Blur-Resilient Tracking Using Group Sparsity

Pengpeng Liang[1], Yi Wu[1,2], Xue Mei[3], Jingyi Yu[4], Erik Blasch[5], Danil Prokhorov[3], Chunyuan Liao[6], Haitao Lang[1,7], and Haibin Ling[1]

[1]Department of Computer and Information Sciences, Temple University, Philadelphia, USA
[2]Jiangsu Key Laboratory of Big Data Analysis Technology, Nanjing University of Information Science and Technology, Nanjing, China
[3]Toyota Research Institute, North America, Ann Arbor, USA
[4]Department of Computer and Information Sciences, University of Delaware, Newark, USA
[5]Air Force Research Lab, Rome, NY USA
[6]HiScene Information Technologies, Shanghai, China
[7]Department of Physics and Electronics, Beijing University of Chemical Technology, Beijing, China

Abstract. In this paper, a Blur Resilient target Tracking algorithm (BReT) is developed by modeling target appearance with a groupwise sparse approximation over a template set. Since blur templates of different directions are added to the template set to accommodate motion blur, there is a natural group structure among the templates. In order to enforce the solution of the sparse approximation problem to have group structure, we employ the mixed $\ell_1 + \ell_1/\ell_2$ norm to regularize the model coefficients. Having observed the similarity of gradient distributions in the blur templates of the same direction, we further boost the tracking robustness by including gradient histograms in the appearance model. Then, we use an accelerated proximal gradient scheme to develop an efficient algorithm for the non-smooth optimization resulted from the representation. After that, blur estimation is performed by investigating the energy of the coefficients, and when the estimated target can be well approximated by the normal templates, we dynamically update the template set to reduce the drifting problem. Experimental results show that the proposed BReT algorithm outperforms state-of-the-art trackers on blurred sequences.

1 Introduction

Visual object tracking is an important topic in computer vision and it has many applications, such as automatic surveillance, human computer interaction, vehicle navigation, etc. Designing a useful real-world visual tracking algorithm is very challenging, and tremendous efforts have been made toward handling issues such as illumination changes [1], occlusions [2, 3], background clutter [4], and abrupt motions [5]. However, most of the current trackers do not explicitly take motion blur into account, which is pervasive in the real videos when

the targets move fast. One possible approach is to deblur the videos first before performing tracking. Nevertheless, though some fast deblurring methods such as [6, 7] have been recently developed, they are still computationally expensive. So this approach is not suitable for time sensitive visual tracking tasks. Also, the tracking performance will depend on the quality of deblurred videos directly.

Tracking through blur has been previously studied in [8–10]. In [8], blurred regions are matched with a matching score governed by a cost function in terms of region deformation parameters and two motion blur vectors, where the cost function is optimized with a gradient-based search technique. In [9], a mean-shift tracker is first used to locate the target. When the matching score is low, blur detection and estimation is performed, then mean-shift with blur template is applied. In [10], a blur driven tracker using sparse representation is proposed, which incorporates blur templates of different directions into the template space to model blur degradations. However, though the enhanced appearance space is more expressive, ambiguity also increases. For example, a target candidate that belongs to the background might be well represented by some blur templates. Also, the templates of the blur driven tracker are fixed, therefore when the appearance of the target changes significantly, the tracker is susceptible to drifting.

In this paper, we propose a robust blurred target tracking algorithm using group sparse representation under a particle filter framework with enhanced template space. Three components distinguish our work from previous ones: (1) since blur templates of different directions are added to the template space and the motion blur of the target always tends only one direction in a frame, there is a natural group structure among the templates, i.e., the blur templates of one direction belong to the same group. In order to enforce the solution of the sparse representation of a target candidate to have group structure, we adopt a structured sparsity inducing norm which is a combination of ℓ_1 norm and a sum of ℓ_2 norms over groups of variables [11]; (2) to account for the increase of ambiguity in the template space after enhancing it with blur templates, based on the observation that blur templates of the same direction have much more similar gradient histograms than blur templates in different directions, we use a combination of the reconstruction error and a sum of weighted distances between gradient histograms of a target candidate and each of the non-trivial templates as loss function. The resulting non-smooth convex optimization problem is solved using an accelerated proximal gradient method that guarantees fast convergence; and (3) in order to capture the appearance changes of the target and reduce the drifting problem, we perform blur detection by investigating the energy of the reconstruction coefficients. The template set is updated dynamically when two criteria based on the coefficients associated with templates are satisfied.

In the rest of the paper, related work is reviewed in §2. In §3, we present the proposed tracking algorithm and the approach for solving the resulting non-smooth convex optimization problem. §4 experimentally compares the proposed tracker with several state-of-the-art trackers over blurred sequences. §5 concludes the paper.

2 Related Work

Due to the extensive literature about visual tracking, we only review the typical works and those most related to ours. For a through survey of the tracking algorithms, we refer the readers to [12] or recent tracking evaluation papers [13–16]. Current tracking algorithms can be categorized as either discriminative or generative approaches. Discriminative approaches formulate tracking as a binary classification problem, the aim of which is to distinguish the target from background. Typical discriminative tracking approaches include online boosting [17], semi-online boosting [18], MIL tracking [19] and structured output tracking [20]. Generative approaches are based on appearance model, where tracking is performed by searching for the region most similar to the target model. Typical generative tracking methods include mean shift tracker [21], eigentracker [22], incremental tracker [23], and VTD tracker [24]. The appearance model is usually dynamically updated in order to adapt to the target appearance variations caused by pose and illumination changes.

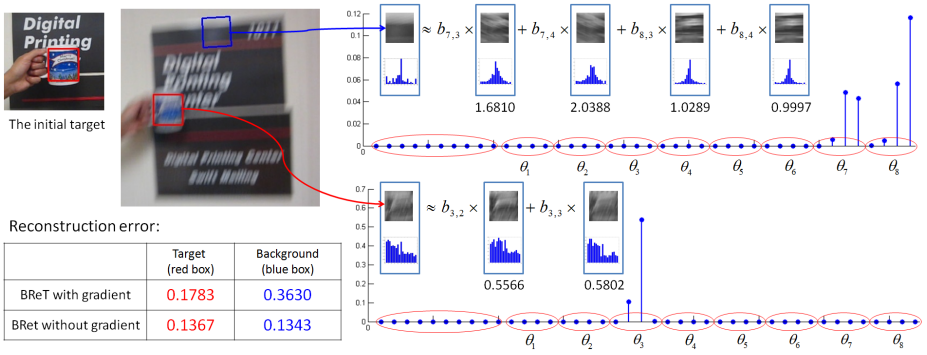
Sparse representation has been successfully applied to visual tracking in [25], and further exploited in [26–28]. In [25], the tracker represents each target candidate as a sparse linear combination of dynamically updated templates, and the tracking task is formulated as finding the candidate with the minimum reconstruction error. In [26], a real-time L1 tracker is proposed by using accelerated proximal gradient approach to solve a modified ℓ_1 norm related minimization model with ℓ_2 norm regularization of the trivial templates. In [27], dynamical group sparsity is used to explore the spatial relationship among discriminative features and temporal relationship among templates. In [28], multi-task sparse learning is adopted to mine the interdependencies among the target candidates during tracking.

Among the previous works [8–10] on blurred target tracking, the work in [10] is most related to ours in that both incorporate blur templates into the appearance space. Our method is however different in several aspects: (1) our method exploits the natural group structure among the templates by a structured sparsity regularization; (2) our method integrates gradient information in the appearance to boost the tracking robustness; and (3) we update the template set dynamically guided by blur estimation so that our method is more robust to target appearance variations.

3 Blur Resilient Tracking using Group Sparse Representation

3.1 Review of the Blur-driven Tracker (BLUT)

We briefly review the blur-driven tracker (BLUT) proposed in [10], which is the main inspiration of the proposed BReT tracker and we inherit its notations as well.



	Target (red box)	Background (blue box)
BReT with gradient	0.1783	0.3630
BRet without gradient	0.1367	0.1343

Fig. 1. Top left: The tracking results of BReT with and without gradient information, indicated by red box and blue box respectively. **Bottom left**: the reconstruction error of the two candidates measured by $0.5||\mathbf{Tc}-\mathbf{y}||_2^2$ using different tracking approaches. **Right**: The group sparse representation of the two candidates using BReT with gradient information, the L1 distance between the gradient histograms of the estimated target and each of the selected templates are also given.

Particle Filter: The particle filter [29] is a Bayesian sequential importance sampling technique for estimating the posterior distribution of state variables characterizing a dynamic system. It uses finite set of weighted samples to approximate the posterior distribution regardless of the underlying distribution. For visual tracking, we use $\mathbf{x}_t$ as the state variable to describe the location and shape of the target at time t. Given all available observations $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t\}$ up to time t, the posterior $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ is approximated by a set of N samples $\{\mathbf{x}_t\}_{i=1}^N$ with importance weights w_t^i. The optimal $\mathbf{x}_t$ is obtained by maximizing the approximate posterior probability: $\mathbf{x}_t^* = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}_{1:t})$.

Subspace Representation with Blur Templates: In order to model the blur degradations, blur templates are incorporated into the appearance space in [10]. The appearance of the tracking target $\mathbf{y} \in \mathbb{R}^d$ is represented by templates $\mathbf{T} = [\mathbf{T}_a, \mathbf{T}_b, \eta\mathbf{I}]$,

$$\mathbf{y} = [\mathbf{T}_a, \mathbf{T}_b, \eta\mathbf{I}] \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{e} \end{bmatrix} \hat{=} \mathbf{Tc}, \quad \text{s.t.} \quad \mathbf{c}_T \succeq 0, \tag{1}$$

where $\mathbf{T}_a = [\mathbf{t}_1, \cdots, \mathbf{t}_{n_a}] \in \mathbb{R}^{d \times n_a}$ contains n_a normal templates, $\mathbf{T}_b = [\mathbf{t}_{1,1}, \cdots, \mathbf{t}_{1,n_l}, \cdots, \mathbf{t}_{n_\theta,1}, \cdots, \mathbf{t}_{n_\theta,n_l}] \in \mathbb{R}^{d \times n_b}$ contains n_b blur templates, $\mathbf{I}$ is the $d \times d$ identity matrix containing the trivial templates used for modeling image corruption, η is used to control the weight of the trivial templates. Accordingly, $\mathbf{a} = (a_1, a_2, \cdots, a_{n_a})^\top \in \mathbb{R}^{n_a}$, and $\mathbf{b} \in \mathbb{R}^{n_b}$ are called *normal coefficients* and *blur coefficients* respectively, $\mathbf{e} = (e_1, e_2, \cdots, e_d)^\top$ is called *trivial coefficients*, $\mathbf{c} = [\mathbf{a}^\top, \mathbf{b}^\top, \mathbf{e}^\top]^\top$ and $\mathbf{c}_T = [\mathbf{a}^\top, \mathbf{b}^\top]^\top$.

The first normal template $\mathbf{t}_1$ is obtained from the unblurred object patch of the target in the first frame, which is usually selected manually or by detection algorithms, other templates are shifted from it. Given a blur free patch I of the

target, different blurred versions I_b of the target can be modeled as convolving I with different kernels. In our framework, $\mathbf{t}_{i,j} = \mathbf{t}_1 \otimes \mathbf{k}_{i,j}$ is the $(i,j)^{th}$ blur template, where $\mathbf{k}_{i,j}$ is a Gaussian kernel that represents a 2D motion toward direction θ_i with magnitude l_j, where $\theta_i \in \Theta = \{\theta_1, \cdots, \theta_{n_\theta}\}$, and $l_j \in \mathcal{L} = \{l_1, \cdots, l_{n_l}\}$. Consequently, we have $n_b = n_\theta \times n_l$ blur templates. Based on the directions of the blur kernels, we have $\mathbf{b} = [\mathbf{b}_1^\top, \cdots, \mathbf{b}_{n_\theta}^\top]^\top \in \mathbb{R}^{n_b}$, where $\mathbf{b}_i = (b_{i,1}, b_{i,2}, \cdots, b_{i,n_l})^\top \in \mathbb{R}^{n_l}$ are the coefficients for the blur templates toward i^{th} direction.

Blur-driven Proposal Distribution: In [10], to use the estimated motion information from the sparse representation to guide the particle sampling process, estimated motion information from different sources are integrated into the proposal distribution, which is a combination of the first-order Markov transition $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, the second-order Markov transition $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_{t-2})$, and $q_i(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1})$ based on the blur motion estimation along direction θ_i.

3.2 Loss Function with Gradient Information

Incorporating blur templates into the appearance space allows for a more expressive appearance space to model blur degradations. However, with the augmented template space, ambiguity also increases, and some background might be well represented by some blur templates, especially when only grayscale information is used, as shown in Fig.1. In order to make the tracking algorithm more robust, based on the observation that though motion blur significantly changes the statistics of the gradients of the templates, the blur templates in the same direction have much more similar gradient histograms than blur templates of different directions, we propose to use the combination of the reconstruction error and a sum of weighted distances between the target candidate and each of the non-trivial templates as loss function.

For each template of $[\mathbf{T}_a, \mathbf{T}_b]$, we calculate its gradient histogram by letting each pixel vote for an gradient histogram channel, and get $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_{n_a+n_b}] \in \mathbb{R}^{h\times(n_a+n_b)}$, where h is the number of bins of the gradient histogram; and for the target candidate, we calculate its gradient histogram $\mathbf{g} \in \mathbb{R}^h$. Since we don't consider the trivial templates when calculating the sum of weighted distances, we let $\mathbf{d} = [||\mathbf{d}_1 - \mathbf{g}||_1, ||\mathbf{d}_2 - \mathbf{g}||_1, \cdots, ||\mathbf{d}_{n_a+n_b} - \mathbf{g}||_1, 0, \cdots, 0] \in \mathbb{R}^{(n_a+n_b+d)}$ indicate the distance between $\mathbf{g}$ and the gradient histogram of each element in $\mathbf{T}$. $||\mathbf{dc}||_2^2$ is used to measure the sum of the weighted distances, and

$$\frac{1}{2}||\mathbf{Tc} - \mathbf{y}||_2^2 + \beta||\mathbf{dc}||_2^2 \tag{2}$$

is used as the loss function.

3.3 Group Sparsity via $\ell_1 + \ell_1/\ell_2$ Mixed Norm

For the augmented template set with blur templates of different directions, since the motion blur of the target is always toward only one direction at time t, there is

a natural group structure among the templates. The representation of the target candidate should not only be sparse, but also have group structure, i.e., the coefficients should also be sparse at the group level. In our tracking framework, we divide the templates into $n_g = n_\theta + d + 1$ groups $\mathcal{G} = \{G_1, G_2, \cdots, G_{n_\theta+d+1}\}$ using the following scheme: the normal templates are in one group; the blur templates in the same direction forms a group; and each trivial template is an individual group. In order to capture the group information among the templates and achieve sparsity at the same time, we employ a structured sparsity inducing norm which combines the ℓ_1 norm and a sum of ℓ_2 norms over groups of variables [11]. The mixed norm is known as "sparse group Lasso".

Combining the loss function (2) and the $\ell_1 + \ell_1/\ell_2$ mixed norm results in the following non-smooth convex optimization problem:

$$\begin{aligned} \min_{\mathbf{c}} \frac{1}{2}\|\mathbf{T}\mathbf{c} - \mathbf{y}\|_2^2 + \beta\|\mathbf{d}\mathbf{c}\|_2^2 + \lambda_1\|\mathbf{c}\|_1 + \lambda_2 \sum_{i=1}^{n_g} \|\mathbf{c}_{G_i}\|_2, \\ \text{s.t.} \quad \mathbf{c}_T \succeq 0 \end{aligned} \tag{3}$$

where $\mathbf{c}_{G_i}$ are coefficients associated with G_i.

Once (3) is solved, the observation likelihood can be derived from the reconstruction error of $\mathbf{y}$ as $p(\mathbf{y}_t|\mathbf{x}_t) \propto \exp\{-\alpha||\mathbf{T}\mathbf{c} - \mathbf{y}||_2^2\}$, where α is a constant used to control the shape of the Gaussian kernel.

3.4 Solve (3) by Accelerated Proximal Gradient

To solve the non-smooth convex optimization problem in Eq.(3), we adopt the accelerated proximal gradient method FISTA [30] which has convergence rate of $O(\frac{1}{k^2})$, where k is the number of iterations. FISTA is designed for solving the following unconstrained optimization problem:

$$\min_{\mathbf{z}} F(\mathbf{z}) = f(\mathbf{z}) + g(\mathbf{z}) \tag{4}$$

where f is a smooth convex function with Lipschitz continuous gradient, and g is a continuous convex function which is possibly non-smooth.

In order to solve Eq.(3) with FISTA, we let $\mathbf{z} = [z_1, z_2, \cdots, z_{n_a+n_b+d}]^\top$ and make the substitution $\mathbf{c} = [z_1^2, z_2^2, \cdots, z_{n_a+n_b}^2, z_{n_a+n_b+1}, \cdots, z_{n_a+n_b+d}]^\top$ to incorporate the explicit constraint in Eq.(3) into the objective function, and solve the following optimization problem:

$$\min_{\mathbf{z}} \frac{1}{2}\|\mathbf{T}\mathbf{c} - \mathbf{y}\|_2^2 + \beta\|\mathbf{d}\mathbf{c}\|_2^2 + \lambda_1\|\mathbf{z}\|_1 + \lambda_2 \sum_{i=1}^{n_g} \|\mathbf{z}_{G_i}\|_2 \tag{5}$$

where $\mathbf{z}_{G_i}$ is associated with group G_i. Then, Eq.(5) can be re-expressed as Eq.(4), where

$$f(\mathbf{z}) = \frac{1}{2}||\mathbf{T}\mathbf{c} - \mathbf{y}||_2^2 + \beta||\mathbf{d}\mathbf{c}||_2^2 \tag{6}$$

and

$$g(\mathbf{z}) = \lambda_1||\mathbf{z}||_1 + \lambda_2 \sum_{i=1}^{n_g} ||\mathbf{z}_{G_i}||_2 \tag{7}$$

To develop a proximal gradient method, the following quadratic approximation of $F(\mathbf{z})$ at a given point $\mathbf{z}^{(k)}$is considered, for $L > 0$

$$\begin{aligned} Q_L(\mathbf{z}, \mathbf{z}^{(k)}) =& f(\mathbf{z}^{(k)}) + \langle \mathbf{z} - \mathbf{z}^{(k)}, \nabla f(\mathbf{z}^{(k)}) \rangle \\ &+ \frac{L}{2} \|\mathbf{z} - \mathbf{z}^{(k)}\|_2^2 + g(\mathbf{z}) \end{aligned} \tag{8}$$

where $\nabla f(\mathbf{z}^{(k)})$ is the gradient function of $f(\cdot)$ at point $\mathbf{z}^{(k)}$.

Lemma 1 [30] *Let f be a continuously differentiable function with Lipschitz continuous gradient and Lipschitz constant $L(f)$. Then, for any $L \geq L(f)$,*

$$F(\mathbf{z}) \leq Q_L(\mathbf{z}, \mathbf{z}^{(k)})$$

According to Lemma 1, given $L \geq L(f)$, a unique solution of $F(\mathbf{z})$ can be obtained by minimizing $Q_L(\mathbf{z}, \mathbf{z}^{(k)})$,

$$p_L(\mathbf{z}^{(k)}) = \arg\min_{\mathbf{z}} \frac{1}{2} ||\mathbf{z} - \hat{\mathbf{z}}|| + \frac{1}{L} g(\mathbf{z}) \tag{9}$$

where $\hat{\mathbf{z}} = \mathbf{z}^{(k)} - \frac{1}{L} \nabla f(\mathbf{z}^{(k)})$, and

$$\nabla f(\mathbf{z}) = \text{diag}(\mathbf{w})(\mathbf{T}^\top \mathbf{T}\mathbf{c} - \mathbf{T}^\top \mathbf{y}) + 2\beta \text{diag}(\mathbf{w})\mathbf{d}^\top \mathbf{d}\mathbf{c} \tag{10}$$

where $\mathbf{w} = [2z_1, 2z_2, \cdots, 2z_{n_a+n_b}, 1, \cdots, 1] \in \mathbb{R}^{n_a+n_b+d}$. Eq.(9) just ignores the constant $z^{(k)}$ in Eq.(8), and the minimum of Eq.(8) can be obtained by solving Eq.(9). Alg. 1 describes FISTA with backtracking.

Algorithm 1 FISTA with backtracking [30]

Input: $L_0 > 0$, $\tau > 1$, $\mathbf{v}^{(1)} = \mathbf{z}^{(0)}$, $t_1 = 1$

1: **for** k=1,2,..., iterate until convergence **do**
2: set $L = L_{k-1}$
3: **while** $F(p_L(\mathbf{v}^{(k)})) > Q_L(p_L(\mathbf{v}^{(k)}), \mathbf{v}^{(k)})$ **do**
4: $L = \tau L$
5: **end while**
6: set $L_k = L$ and update
7: $\mathbf{z}^{(k)} = p_{L_k}(\mathbf{v}^{(k)})$,
8: $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$,
9: $\mathbf{v}^{(k+1)} = \mathbf{z}^{(k)} + (\frac{t_k - 1}{t_{k+1}})(\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)})$
10: **end for**

A critical step of Alg. 1 is to solve Eq.(9) efficiently. Since the $\ell_1 + \ell_1/\ell_2$-norm is a special case of the tree structured group Lasso, Eq.(9) can be converted to

$$p_L(\mathbf{z}^{(k)}) = \arg\min_{\mathbf{z}} \frac{1}{2} ||\mathbf{z} - \hat{\mathbf{z}}|| + \sum_{i=0}^{m} \sum_{j=1}^{n_i} w_j^i ||\mathbf{z}_{G_j^i}||_2 \tag{11}$$

where m is the depth of the index tree defined in [31], n_i is the number of groups at depth i, $w_j^i \geq 0 (i = 0, 1, \cdots, m, j = 1, 2, \cdots, n_i)$ is the pre-defined weight for group G_j^i. We apply the $\text{MY}_{\text{tgLasso}}$ algorithm [31] to solve Eq.(11) efficiently. $\text{MY}_{\text{tgLasso}}$ algorithm maintains a working variable $\mathbf{u}$ initialized with $\hat{\mathbf{z}}$, then it traverses the index tree in the reverse breadth-first order to update $\mathbf{u}$ with $\mathbf{u}_{G_j^i}^i = \mathbf{u}_{G_j^i}^{i+1} \max\left(0, 1 - w_j^i / \|\mathbf{u}_{G_j^i}^{i+1}\|\right)$.

The time complexity of $\text{MY}_{\text{tgLasso}}$ algorithm is $O(mn)$, where n is the dimension of $\mathbf{z}$. After converting Eq.(9) to Eq.(11), the index tree has a constant depth 2, so the time complexity for solving Eq.(9) is $O(n)$, where $n = n_a + n_b + d$.

3.5 Template Update with Blur Detection

In order to capture the appearance variations of the target caused by illumination or pose changes, the template set needs to be updated during tracking. Since the appearance of the target is corrupted when heavy blur appears, updating the template set with heavily blurred target cannot capture the appearance changes of the target. So we propose to perform blur detection of the tracking result before updating the template set.

To detect blur, we investigate the response of both normal coefficients and blur coefficients obtained from solving the optimization problem Eq.(3). If the target is not blurred, the energy of the normal coefficients will be dominant. One criterion for updating the template set is $\frac{E(\mathbf{a})}{E(\mathbf{a})+E(\mathbf{b})} > 0.9$, where $E(\cdot)$ represents the energy which is the sum of the absolute value of the corresponding coefficients. Also, trivial templates are activated when the target cannot be well approximated by the template set. In order to avoid contaminating the template set, another criterion for updating template set is $\frac{E(\mathbf{e})}{E(\mathbf{a})+E(\mathbf{b})+E(\mathbf{e})} < 0.1$. When the target is not similar to any of the normal templates, and both of the above two criteria are satisfied, we replace the normal template having lowest response with the target template.

Algorithm 2 BReT: Blur resilient tracker

Input: Current frame F_t, sample set $\mathbf{S}_{t-1}$, template set $\mathbf{T}_{t-1}$

1: **for** $i = 1$ to N **do**
2: Draw N particles $\mathbf{x}_t^i$ with the blur driven proposal distribution
3: Obtain the candidate patch $\mathbf{y}_t^i$ of $\mathbf{x}_t^i$
4: Solving the optimization problem (3)
5: Calculate the observation likelihood $p(\mathbf{y}_t^i | \mathbf{x}_t^i)$
6: **end for**
7: Locate the target $\mathbf{x}_t^*$ with the maximum observation likelihood
8: Estimate motion from blur via the blur coefficients of the estimated target
9: Update the template set $\mathbf{T}_t$ with blur detection
10: Update the sample set $\mathbf{S}_t$ by resampling with p

4 Experiments

We implemented the proposed algorithm with MATLAB R2011b, and the SLEP package [32] is used to solve Eq.(11). In our tracking framework, the state variable $\mathbf{x}_t$ is modeled by four parameters $\mathbf{x}_t = (t_x, t_y, s, \theta)$, where (t_x, t_y) are the 2D translation parameters, s is the scale variation parameter, and θ is the rotation variation parameter. In our experiment, $n_a = 10$ normal templates are used, and we set $n_\theta = 8$, $n_l = 4$, so there are $10 + 8 \times 4 = 42$ non-trivial templates in total. For the optimization problem Eq.(3), we set $\beta = 0.35$, $\lambda_1 = 0.03$, $\lambda_2 = 0.03$; for the number of bins of the gradient histogram, we set $h = 19$; and for the weight of trivial templates, we set $\eta = 0.15$. These parameters are kept the same for all the sequences.

To evaluate the performance of the blurred target tracking algorithm, we compile a set of 12 challenging blurred tracking sequences, denoted as *owl, face, body, car1, car2, car3, car4, jumping, running, cola, dollar* and *cup*. The sequences *owl, face, body, car1, car2, car3* and *car4* were used in [10] and can be downloaded from an online source including the ground truth.[1] The sequence *jumping* and the associated ground truth can be downloaded from an online source.[2] For the sequences *running, cola, dollar* and *cup*, we collected them ourselves and labeled the ground truth manually, and these four sequences contain 2400 frames. In total we use 6235 frames for the experiments.

We compared the proposed BReT algorithm with seven state-of-the-art visual trackers: VTD [24], L1APG [26], IVT [23], MIL [19], OAB [17], Struck [20] and BLUT [10]. We use the publicly available source codes or binaries from the referenced authors with the same initialization of the target in the first frame. We first used the default parameter settings of the above trackers to evaluate them. Nevertheless, all but the BLUT [10], which is specifically designed for blurred target tracking, failed on most of the sequences. One critical reason for the failure of these trackers is that the search radius is not large enough or the variance of the motion model under the particle filter framework is not large enough to cover the fast motion in these blurred sequences. So we tuned each of these trackers specifically. We increased the search radius of Struck from 30 to 100, and the search radius of MIL and OAB from 25 to 100. For L1APG, VTD, IVT and BReT which use the particle filter technique, we set the number of particles to $N = 600$. For L1APG and IVT, we set the variance of the motion model the same as BReT. For VTD, since the binary code has a predefined value for the variance of the motion model and only allows the user to specify how many times to enlarge or shrink this value, we set the variance ten times as large as the predefined one. For BLUT, the parameter setting provided in the original code works better, so we kept it unchanged. Among all these trackers, only VTD uses color information, while the others only use grayscale information.

[1] http://www.dabi.temple.edu/ hbling/data/TUblur.zip
[2] http://cvlab.hanyang.ac.kr/tracker_benchmark/seq/Jumping.zip

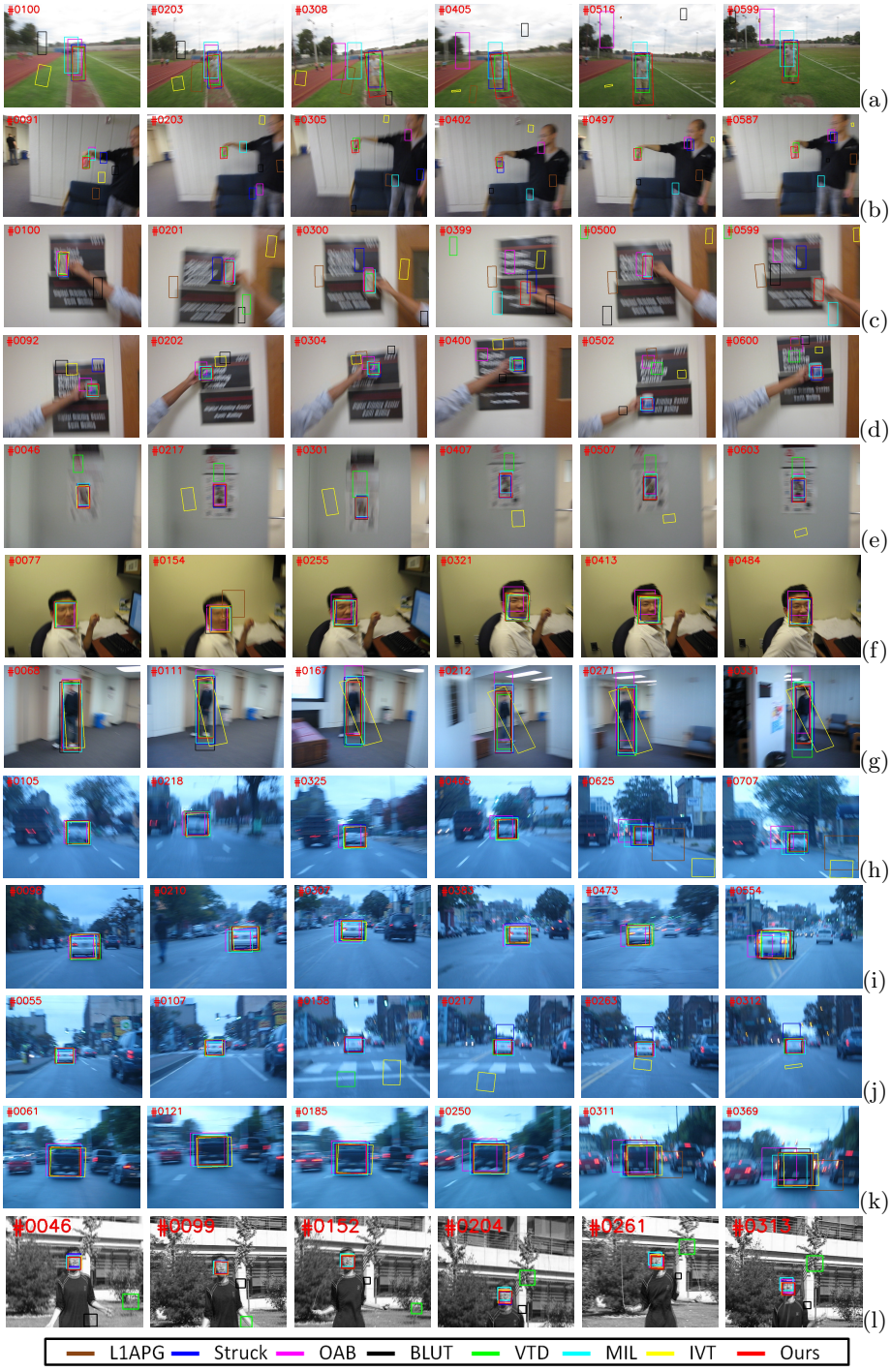


Fig. 2. Tracking results of eight algorithms on 12 sequences. The name of the sequences are (a) running, (b) cola, (c) dollar, (d) cup, (e) owl, (f) face, (g) body, (h) car1, (i) car2, (j) car3, (k) car4 and (l) jumping.

4.1 Qualitative Evaluation

The sequence *running* was captured outdoors, a person is running on the playground while being videotaped by another person who is also running. Some tracking results are shown in Fig. 2(a). Both our tracker and VTD track the target well. Without using color information, the target is very similar to the grass, especially when blur appears. We believe that the success of VTD on this sequence can be attributed to the use of color information.

In the sequence *cola*, a moving cola can was tracked, as shown in Fig. 2(b). When blur appears, it is hard to distinguish the target from the sofa and the clothes of the person with only grayscale information. The appearance of the target also changed slightly by rotation. Our tracker and VTD perform well on this sequence, while the other trackers start to fail when heavy blur appears.

Results on the sequence *dollar* are shown in Fig. 2(c), in which a paper dollar was tracked. Our tracker is capable of tracking the target for most frames of the sequence. MIL also exhibits comparable results, but starts to lose the target completely at frame 505.

Fig. 2(d) demonstrates the results of the *cup* sequence. Our tracker, Struck and MIL can track the target for almost all frames. Other trackers often locate the area near the edge between the indicator board and the wall as target. A possible reason is that there is a white stripe containing characters in the top of cup surface, and the rest of the surface of the cup is blue in general.

In the sequence *owl*, a plane object was tracked as shown in Fig. 2(e). VTD starts to fail at frame 46 and locates a sign that has similar color as the target. IVT starts to fail at frame 183. Our tracker, BLUT, Struck, L1APG can track the target accurately through the whole sequence. MIL and OAB also obtain comparable results on this sequence.

Results on the sequence *face* are shown in Fig. 2(f), in which the target is not only blurred but also has slight pose variation. All the trackers except OAB perform well on this sequence, while OAB meets problems from frame 68.

Results on the sequence *body* are given in Fig. 2(g). A person is walking in this sequence, and most of the frames are severely blurred. IVT cannot correctly obtain the direction of the target. Though OAB does not completely lose the target through the sequence, for most of the sequence, it cannot get the accurate position of the target. Other trackers works well on this sequence and get comparable results.

The sequences *car1,car2, car3, car4* are captured from outdoor traffic scenes. The results for *car*1 are shown in Fig. 2(h). OAB starts to drift at frame 100, MIL, IVT and L1APG start to drift at frame 533 when very heavy blur appears. Our tracker, BLUT, Struck and VTD can track the target successfully for most of the frames. Fig. 2(i) shows the results on *car2*, MIL and OAB perform poorly, and start to drift at frame 107 and frame 77 respectively. Other trackers work well on this sequence, and our tracker, BLUT, L1APG, VTD, IVT perform a little better than Struck. Fig. 2(j) gives the results on *car3*. IVT lost the target at frame 121, Struck lost the target at frame 186 and the sky was tracked as the target by Struck. Results of sequence *car4* are given in Fig. 2(k), where OAB

starts to locate the target inaccurately from frame 34. Our tracker, VTD and Struck get excellent results on this sequence.

Fig. 2(l) shows the results of *jumping*, in which the target was jumping and causing obvious motion blur. Compared with other sequences, the motion in *jumping* is relatively simple, mainly includes up and down motion. All the trackers except BLUT and VTD perform well on this sequence.

4.2 Quantitative Evaluation

Table 1. Success rate of the trackers (median). The best, second and third bests are in red, blue and green respectively.

Video	MIL	OAB	VTD	IVT	Struck	L1APG	BLUT	Ours
running	0.03	0.05	0.88	0.07	0.45	0.16	0.07	0.85
cola	0.10	0.07	0.73	0.06	0.14	0.04	0.06	0.81
dollar	0.78	0.42	0.43	0.33	0.17	0.17	0.09	0.90
cup	0.95	0.11	0.65	0.11	0.97	0.42	0.11	0.99
owl	0.71	0.68	0.07	0.28	0.99	0.98	1.00	1.00
face	0.76	0.14	1.00	0.95	1.00	0.96	0.98	0.99
body	0.59	0.09	0.56	0.20	0.88	0.65	0.67	0.77
car1	0.72	0.15	0.99	0.70	0.99	0.69	0.99	0.99
car2	0.41	0.12	0.94	0.96	0.88	0.98	0.97	0.99
car3	0.74	0.93	0.89	0.33	0.54	0.99	0.94	1.00
car4	0.68	0.11	1.00	0.75	0.99	0.71	0.69	0.97
jumping	0.92	1.00	0.11	1.00	1.00	1.00	0.03	0.99
median	0.72	0.13	0.81	0.33	0.93	0.70	0.68	0.99

We use two criteria to evaluate these trackers quantitatively. We first use the percentage of frames for which the estimated target location is within a threshold distance from the ground truth to measure the success rate of each tracker, and we use 15 pixels as the threshold. Since all these trackers involve randomness, we run each tracker five times on all the sequences, and report both the median result and the average result with standard deviation. Table 1 summarizes the results using median, and the results of mean with standard deviation are given in Table 2. We also plot the center location error of each tracker on all the sequences over time according to the median results, as shown in Fig. 3. From the results, we can see that our tracking algorithm works favorably against state-of-the-art methods. The good performance can be attributed to the use of group sparse representation to approximate the target candidate and incorporating gradient information to the loss function to reduce the ambiguity of the enhanced appearance space.

Table 2. Success rate of the trackers (mean±standard deviation). The best, second and third bests are in red, blue and green respectively.

Video	MIL [19]	OAB [17]	VTD [24]	IVT [23]	Struck [20]	L1APG [26]	BLUT [10]	Ours
running	0.06±0.05	0.07±0.04	0.91±0.04	0.07±0.02	0.45±0.00	0.17±0.08	0.07±0.01	0.70±0.28
cola	0.12±0.04	0.09±0.04	0.72±0.06	0.05±0.02	0.15±0.06	0.04±0.02	0.07±0.03	0.78±0.12
dollar	0.74±0.13	0.33±0.13	0.47±0.07	0.33±0.00	0.30±0.32	0.29±0.27	0.10±0.04	0.73±0.32
cup	0.82±0.24	0.27±0.37	0.68±0.09	0.20±0.21	0.80±0.33	0.42±0.19	0.11±0.01	0.99±0.00
owl	0.77±0.18	0.64±0.34	0.07±0.00	0.27±0.03	0.98±0.00	0.91±0.17	1.00±0.00	1.00±0.00
face	0.79±0.06	0.13±0.06	1.00±0.00	0.95±0.03	1.00±0.00	0.77±0.29	0.98±0.01	0.99±0.00
body	0.53±0.21	0.09±0.03	0.55±0.10	0.17±0.09	0.88±0.01	0.60±0.20	0.68±0.03	0.77±0.03
car1	0.71±0.19	0.18±0.06	0.99±0.01	0.80±0.15	0.99±0.00	0.69±0.00	0.99±0.00	0.99±0.00
car2	0.41±0.15	0.11±0.06	0.78±0.36	0.93±0.08	0.88±0.00	0.98±0.01	0.97±0.01	0.99±0.00
car3	0.71±0.21	0.80±0.29	0.91±0.08	0.60±0.37	0.53±0.01	0.86±0.30	0.95±0.05	1.00±0.00
car4	0.74±0.13	0.16±0.12	0.95±0.11	0.75±0.03	0.99±0.00	0.74±0.09	0.67±0.05	0.97±0.02
jumping	0.89±0.15	1.00±0.00	0.18±0.10	0.92±0.18	1.00±0.00	1.00±0.00	0.03±0.02	0.98±0.03
mean	0.61±0.27	0.32±0.31	0.68±0.31	0.50±0.36	0.75±0.31	0.62±0.32	0.55±0.43	0.91±0.12

5 Conclusion

We propose a robust blurred target tracking algorithm using group sparse representation. The proposed algorithm exploits the natural group structure among templates by employing a $\ell_1 + \ell_1/\ell_2$ mixed norm. We also incorporate the appearance information from gradient histograms in a loss function that helps reduce the ambiguity of the enhanced appearance space. An accelerated proximal gradient approach is adopted to solve the resulting non-smooth convex optimization problem. In addition, we dynamically update the template set with blur detection to make the tracker more robust to appearance variations of the target. Finally, we compared our tracker with seven state-of-the-art trackers on 12 blurred sequences to demonstrate the effectiveness and robustness of the proposed tracker.

Acknowledgement. This work was supported in part by the US NSF Grants IIS-1218156 and IIS-1350521. Wu was supported in part by NSFC under Grants 61005027 and 61370036, and Lang was supported by "Beijing Higher Education Young Elite Teacher Project" (No.YETP0514).

References

1. Silveira, G.F., Malis, E.: Real-time visual tracking under arbitrary illumination changes. In: CVPR. (2007)
2. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: CVPR. (2006)
3. Jia, X., Lu, H., Yang, M.H.: Visual tracking via adaptive structural local sparse appearance model. In: CVPR. (2012)
4. Hu, W., Li, X., Zhang, X., Shi, X., Maybank, S.J., Zhang, Z.: Incremental tensor subspace learning and its applications toforeground segmentation and tracking. IJCV **91** (2011) 303–327
5. Kwon, J., Lee, K.M.: Wang-landau monte carlo-based tracking methods for abrupt motions. PAMI **35** (2013) 1011–1024

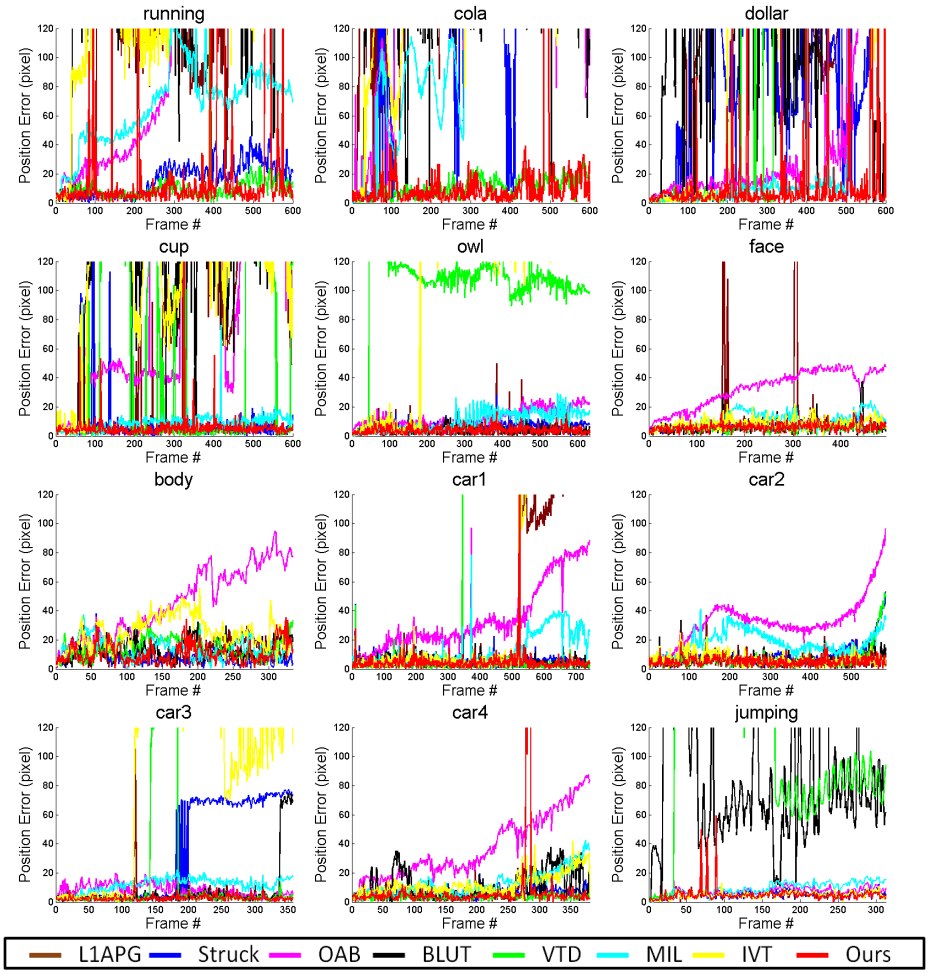


Fig. 3. Quantitative comparison of the trackers in terms of position errors (in pixel).

6. Cho, S., Lee, S.: Fast motion deblurring. ACM Trans. Graph. **28** (2009)
7. Xu, L., Zheng, S., Jia, J.: Unnatural l0 sparse representation for natural image deblurring. In: CVPR. (2013)
8. Jin, H., Favaro, P., Cipolla, R.: Visual tracking in the presence of motion blur. In: CVPR. (2005)
9. Dai, S., Yang, M., Wu, Y., Katsaggelos, A.K.: Tracking motion-blurred targets in video. In: ICIP. (2006)
10. Wu, Y., Ling, H., Yu, J., Li, F., Mei, X., Cheng, E.: Blurred target tracking by blur-driven tracker. In: ICCV. (2011)
11. Bach, F., Jenatton, R., Mairal, J., Obozinski, G.: Convex optimization with sparsity-inducing norms. Optimization for Machine Learning (2011) 19–53
12. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. Acm Computing Surveys **38** (2006) 13

13. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. Proc. IEEE Conf. Computer Vision and Pattern Recognition (2013) 2411–2418
14. Pang, Y., Ling, H.: Finding the best from the second bests-inhibiting subjective bias in evaluation of visual tracking algorithms. Proc. IEEE Int'l Conf. Computer Vision (2013) 2784–2791
15. Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Porikli, F., Khajenezhad, A., Salahledin, A., Soltani-Farani, A., Zarezade, A., Petrosino, A., et al.: The visual object tracking vot2013 challenge results. IEEE Workshop on visual object tracking challenge (2013)
16. Smeulders, A.W.M., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: an experimental survey. IEEE Trans. Pattern Anal. Mach. Intell. **36** (2014)
17. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via on-line boosting. In: BMVC. (2006)
18. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: ECCV. (2008)
19. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. PAMI **33** (2011) 1619–1632
20. Hare, S., Saffari, A., Torr, P.H.S.: Struck: Structured output tracking with kernels. In: ICCV. (2011)
21. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. PAMI **25** (2003) 564–577
22. Black, M.J., Jepson, A.D.: Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. IJCV **26** (1998) 63–84
23. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. IJCV **77** (2008) 125–141
24. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: CVPR. (2010)
25. Mei, X., Ling, H.: Robust visual tracking and vehicle classification via sparse representation. PAMI **33** (2011) 2259–2272
26. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust l1 tracker using accelerated proximal gradient approach. In: CVPR. (2012)
27. Liu, B., Yang, L., Huang, J., Meer, P., Gong, L., Kulikowski, C.: Robust and fast collaborative tracking with two stage sparse optimization. In: ECCV. (2010)
28. Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Robust visual tracking via structured multi-task sparse learning. IJCV **101** (2013) 367–383
29. Doucet, A., De Freitas, N., Gordon, N., et al.: Sequential Monte Carlo methods in practice. Volume 1. Springer New York (2001)
30. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. on Imaging Sciences **2** (2009) 183–202
31. Liu, J., Ye, J.: Moreau-yosida regularization for grouped tree structure learning. In: NIPS. (2010)
32. Liu, J., Ji, S., Ye, J.: SLEP: Sparse Learning with Efficient Projections. Arizona State University. (2009)