 2.1 Particle filter

Before introducing the proposed appearance model, we briefly review the particle filter framework [15] that we use to integrate the proposed representation for visual tracking. Particle filter is aimed to estimate the posterior distribution of state

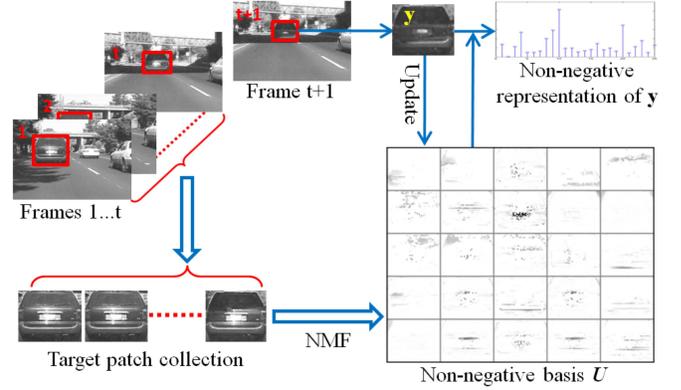


Fig. 1. Online NMF for target representation.

variables of a dynamic system. It uses a set of weighted particles to approximate the probability distribution of the state, which enables it to model the nonlinearity and non-Gaussianity of dynamics. Specifically, we denote the state of the system at time  $t$  by  $\mathbf{x}_t$  and the observation by  $\mathbf{y}_t$ . In addition, notations  $X_t = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t]$ ,  $Y_t = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_t]$  are used for the states and observations till time  $t$ , respectively. The particle filter adopts a weighted particle set  $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N_s}$  to approximate the posterior distribution  $p(\mathbf{x}_t|Y_t)$ . The state  $\mathbf{x}_t$  is estimated as  $\hat{\mathbf{x}}_t = \sum_{i=1}^{N_s} w_t^{(i)} \mathbf{x}_t^{(i)}$ .

The particle filter consists of two steps, one for state prediction and the other for model update. The two steps can be used to recursively estimate the posterior probability according to the following two rules:

$$p(\mathbf{x}_t|Y_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|Y_{t-1})d\mathbf{x}_{t-1}, \quad (1)$$

$$p(\mathbf{x}_t|Y_t) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|Y_{t-1})}{p(\mathbf{y}_t|Y_{t-1})}. \quad (2)$$

The integration in the estimation is simulated using particle samples, which are drawn from some importance distribution  $q(\mathbf{x}_t|X_{t-1}, Y_t)$ , and the weights of the particles should be updated as

$$w_t^{(i)} = w_{t-1}^{(i)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t|X_{t-1}, Y_t)}. \quad (3)$$

In our implementation,  $q(\mathbf{x}_t|X_{t-1}, Y_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$ , which is a Gaussian distribution. The weight of a sample is equal to  $p(\mathbf{y}_t|\mathbf{x}_t)$  after being updated, since the weights of particles are equally weighted before being updated due to resampling.

In our tracking system, we use a 3D state space that represents the position and scale of a target. Given a state  $\mathbf{x}$ , its corresponding observation  $\mathbf{y}$  is collected by first cropping a patch according to  $\mathbf{x}$  and then normalizing the patch. Then, within the particle filter framework, the tracking boils down to estimate the observation likelihood  $p(\mathbf{y}_t|\mathbf{x}_t)$ , which is calculated as Equation (7) in §2.3.

### 2.2 Constrained non-negative matrix factorization

It is commonly agreed that the appearance model for tracking targets should be able to reflect their variations across frames. Subspace representation has been popular toward this end.

In these methods, a target is usually treated as a linear combination of a set of basis, which can be either in a transferred low dimensional space (e.g. PCA) or in the original intensity space (e.g., template). We follow the same philosophy but seek more benefits from non-negativity constraint.

Let  $Y = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n] \in \mathbb{R}^{m \times n}$  denote  $n$  target samples (at different time), each with a dimension of  $m$ , we aim to build a representation using a set of non-negative basis  $U = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p] \in \mathbb{R}^{m \times p}$  such that

$$Y \approx UV, \quad \text{s.t. } U \geq 0, V \geq 0, \quad (4)$$

where  $V = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n] \in \mathbb{R}^{p \times n}$  denotes the linear coefficients for all samples. This is a standard non-negative matrix factorization problem [22]. As has been demonstrated in previous studies, such a representation naturally encodes both the part structures through non-negative basis ( $U \geq 0$ ) and the part composition through non-negative combination ( $V \geq 0$ ).

Aside from the non-negativity, in the context of visual tracking, we include the following constraints: (1) *sparsity* in  $V$ , which has been shown to be effective for visual tracking in [29], (2) *smoothness constraint*, which aims to preserve the locality [14] on the manifold such that neighboring samples in the original space should also be close in the new space spanned by  $U$ , and (3) *regularization* on  $U$ .

Addressing all the above constraints, we have the following (off line) objective function for the constrained NMF problem,

$$\begin{aligned} \mathcal{E}_{\text{off}}(U, V; Y) = & \|Y - UV\|_F^2 + \alpha \sum_i \|\mathbf{v}_i\|_1^2 \\ & + \beta \sum_{i>j} w_{i,j} \|\mathbf{v}_i - \mathbf{v}_j\|_2^2 + \gamma \|U\|_F^2, \end{aligned} \quad (5)$$

where  $w_{i,j}$  is defined as follows:

$$w_{i,j} = \begin{cases} 1, & \mathbf{y}_i \in N_k(\mathbf{y}_j) \text{ or } \mathbf{y}_j \in N_k(\mathbf{y}_i) \\ 0, & \text{otherwise} \end{cases}$$

where  $N_k(\mathbf{y}_i)$  denotes the  $k$  nearest neighbors of  $\mathbf{y}_i$ .  $\mathcal{E}_{\text{off}}$  has four terms: the first term tries to minimize the reconstruction error; the second term is the sparsity penalty, weighted by  $\alpha$ ; the third term is the smoothness constraint, similar with the locality constraint on manifold [14], weighted by  $\beta$ ; and the last term is a regularizer weighted by  $\gamma$ . Note that to encourage the sparseness the square of  $L1$  norm is employed due to its effectiveness and the efficiency of the resulting solution [17], [36]. This objective function is different from the NMF on manifold [8] in that we also consider the sparseness of the coefficient matrix  $V$ . This is because we assume that each point on the manifold can be represented by a sparse linear combination of basis.

To learn the representation usually requires an offline learning algorithm, which minimizes  $\mathcal{E}_{\text{off}}(U, V; Y)$  with respect to  $U$  and  $V$ ,

$$[\hat{U}, \hat{V}] = \arg \min_{U \geq 0, V \geq 0} \mathcal{E}_{\text{off}}(U, V; Y). \quad (6)$$

This is impractical for tracking due to the dynamically arrived data in  $Y$  and computational complexity. Instead we propose an efficient online solution described in §3.

## 2.3 Inference for visual tracking

In the tracking problem, our task is to estimate its observation likelihood  $p(\mathbf{y}_i|\mathbf{x})$  in the particle filter framework. This is derived from the distance, denoted as  $d(\mathbf{y}_i, U)$ , from  $\mathbf{y}_i$  to the target manifold with basis  $U$ . The distance  $d(\mathbf{y}_i, U)$  is calculated by the minimum reconstruction error subject to a sparsity constraint. Specifically, we first solve the following regularized least square problem,

$$\hat{\mathbf{v}}_i = \arg \min_{\mathbf{v}_i \geq 0} \|\mathbf{y}_i - U\mathbf{v}_i\|_2^2 + \alpha \|\mathbf{v}_i\|_1^2 + \beta \sum_{j<i} w_{i,j} \|\mathbf{v}_j - \mathbf{v}_i\|_2^2.$$

Then we have  $d(\mathbf{y}_i, U) = \|\mathbf{y}_i - U\hat{\mathbf{v}}_i\|_2^2$  and

$$p(\mathbf{y}_i|\mathbf{x}) \propto \exp\{-Cd(\mathbf{y}_i, U)\}, \quad (7)$$

where  $C$  is a positive constant.

## 3 VISUAL TRACKING THROUGH ONLINE NMF

As mentioned in the previous section, the traditional offline NMF learning algorithm is not applicable to visual tracking. Recently, the online NMF algorithm has been proposed for document analysis [9] and background modeling [7]. However, it does not handle the sparsity and smoothness, which are very important for tracking. For this reason, we design a novel online solution, CONMF, to incrementally update the basis  $U$  as new frame arrives.

### 3.1 Objective function for CONMF

Suppose we already have  $n$  samples  $Y$ , and now the  $(n+1)^{\text{th}}$  sample  $\mathbf{y} \in \mathbb{R}^m$  is coming. In tracking, it means that the target in current frame is cropped using the estimated state  $\mathbf{x}$  as a training sample. The basis  $U \in \mathbb{R}^{m \times p}$ , which is the appearance model the algorithm is going to learn, should be updated based on  $\mathbf{y}$ . If  $n = 0$ , we can initialize  $U$  to some random positive matrix or by some offline algorithm which may incorporate some prior information about the appearance of the target.

To use the new sample to update  $U$  incrementally, we get the (online) objective function of CONMF as follows:

$$\begin{aligned} \mathcal{E}_{\text{on}}(U, \mathbf{v}; Y, V, \mathbf{y}) = & \|Y - UV\|_F^2 + \lambda \|\mathbf{y} - U\mathbf{v}\|_2^2 \\ & + \alpha \|\mathbf{v}\|_1^2 + \beta \sum_i w_{n+1,i} \|\mathbf{v} - \mathbf{v}_i\|_2^2 + \gamma \|U\|_F^2, \end{aligned} \quad (8)$$

where  $\mathbf{v} \in \mathbb{R}^p$  is the approximation coefficient for  $\mathbf{y}$ . The parameters  $\alpha, \beta, \gamma$  are the same as in the offline objective function. The parameter  $\lambda$  is used to determine how important the new sample is. In our experiments, we set  $\lambda = 10$  to emphasize the appearance information from the new observation.

In the online tracking, it is generally impractical to store all the samples  $Y$  in memory. This raises a problem when calculating  $w_{n+1,i}$ . Fortunately, we can reasonably relax this requirement by observing that object appearance changes smoothly over time. Based on this observation, we assume that the object appearances in consecutive frames are similar to each other. As a result, we relax the online objective function  $\mathcal{E}_{\text{on}}$  as following

$$\begin{aligned} \mathcal{E}(U, \mathbf{v}; Y, V, \mathbf{y}) = & \|Y - UV\|_F^2 + \lambda \|\mathbf{y} - U\mathbf{v}\|_2^2 \\ & + \alpha \|\mathbf{v}\|_1^2 + \beta \|\mathbf{v} - \mathbf{v}_n\|_2^2 + \gamma \|U\|_F^2, \end{aligned} \quad (9)$$

where  $\mathbf{v}_n$  is the approximation coefficient of sample  $\mathbf{y}_n$ , which is cropped using the estimated state in last frame. In this objective function, we only consider the smoothness between two consecutive frames. Theoretically we can extend the constraints to more consecutive frames, but practically we observe no significant improvement in performance. Note that from the definition in (9) it seems that we still need to store  $V$  that contains coefficients of all previously seen samples. Fortunately this is not the case as shown in the following subsection.

### 3.2 Optimization

Now the task is to minimize the objective function  $\mathcal{E}$  of CONMF defined in (9). However, it is not convex with respect to jointly  $U$ , and  $\mathbf{v}$ . Similar with traditional NMF [23], we devise an iterative approach that uses two steps to alternatively update  $U$  and  $\mathbf{v}$ . In the following, we use two notations in describing the two steps: (1) Superscript “ $t$ ” is used to indicate the  $t^{\text{th}}$  updating iteration; and (2) Subscript outside parenthesis is used to indicate the element in the vector (or matrix), e.g., “ $(\mathbf{v})_i$ ” for the  $i^{\text{th}}$  element of  $\mathbf{v}$  and “ $(U)_{i,j}$ ” for the  $(i,j)^{\text{th}}$  element of  $U$ .

#### 3.2.1 Step 1: Fix $U$ , update $\mathbf{v}$

First, we fix the basis  $U$ , and update  $\mathbf{v}$  to decrease the objective function.

$$\begin{aligned}
& \arg \min_{\mathbf{v} \geq 0} \mathcal{E}(U, \mathbf{v}; Y, V, \mathbf{y}) \\
&= \arg \min_{\mathbf{v} \geq 0} \lambda \|\mathbf{y} - U\mathbf{v}\|_2^2 + \alpha \|\mathbf{v}\|_1^2 + \beta \|\mathbf{v} - \mathbf{v}_n\|_2^2 \\
&= \arg \min_{\mathbf{v} \geq 0} \|\mathbf{y} - U\mathbf{v}\|_2^2 + \alpha' \|\mathbf{v}\|_1^2 + \beta' \|\mathbf{v}\|_2^2 - 2\beta' \mathbf{v}^\top \mathbf{v}_n \\
&= \arg \min_{\mathbf{v} \geq 0} \left\| \begin{bmatrix} \mathbf{y} \\ 0 \\ \mathbf{0}_{p \times 1} \end{bmatrix} - \begin{bmatrix} U \\ \sqrt{\alpha'} \mathbf{1}_{1 \times p} \\ \sqrt{\beta'} \mathbf{I}_p \end{bmatrix} \mathbf{v} \right\|_2^2 - 2\beta' \mathbf{v}^\top \mathbf{v}_n \\
&= \arg \min_{\mathbf{v} \geq 0} \|\mathbf{y}' - U'\mathbf{v}\|_2^2 - 2\beta' \mathbf{v}^\top \mathbf{v}_n, \tag{10}
\end{aligned}$$

where  $\alpha' = \alpha/\lambda$ ,  $\beta' = \beta/\lambda$ ,  $\mathbf{y}' = [\mathbf{y}^\top, 0, \mathbf{0}_{1 \times p}]^\top$ ,  $U' = [U^\top, \sqrt{\alpha'} \mathbf{1}_{p \times 1}, \sqrt{\beta'} \mathbf{I}_p]^\top$ , and  $\mathbf{I}_p$  is the identity matrix of size  $p$ .

Denote  $\mathcal{E}'(\mathbf{v}) = \|\mathbf{y}' - U'\mathbf{v}\|_2^2 - 2\beta' \mathbf{v}^\top \mathbf{v}_n$ , its gradient is

$$\nabla \mathcal{E}' = -2U'^\top \mathbf{y}' + 2U'^\top U' \mathbf{v} - 2\beta \mathbf{v}_n. \tag{11}$$

To minimize  $\mathcal{E}'$ , we use an iterative algorithm inspired by the work in [23]. The algorithm is based on the following theorem and definition:

**Definition 1:** [23]  $G(\mathbf{v}; \mathbf{v}')$  is an auxiliary function for  $F(\mathbf{v})$  if  $G(\mathbf{v}; \mathbf{v}')$  satisfies the following two conditions

$$G(\mathbf{v}; \mathbf{v}') \geq F(\mathbf{v}), \quad G(\mathbf{v}; \mathbf{v}) = F(\mathbf{v}). \tag{12}$$

**Lemma 1:** [23] If  $G$  is an auxiliary function, then  $F$  is nonincreasing under the update

$$\mathbf{v}^{t+1} = \arg \min_{\mathbf{v}} G(\mathbf{v}; \mathbf{v}^t). \tag{13}$$

**Theorem 1:** Let  $K(\mathbf{v}^t)$  be the diagonal matrix such that  $(K(\mathbf{v}^t))_{i,j} = \delta_{ij} (U'^\top U' \mathbf{v}^t)_i / (\mathbf{v}^t)_i$ , then

$$G(\mathbf{v}; \mathbf{v}^t) = \mathcal{E}'(\mathbf{v}^t) + (\mathbf{v} - \mathbf{v}^t) \nabla \mathcal{E}'(\mathbf{v}^t) + (\mathbf{v} - \mathbf{v}^t)^\top K(\mathbf{v}^t) (\mathbf{v} - \mathbf{v}^t) \tag{14}$$

is an auxiliary function for  $\mathcal{E}'(\mathbf{v})$ .

**Proof:**

For the first condition in (12), we have obviously  $G(\mathbf{v}; \mathbf{v}) = \mathcal{E}'(\mathbf{v})$ .

For the second condition, the proof of  $G(\mathbf{v}; \mathbf{v}^t) \geq F(\mathbf{v})$  is the same as in [23], since the only difference between our objective function and the one in [23] is in  $\nabla \mathcal{E}'(\mathbf{v})$ . ■

So, according to Lemma 1, we can update  $\mathbf{v}$  in the following way:

$$\mathbf{v}^{t+1} = \arg \min_{\mathbf{v}} G(\mathbf{v}; \mathbf{v}^t).$$

Specifically, for the  $i^{\text{th}}$  element we have the following update rule:

$$\begin{aligned}
(\mathbf{v}^{t+1})_i &= (\mathbf{v}^t)_i - \frac{1}{2} (K(\mathbf{v}^t)^{-1} \nabla \mathcal{E}'(\mathbf{v}^t))_i \\
&= (\mathbf{v}^t)_i - \frac{(\mathbf{v}^t)_i}{(U'^\top U' \mathbf{v}^t)_i} (-U'^\top \mathbf{y}' + U'^\top U' \mathbf{v} - \beta' \mathbf{v}_n)_i \\
&= (\mathbf{v}^t)_i \frac{(U'^\top \mathbf{y}')_i + \beta' (\mathbf{v}_n)_i}{(U'^\top U' \mathbf{v}^t)_i}. \tag{15}
\end{aligned}$$

The nonnegativity constraint naturally holds. Note here  $U'^\top \mathbf{y}' = U^\top \mathbf{y}$  and  $U'^\top U' = U^\top U + \alpha' \mathbf{1}_{p \times p} + \beta' \mathbf{I}_p$ .

#### 3.2.2 Step 2: Fix $\mathbf{v}$ , update $U$

Now we fix  $\mathbf{v}$ , update non-negative basis  $U$  to decrease the objective function. We have

$$\begin{aligned}
& \arg \min_{U \geq 0} \mathcal{E}(U, \mathbf{v}; Y, V, \mathbf{y}) \\
&= \arg \min_{U \geq 0} \|Y - UV\|_F^2 + \lambda \|\mathbf{y} - U\mathbf{v}\|_2^2 + \gamma \|U\|_F^2 \\
&= \arg \min_{U \geq 0} \|[Y, \sqrt{\lambda} \mathbf{y}] - U[V, \sqrt{\lambda} \mathbf{v}]\|_F^2 + \|\sqrt{\gamma} U\|_F^2 \\
&= \arg \min_{U \geq 0} \|[Y, \sqrt{\lambda} \mathbf{y}, \mathbf{0}_{m \times p}] - U[V, \sqrt{\lambda} \mathbf{v}, \sqrt{\gamma} \mathbf{I}_p]\|_F^2 \\
&= \arg \min_{U \geq 0} \|Y' - UV'\|_F^2, \tag{16}
\end{aligned}$$

where  $Y' = [Y, \sqrt{\lambda} \mathbf{y}, \mathbf{0}_{m \times p}]$  and  $V' = [V, \sqrt{\lambda} \mathbf{v}, \sqrt{\gamma} \mathbf{I}_p]$ . This is now part of a standard NMF problem. We can therefore update  $U$  using the traditional NMF updating rule [22]. Specifically, for each element  $(U)_{i,j}$ ,

$$(U)_{i,j} = (U)_{i,j} \frac{(Y' V'^\top)_{i,j}}{(U V' V'^\top)_{i,j}}. \tag{17}$$

Note  $Y' V'^\top = Y V^\top + \lambda \mathbf{y} \mathbf{v}^\top$  and  $V' V'^\top = V V^\top + \lambda \mathbf{v} \mathbf{v}^\top + \gamma \mathbf{I}$ , where  $Y V^\top$  and  $V V^\top$  do not change in the iteration for the sample at time  $n + 1$ . This enables us to use values of  $Y V^\top$  and  $V V^\top$  directly without storing all individual  $\mathbf{v}_i$  for  $i = 1, 2, \dots, n$ . Similar to traditional NMF, this updating rule can be applied multiple times. In our experiments, applying this updating rule one time per frame is enough, which probably results from the fact that a single sample usually changes  $U$  slightly.

### 3.2.3 Convergence analysis

The objective function has an obvious lower bound 0. Both updating rules for  $U$  and  $\mathbf{v}$  are non-increasing for the objective function. Consequently, the optimization converges. In practice, we observed that it usually takes no more than 10 iterations. Also, the stationary point of the updating rules (15) and (17) is the Karush-Kuhn-Tucker (KKT) point for equation (9) (see Appendix for proof).

### 3.2.4 Complexity analysis

As an online algorithm, it is important that the complexity of algorithm in each frame does not increase over time. It is obviously that the time complexity of our algorithm does not depend on the time instance  $n$ .

For the space complexity, in the  $n^{\text{th}}$  frame, the proposed algorithm only needs to store the basis  $U$ , the current sample  $\mathbf{y}$ , the current coefficient vector  $\mathbf{v}$ , the neighboring sample coefficient vector  $\mathbf{v}_{n-1}$ , parameters  $\alpha, \beta, \gamma, \lambda$ , and  $VV^T, YV^T$  for the updating rules. Note  $V_n V_n^T = V_{n-1} V_{n-1}^T + \mathbf{v} \mathbf{v}^T$  and  $Y_n V_n^T = Y_{n-1} V_{n-1}^T + \mathbf{y} \mathbf{v}^T$ , where subscript  $n$  or  $n-1$  indicates the status of  $V$  and  $Y$  at time  $n$  or  $n-1$ . In this way, the algorithm maintains  $VV^T \in \mathbb{R}^{p \times p}$  and  $YV^T \in \mathbb{R}^{m \times p}$ . Also, given  $VV^T$  and  $YV^T$ , the algorithm does not need  $V$  and  $Y$  any more. Therefore, the memory cost does not depend on  $n$ , and thus it keeps constant over time, which means once the proposed algorithm is assigned some space at the initial frame, it does not require any additional space later.

## 3.3 Tracking algorithm

Combining the CONMF for target representation with the particle framework, we summarize the proposed NMF-Tracker in Algorithm 1.

---

### Algorithm 1 NMF-Tracker

---

- 1: Initialize  $U$  in the first frame ( $U$  may incorporate some prior knowledge)
  - 2: **for** each following frame **do**
  - 3:   Draw sample particles
  - 4:   **for** each particle  $\mathbf{x}$  **do**
  - 5:     Prepare observation  $\mathbf{y}$  for  $\mathbf{x}$
  - 6:     Compute the likelihood  $p(\mathbf{y}|\mathbf{x})$  using (7)
  - 7:     Update the particle weight
  - 8:   **end for**
  - 9:   Estimate the state of the object
  - 10:   Get the tracking sample  $\mathbf{y}$
  - 11:   Update the  $U$  by minimizing  $\mathcal{E}(U, \mathbf{v}; Y, V, \mathbf{y})$
  - 12:   Update auxiliary variables  $VV^T, YV^T$
  - 13:   Propagate samples with resampling
  - 14: **end for**
- 

## 4 EXPERIMENTS

To evaluate the performance of the proposed algorithm, we collected a set of videos, including both indoor and outdoor scenes where the targets undergo scale, pose change, illumination variation, and occlusions.

## 4.1 Settings

The proposed CONMF for sparse non-negative representation on manifold is embedded into the particle filter tracking framework. One of the goals of this work is to demonstrate that using CONMF results in a more robust and stable tracker. For this reason, the parameters of NMF-Tracker are set as:  $\lambda, \alpha, \beta$  and  $\gamma$  are fixed at 10, 1, 100 and 0.01 for all experiments. The number of iterations for the computation of coefficient vector over the learned basis  $U$  for each particle is set to 10. This is enough for all our testing video sequences. Note that, for sequence *car4*, we found that it needs only one iteration to converge. 100 particles are used to propagate the probability distribution of the target state for all the videos except *girl*, for which 300 particles are used since the variation is greater there. For fast convergence, the initial value of coefficient vector of each particle is set as the coefficient vector of the estimated state of the target in the previous frame. In practice, we find that it needs much more iteration steps (at least tens of thousands) to generate the part-based basis images like the templates shown in Fig.1. Due to the computation efficiency, we set the maximum number of iterations to be 300. Although the final templates are not too sparse, this does not affect the performance much. Our experiments were performed on ten publicly available video sequences, as well as two of our own. In our mixed Matlab/C implementation, our tracker runs about 150 frames per second on a PC with Intel i7-2600 CPU (3.4GHz, 50% CPU usage), using sequence *car4*.

The proposed NMF-Tracker is compared with eight state-of-the-art trackers: GKT tracker [35], OAB tracker [12], color based probabilistic tracker(CBPT) [33], MIL tracker [3], VTD tracker [20], IVT tracker [34], L1 tracker [5], and ICTL tracker [40], [41]. For all the compared trackers, we use the publicly available codes or the codes from the original authors and the same parameters are used as the authors.

## 4.2 Tracking results

In this subsection, we show some tracking results of our proposed tracker and the comparison results with other trackers will be illustrated in the followed two subsections.

The results on sequence *mhyang* is illustrated in Figure 2 [34]. The illumination of the target changes frequently (#414, #710, #1100) and our tracker can adaptively update the model for these variations.

The intensity of the *singer* [20] in Figure 3 changes dramatically (#85, #113) and the scale is also with large variations. Due to the proposed appearance model and the particle filter tracking technique, our tracker can follow the target throughout the sequence.

The sequence *redteam* is got from [46]. As shown in Figure 4, the car is with frequent scale variations and small occlusions. Our tracker can successfully recover the scale and follow the target.

Figure 5 illustrates the results on *dog* [34]. The pose of the target changes frequently (#102, #171, #599) and the CONMF can learn these poses into the appearance model. Also, the scale of the target changes a lot (#1002, #1334) and our tracker can estimate the state of target accurately.

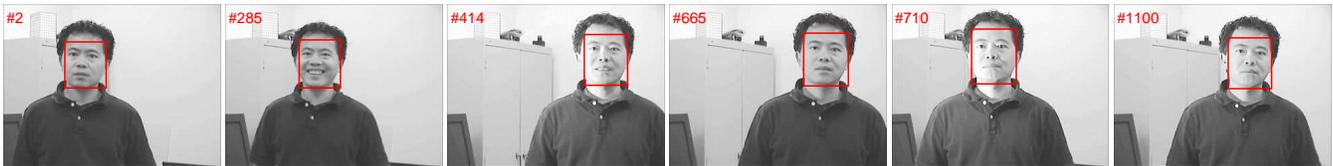


Fig. 2. Tracking results of the NMF-Tracker on *mhyang*.



Fig. 3. Tracking results of the NMF-Tracker on *singer*.



Fig. 4. Tracking results of the NMF-Tracker on *redteam*.



Fig. 5. Tracking results of the NMF-Tracker on *dog*.



Fig. 6. Tracking results of the NMF-Tracker on *doll*.

*doll* [30] is a very long sequence which lasts around 3000 frames and we use this sequence to test if our tracker is qualified for the long-term tracking. As shown in Figure 6, the target is undergoing large scale variations (#920), occlusions (#1714, #2637, #2983) and pose changes (#260, #2356). Our tracker can handle these challenging conditions and successfully track the target throughout the sequence.

#### 4.2.1 Tracking results with different constraints

In this subsection we illustrate the effect of different constraints in Equation (9) using sequence *dudek* [16]. Figure 7 shows the results for the trackers with different constraints. The tracking results of the tracker with all the constraints are shown with red rectangle, the ones without smoothness and sparseness are shown with blue ( $\beta = 0$ ) and yellow ( $\alpha = 0$ ) respectively, and green represents the tracker without all the constraints ( $\alpha = 0, \beta = 0, \gamma = 0$ ). As shown in this figure, the trackers without constraints fail to track the target due to the background clutter (#738) and pose changes (#755). While the tracker with all the constraints can successfully follow the

target throughout the sequence.

### 4.3 Qualitative comparison

In this subsection, we qualitatively compare the tracking results of our proposed NMF-Tracker together with other trackers. To avoid clutter, only five methods are given exemplary image illustration: proposed NMF-Tracker (red), IVT(blue), ICTL (yellow), L1 (green) and MIL (purple). The results of other compared trackers are listed in the following quantitative subsection. We first tested our algorithm using the sequence, *car4*, which presents challenging lighting and scale variation. The car undergoes drastic illumination changes as it passes beneath a bridge and under trees. Tracking results on some frames are demonstrated in Figure 8, from which we can see that the proposed algorithm can track the target robustly.

In sequence *david* [34], a person moves from a dark room toward a bright area while changing his pose, facial expressions and taking off his glasses. Notice that there is also a large scale variation in the target relative to the camera.



Fig. 7. The results of our tracker with different constraints on *dudek*. Red: with all the constraints. Blue: without smoothness ( $\beta = 0$ ). Yellow: without sparseness ( $\alpha = 0$ ). Green: without all the constraints ( $\alpha = 0, \beta = 0, \gamma = 0$ ).



Fig. 8. Comparison results on *car4*. Red: proposed NMF-Tracker, Blue: IVT, Yellow: ICTL, Green: L1, Purple: MIL.



Fig. 9. Comparison results on *david*. Red: proposed NMF-Tracker, Blue: IVT, Yellow: ICTL, Green: L1, Purple: MIL.



Fig. 10. Comparison results on *girl*. Red: proposed NMF-Tracker, Blue: IVT, Yellow: ICTL, Green: L1, Purple: MIL.

The results are shown in Figure 9. Our algorithm is able to track the target throughout the sequence.

In the first two sequences, NMF-Tracker is comparable to best performed tracker IVT [34] as shown in 1, but the number of particles employed in our tracker is 80% less than that used in IVT.

Figure 10 shows the results for sequence *girl* [6] which contains a girl moving in different pose, scale and occlusion. Once initialized in the first frame, our algorithm is able to track the target object accurately as it experiences 360 degree out of plane rotation, long-term severe occlusion and scale variation, while all other four algorithms are more or less inaccurate, especially when there is rotation(#119) or occlusion(#433). L1 tracker is also accurate when there is occlusion, however, it is not as accurate when there is rotation(#81).

The tracking results on the sequence *ped* are shown in Figure 11. The pedestrian's appearance is changing over time and experiences a short time partial occlusion by a pole. Again, our method obtains good tracking results. Besides, MIL and ICTL can also track the target accurately, while other trackers drift apart and get stuck to the background.

Figure 12 shows the comparison results on the sequence *boy*, whose pose is changing quickly and frequently. Motion blur occasionally happens, which could jeopardize image

features temporarily. The results show that our algorithm faithfully models the appearance of the target and is able to track the target in the presence of motion blur.

We finally test our algorithm on a very challenging sequence *car*, which is cropped from a movie. The target undergoes dramatical scale variation and partial occlusion, and the results are shown in Figure 13, from which we can see that our NMF-Tracker adapts to the scale change and occlusion very well(#166,#175). In comparison, other trackers meet problems when there are occlusions or scale changes.

#### 4.4 Quantitative evaluation

Here we quantitatively compare NMF-Tracker with all the eight trackers mentioned above. For all sequences used, we manually labeled the ground truth bounding box of the target in each frame.

The first quantitative criterion to evaluate the performance is the center error, which is defined as the Euclidean distance between the center of tracking result and the center of the ground truth bounding box. The quantitative errors are shown in Figure 14, with one subfigure per sequence. Our NMF-Tracker is represented by red line, and the results show that in all testing sequences NMF-Tracker performs better than or tiers the best of the state-of-the-art algorithms.

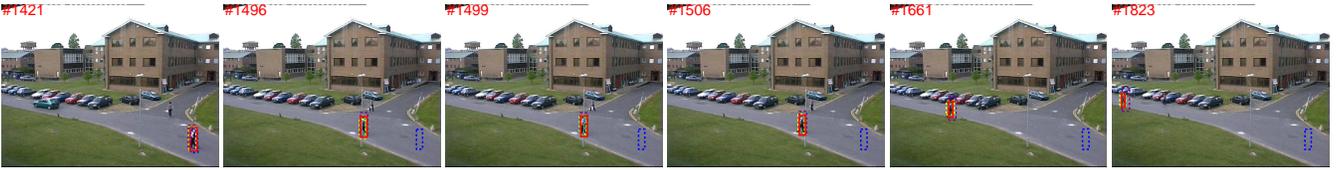


Fig. 11. Comparison results on *ped.* Red: proposed NMF-Tracker, Blue: IVT, Yellow: ICTL, Green: L1, Purple: MIL.



Fig. 12. Comparison results on *boy.* Red: proposed NMF-Tracker, Blue: IVT, Yellow: ICTL, Green: L1, Purple: MIL.



Fig. 13. Comparison results on *car.* Red: proposed NMF-Tracker, Blue: IVT, Yellow: ICTL, Green: L1, Purple: MIL.

Alg.	car4	david	girl	ped.	boy	car	ave.
GKT	121.3	65.8	9.4	18.2	50.7	140.0	67.5
MIL	49.4	25.5	12.1	5.2	7.7	16.7	19.4
OAB	53.4	22.9	8.6	7.3	10.5	26.3	21.5
CBPT	45.1	92.4	7.3	5.8	38.9	25.5	35.8
ICTL	20.6	44.6	10.5	5.6	22.2	22.5	21.0
VTD	19.5	7	<b>2.7</b>	37.4	<b>2.5</b>	34.0	17.1
IVT	<b>3.9</b>	<b>3.5</b>	25.4	398.8	28.6	17.1	79.5
L1	5.8	19.1	5.1	4.5	3.7	18.3	9.4
NMF	6.1	4.5	4.3	<b>4.4</b>	3.3	<b>16.3</b>	<b>6.4</b>

TABLE 1

The mean tracking errors measured using the Euclidian distance between center points. The last column is the average over all sequences.

The second quantitative criterion employed for evaluation is the average center error, which is defined as the average of center error over all the frames of a given sequence. The evaluation results are listed in Table 1. For the first five sequences, average center errors of NMF-Tracker are 6.1, 4.5, 4.3, 4.4 and 3.3 respectively, which are small enough. For the last challenging sequence *car*, our tracker gains an average center error of 16.3, which is not ideal, however, it is still the best one among all the algorithms we compared. These statistical results again verify that NMF-Tracker outperforms or ties the best of the state-of-the-art trackers.

#### 4.5 Discussion

Our proposed approach achieves robust tracking performance by efficiently integrating new information into the appearance model. The combination of NMF with the sparsity, smoothness, and temporal continuity constraints makes the proposed NMF-Tracker robust under different tracking environments. Furthermore, the online algorithm brings great run-time efficiency. That said, there are challenging cases our tracker

meets problems, such as when dealing with full or long term occlusion, dramatical non-rigid motion, abrupt motion or moving out of the frame, etc. Figure 15 shows some example images of challenging sequences where the proposed tracker performs poorly.

To further improve the robustness of the proposed tracker, the fragment-based representation can be adopted, which has been verified to be robust to the partial occlusion [1]. Some advanced motion models [20] can also be employed to handle the abrupt motion cases. Furthermore, the generative model adopted in our tracker can be further enhanced by cooperating with the discriminative model, which has been proved to be effective for tracking in [47].

## 5 CONCLUSION

This paper proposes an efficient tracker based on the sparse non-negative matrix factorization. Instead of traditional offline learning, we design an online algorithm, CONMF, which incrementally learns the nonnegative basis for object representation. This new representation enables the proposed tracker to adapt to severe appearance changes over time. Moreover, the tracker is very efficient even when we take into consideration of the sparseness and smoothness (on manifold) constraints. The experimental results show that the proposed algorithm is able to track target accurately with challenging appearance variations.

## APPENDIX

Introducing Lagrangian multipliers  $\eta \in \mathbb{R}^{m \times p}$  and  $\theta \in \mathbb{R}^{p \times n}$ , we have

$$\mathcal{J}(U, \mathbf{v}; Y, V, \mathbf{y}) = \|Y - UV\|_F^2 + \lambda \|\mathbf{y} - U\mathbf{v}\|_2^2 + \alpha \|\mathbf{v}\|_1^2 + \beta \|\mathbf{v} - \mathbf{v}_n\|_2^2 + \gamma \|U\|_F^2 - \text{tr}(\eta U^T) - \text{tr}(\theta V^T).$$

Taking derivatives, we have

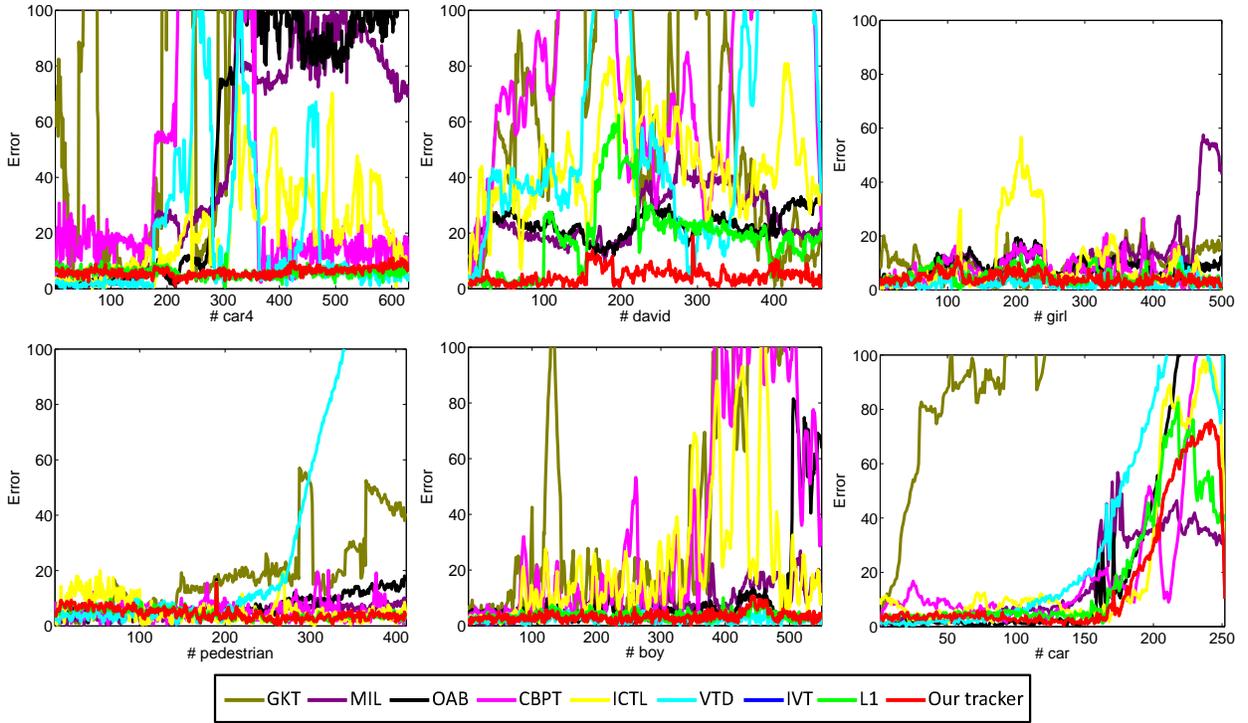


Fig. 14. Quantitative Errors (in Pixels).

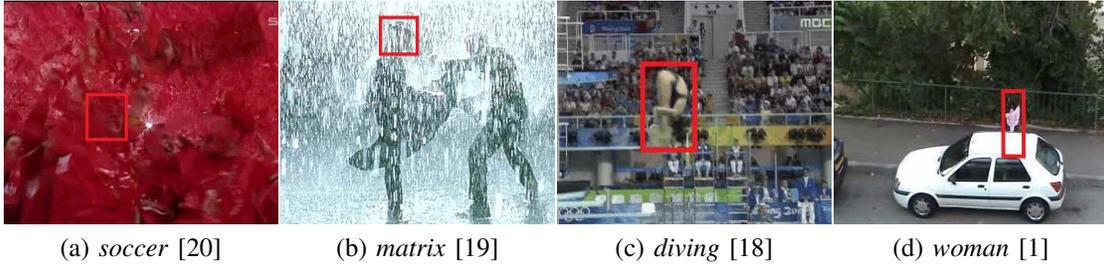


Fig. 15. Challenging sequences to our tracker. The images in (a)-(c) are cropped for better illustration. The targets are indicated by the red bounding boxes.

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial U} &= 2(UVV^T - YV^T + \lambda U\mathbf{v}\mathbf{v}^T - \lambda \mathbf{y}\mathbf{v}^T + \gamma U) - \eta, \\ \frac{\partial \mathcal{J}}{\partial \mathbf{v}} &= -2(\lambda U^T \mathbf{y} + \lambda U^T U\mathbf{v} + \alpha \|\mathbf{v}\|_1 \mathbf{1}_{p \times 1} + \beta \mathbf{v} - \beta \mathbf{v}_n) - \theta.\end{aligned}$$

Setting  $\frac{\partial \mathcal{J}}{\partial U} = 0$  and  $\frac{\partial \mathcal{J}}{\partial \mathbf{v}} = 0$ , we have

$$\begin{aligned}\eta &= 2UVV^T - 2YV^T + 2\lambda U\mathbf{v}\mathbf{v}^T - 2\lambda \mathbf{y}\mathbf{v}^T + 2\gamma U, \\ \theta &= -2\lambda U^T \mathbf{y} + 2\lambda U^T U\mathbf{v} + 2\alpha \|\mathbf{v}\|_1 \mathbf{1}_{p \times 1} + 2\beta \mathbf{v} - 2\beta \mathbf{v}_n.\end{aligned}$$

Then, the KKT condition requires

$$\begin{aligned}2(UVV^T - YV^T + \lambda U\mathbf{v}\mathbf{v}^T - \lambda \mathbf{y}\mathbf{v}^T + \gamma U)_{i,j} U_{i,j} &= 0, \\ 2(-\lambda U^T \mathbf{y} + \lambda U^T U\mathbf{v} + \alpha \|\mathbf{v}\|_1 \mathbf{1}_{p \times 1} + \beta \mathbf{v} - \beta \mathbf{v}_n)_i \mathbf{v}_i &= 0.\end{aligned}$$

The stationary point of the updating rule (15) is

$$\mathbf{v}_i = \mathbf{v}_i \frac{(U^T \mathbf{y})_i + \beta' (\mathbf{v}_n)_i}{(U^T U \mathbf{v})_i}$$

So the condition below should hold:

$$\begin{aligned}\mathbf{v}_i [(U^T \mathbf{y})_i + \beta' (\mathbf{v}_n)_i - (U^T U \mathbf{v})_i] &= 0 \\ \mathbf{v}_i [(U^T \mathbf{y})_i + \beta' (\mathbf{v}_n)_i - ((U^T U + \alpha' \mathbf{1}_{p \times p} + \beta' \mathbf{I}_p) \mathbf{v})_i] &= 0 \\ \mathbf{v}_i (U^T \mathbf{y} + \beta' \mathbf{v}_n - U^T U \mathbf{v} - \alpha' \|\mathbf{v}\|_1 \mathbf{1}_{p \times 1} - \beta' \mathbf{v})_i &= 0\end{aligned}$$

where  $\alpha' = \alpha/\lambda$  and  $\beta' = \beta/\lambda$ .

The stationary point of the updating rule (17) is

$$(U)_{i,j} = (U)_{i,j} \frac{(Y'V'^T)_{i,j}}{(UV'V'^T)_{i,j}}$$

The following condition should hold:

$$\begin{aligned}(U)_{i,j} [(Y'V'^T)_{i,j} - (UV'V'^T)_{i,j}] &= 0 \\ (U)_{i,j} [(YV^T + \lambda \mathbf{y}\mathbf{v}^T)_{i,j} - (U(VV^T + \lambda \mathbf{v}\mathbf{v}^T + \gamma \mathbf{I}))_{i,j}] &= 0 \\ (U)_{i,j} (YV^T + \lambda \mathbf{y}\mathbf{v}^T - UVV^T - \lambda U\mathbf{v}\mathbf{v}^T - \gamma U)_{i,j} &= 0\end{aligned}$$

We find that the stationary point satisfies the KKT condition by comparing them.

## REFERENCES

- [1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. *CVPR*, 2006.
- [2] S. Avidan. Ensemble Tracking. *CVPR*, 2005.
- [3] B. Babenko, M.-H. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. *CVPR*, 2009.
- [4] A. Balan and M. Black. An adaptive appearance model approach for model-based articulated object tracking. *CVPR*, 2006.
- [5] C. Bao, Y. Wu, H. Ling, and H. Ji. Real Time Robust L1 Tracker Using Accelerated Proximal Gradient Approach. *CVPR*, 2012.
- [6] S. Birchfield. Elliptical Head Tracking Using Intensity Gradients and Color Histograms. *CVPR*, 232–237, 1998.
- [7] S.S. Bucak and B. Günsel. Incremental subspace learning via nonnegative matrix factorization. *Pattern Recognition*, 42(5):788–797, 2006.

- [8] D. Cai, X. He, X. Wang, H. Bao, and J. Han. Locality preserving non-negative matrix factorization. *IJCAI*, 2009.
- [9] B. Cao, D. Shen, J. Sun, X. Wang, Q. Yang, and Z. Chen. Detect and track latent factors with online nonnegative matrix factorization. *IJCAI*, 2007.
- [10] P. Chockalingam, N. Pradeep, and S. Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. *ICCV*, 2009.
- [11] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *PAMI*, 25(5):564–575, 2003.
- [12] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via online boosting. *BMVC*, 2006.
- [13] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 20(10):1025–1039, 1998.
- [14] X. He, P. Niyogi. Locality preserving projections. *NIPS*, 2003.
- [15] M. Isard and A. Blake. Condensation—conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.
- [16] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust Online Appearance Models for Visual Tracking. *PAMI*, vol. 25, no. 10, pp. 1296–1311, 2003.
- [17] H. Kim and H. Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 1495–1502, 23(12), 2007.
- [18] Junseok Kwon, and Kyoung Mu Lee. Tracking of a Non-Rigid Object via Patch-based Dynamic Appearance Modeling and Adaptive Basin Hopping Monte Carlo Sampling. *CVPR*, 2009.
- [19] Junseok Kwon, and Kyoung Mu Lee. Tracking by Sampling Trackers. *ICCV*, 2011.
- [20] J. Kwon and K. M. Lee. Visual tracking decomposition. *CVPR*, 1269–1276, 2010.
- [21] C.-S. Lee and A.M. Elgammal. Modeling view and posture manifolds for tracking. *ICCV*, 2007.
- [22] D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.
- [23] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. *NIPS*, 2001.
- [24] C. Leistner, A. Saffari, and H. Bischof. Miforests: Multiple-instance learning with randomized trees. *ECCV*, 2010.
- [25] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick and A. Hengel. A Survey of Appearance Models in Visual Object Tracking. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2013, in press.
- [26] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Robust tracking using local sparse appearance model and K-selection. *CVPR*, 2011.
- [27] L. Lu and G. D. Hager. A nonparametric treatment for location/segmentation based visual tracking. *CVPR*, 2007.
- [28] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *PAMI*, 26(6):810–815, 2004.
- [29] X. Mei and H. Ling. Robust visual tracking using  $\ell_1$  minimization. *ICCV*, 2009.
- [30] X. Mei and H. Ling. Robust Visual Tracking and Vehicle Classification via Sparse Representation. *PAMI*, vol. 33, no. 11, pp. 2259–2271, 2011.
- [31] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. Minimum Error Bounded Efficient L1 Tracker with Occlusion Detection. *CVPR*, 2011.
- [32] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. Efficient Minimum Error Bounded Particle Resampling L1 Tracker with Occlusion Detection. *IEEE Transactions on Image Processing*, 22(7):2661–2675, 2013.
- [33] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. *ECCV*, 2002.
- [34] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1-3):125–141, 2008.
- [35] C. Shen, J. Kim and H. Wang. Generalized Kernel-Based Visual Tracking. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 119–130, 20(1), 2010.
- [36] B. Shen and L. Si. Nonnegative matrix factorization clustering on multiple manifolds. *AAAI*, 2010.
- [37] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel Tracking. *ICCV*, 2011.
- [38] Q. Wang and M.-H. Yang. Online discriminative object tracking with local sparse representation. *IEEE Workshop on the Applications of Computer Vision (WACV)*, 2012.
- [39] J. Wright, A.Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust Face Recognition via Sparse Representation. *PAMI*, 31(1):210–227, 2009.
- [40] Y. Wu, J. Cheng, J. Wang, and H. Lu. Real-time visual tracking via incremental covariance tensor learning. *ICCV*, 2009.
- [41] Y. Wu, J. Cheng, J. Wang, H. Lu., J. Wang, H. Ling, E. Blasch, and L. Bai. Real-time Probabilistic Covariance Tracking with Efficient Model Update. *IEEE Transactions on Image Processing*, 5(21): 2824–2837 2012.
- [42] Y. Wu, J. Lim, and M.-H. Yang. Online Object Tracking: A Benchmark. *CVPR*, 2013.
- [43] Y. Wu, B. Shen, and H. Ling. Online Robust Image Alignment via Iterative Convex Optimization. *CVPR*, 2012.
- [44] Y. Wu, H. Ling, J. Yu, F. Li, X. Mei, and E. Cheng. Blurred Target Tracking by Blur-driven Tracker. *ICCV*, 2011.
- [45] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), 2006.
- [46] VIVID dataset. [Online] Available: <http://vision.cse.psu.edu/data/vividEval/datasets/datasets.html>.
- [47] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang. Robust Object Tracking via Sparsity-based Collaborative Model. *CVPR*, 2012.



**Yi Wu** received the BS degree in automation from Wuhan University of Technology, China, in 2004, and the PhD degree from Institute of Automation, Chinese Academy of Sciences, China, in Pattern Recognition and Intelligent Systems in 2009. Since fall 2009, he has been a lecturer at Nanjing University of Information Science and Technology. From May 2010 to June 2012, he was a postdoctoral fellow at Temple University, USA. From July 2012 to present, he is working as a postdoctoral fellow at University of California at Merced. His research interests include computer vision, medical image analysis, multimedia analysis, and machine learning.



**Bin Shen** Bin Shen is a PhD student in Computer Science, Purdue University, West Lafayette. Before that, he got his BS and MS degrees in 2007 and 2009 respectively, both from EE, Tsinghua University, China, and another MS degree in CS from Purdue University, West Lafayette in 2011. His current research interests include computer vision, data mining and machine learning techniques.



**Haibin Ling** received the B.S. degree in mathematics and the MS degree in computer science from Peking University, China, in 1997 and 2000, respectively, and the PhD degree from the University of Maryland, College Park, in Computer Science in 2006. From 2000 to 2001, he was an assistant researcher at Microsoft Research Asia. From 2006 to 2007, he worked as a postdoctoral scientist at the University of California Los Angeles. After that, he joined Siemens Corporate Research as a research scientist. Since fall 2008,

he has been an Assistant Professor at Temple University. Dr. Ling's research interests include computer vision, medical image analysis, human computer interaction, and machine learning. He received the Best Student Paper Award at the ACM Symposium on User Interface Software and Technology (UIST) in 2003. He will be an Area Chair for CVPR 2014.