

# Blurred Target Tracking by Blur-driven Tracker\*

Yi Wu<sup>†b</sup> Haibin Ling<sup>†</sup> Jingyi Yu<sup>‡</sup> Feng Li<sup>‡</sup> Xue Mei Erkang Cheng<sup>†</sup>

<sup>†</sup>Computer & Information Science Department, Temple University, Philadelphia, PA 19122, USA

<sup>b</sup>School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, 210044, China

<sup>‡</sup>Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA

{wuyi, hbling, tuc33610}@temple.edu, {feli, yu}@cis.udel.edu, nathanmei@gmail.com

## Abstract

Visual tracking plays an important role in many computer vision tasks. A common assumption in previous methods is that the video frames are blur free. In reality, motion blurs are pervasive in the real videos. In this paper we present a novel *BLUR-driven Tracker (BLUT)* framework for tracking motion-blurred targets. *BLUT* actively uses the information from blurs without performing deblurring. Specifically, we integrate the tracking problem with the motion-from-blur problem under a unified sparse approximation framework. We further use the motion information inferred by blurs to guide the sampling process in the particle filter based tracking. To evaluate our method, we have collected a large number of video sequences with significant motion blurs and compared *BLUT* with state-of-the-art trackers. Experimental results show that, while many previous methods are sensitive to motion blurs, *BLUT* can robustly and reliably track severely blurred targets.

## 1. Introduction

Visual tracking plays an important role in surveillance, robotics, human computer interaction, and medical imaging [30]. Tremendous efforts have been focused on robustly handling issues such as noise [23], illumination [2], occlusions [20] and background clutter [10]. A common assumption in these algorithms is that the video/images are blur free. In reality, motion blurs are pervasive in the real videos due to the low speed of the camera and the fast motions of the target, and they confound visual tracking tasks by destroying both critical features of the target and the observation model in existing approaches. For example, many traditional tracking algorithms will easily fail due to high residuals between the image model and the measurements (Fig. 5).

Tracking blurred target is hard due to several challenges:

(1) The degradation in appearance frequently brings troubles to the target inference; (2) The accompanied abrupt motion brings large uncertainty to the estimation of target position; and (3) The degree of blur itself can vary significantly over frames, ranging from blur-free to drastic blur. A natural solution is to first deblur the contents and then apply tracking. In image processing, a large number of robust deconvolution methods have been developed, from the earlier approaches based on regularization [24] to the latest ones using image statistics [8, 14], edge priors [15], and sparse representation [17, 3]. However, since motion blurs in video resemble a box filter that destroys high-frequency features, results generated by these deconvolution methods often contain strong ringing artifacts, creating harmful “fake” features that further complicate the tracking process. Furthermore, most deblurring algorithms are computationally expensive and therefore not suitable for time sensitive visual tracking tasks. In addition, dealing simultaneous different degrees of blur effects is not a trivial problem.

Two notable previous studies on tracking through blur are the work of [12, 4] and the motion-from-blur techniques [5, 6]. In [12], the blurred regions are matched by computing the matching score in terms of the region deformation parameters and two motion vectors, and then a local gradient descent technique is employed. The assumption there is that the blurred target appears highly coherent in the video sequence and the motion between frames are small. In [4], mean-shift tracker with motion-blurred templates is adopted for motion-blurred target tracking. Dai and Wu [5] treated motion blurs as an alpha matte for estimating the point spread function (PSF). Ding et al. [6] used frequency image statistics as constraints for recovering the motion parameters. These methods assume that the blur regions have been roughly segmented whereas our goal is to automatically track these regions over time.

In this paper we present a novel visual tracking technique called the *BLUR-driven Tracker (BLUT)* for tracking motion-blurred targets. *BLUT* is based on the observation that although motion blurs degrade the visual features of the

\*This work was done when Wu was a postdoctor at Temple University.

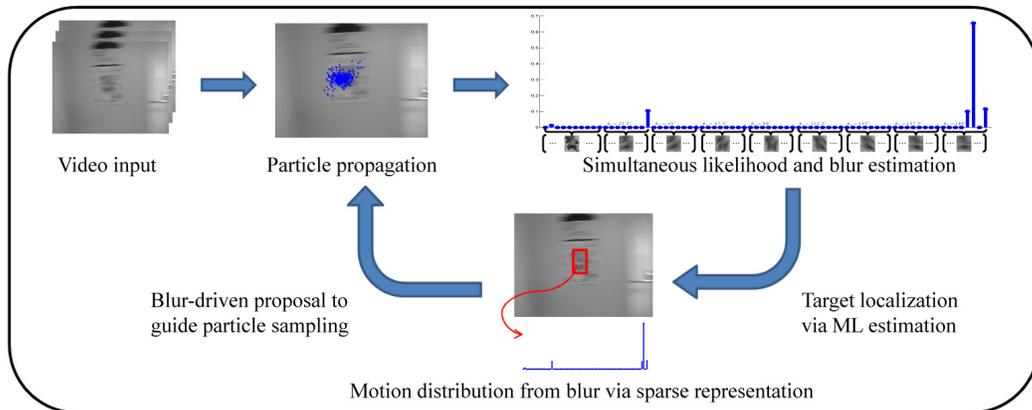


Figure 1. Overview of the proposed BLUT tracker.

target, they, at the same time, provide useful cues about the movements to help tracking. Our solution, therefore, aims to combine blur insensitivity and blur estimation in a unified framework for visual tracking.

Our method follows the “analysis by synthesis” philosophy by incorporating the *blur templates* into the appearance space for modeling blur degradations. We attack the three blur-related challenges in a unified fashion. First, we select the target candidates from the blur-enhanced appearance space by solving an  $\ell_1$ -regularized least squares problem. The candidate with the smallest projection error corresponds to the tracking target. This significantly improves the robustness of target inference. Second, the above sparse reconstruction provides a natural way to estimation the distribution of the blur motion at the current frame. This estimation is then used to guide sample propagation in a particle filter framework over time. Third, the variation of blur is naturally addressed by encoding different degrees of blur in the template set.

To evaluate our method, we have collected a set of video sequences with significant motion blurs<sup>1</sup>. We tested the proposed approach on these sequences and observed promising tracking performances in comparison with several state-of-the-art trackers.

It is worth noting that our study shares similar philosophy with recent studies of deblurring in using sparse representation ([17, 3], etc). Being an ill-posed problem, deblurring by itself is an open problem. Our motivation, instead, is to use the representation for target inference and motion distribution estimation, without explicitly deblurring the input video. In other words, our blur-driven tracker bypasses the difficult (and unnecessary) deblurring procedure, but integrates the blur information seamlessly into the visual tracking process. The effectiveness of this scheme is clearly demonstrated in the experiments. The work in [4]

is the most related work to ours in that both use blur templates for tracking. In comparison, our method is different in several aspects: (1) our method does not distinguish local and global blurs and therefore can handle blurs caused by both target and camera motions; (2) Our method estimate blur effects directly through target inference and therefore requires no off-line training; and (3) We use particle filter framework while mean-shift is used in [4].

**Overview.** Similar to many tracking algorithms, our proposed tracker has two closely related components: object representation (§2) and sequential state inference (§3). For object representation, we introduce blur templates into the standard template set to build an enriched template subspace. Then, we model a target with a sparse approximation using these templates as inspired by recent work on sparse visual tracking [19]. This representation not only improves the tracking accuracy against motion blur, but also estimates the distribution of the target motion. For sequential state estimation, we use the particle filter framework [11] to guide the tracking process. The estimated motion distribution is then integrated into the framework and acts as a guide for the particle propagation. Fig. 1 illustrates the processing pipeline of our blur-driven visual tracker. An outline of our tracking algorithm is shown in Algorithm 1.

## 2. Object Representation and Blur Estimation using Blur Templates

### 2.1. Subspace Representation

Inspired by recent work on sparse representation for visual tracking [19], we use a template subspace representation to model the appearance of tracking target. To handle the blur effects in the target’s appearance, we introduce blur templates to augment the template set. This expanded set is then used to span the subspace. We follow notations in [19] whenever applicable.

<sup>1</sup><http://www.dabi.temple.edu/hbling/data/TUblur.zip>

Let  $\mathbf{y} \in \mathbb{R}^d$  (we concatenate pixel intensities into a vector) be the appearance of a tracking target. It is approximated by using a low dimensional subspace spanned by a set of target templates  $[\mathbf{T}_a, \mathbf{T}_b]$ ,

$$\mathbf{y} \approx [\mathbf{T}_a, \mathbf{T}_b] \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \quad (1)$$

where  $\mathbf{T}_a = [\mathbf{t}_1, \dots, \mathbf{t}_{n_a}] \in \mathbb{R}^{d \times n_a}$  containing  $n_a$  *normal templates* and  $\mathbf{T}_b \in \mathbb{R}^{d \times n_b}$  containing  $n_b$  *blur templates* which will be described later; accordingly,  $\mathbf{a} = (a_1, a_2, \dots, a_{n_a})^\top \in \mathbb{R}^{n_a}$  and  $\mathbf{b} \in \mathbb{R}^{n_b}$  are approximation coefficients named *normal coefficients* and *blur coefficients* respectively.

The normal templates  $\mathbf{T}_a$  are obtained from unblurred object patches, which are usually selected manually or by detection algorithms in the first frame. The blurred templates  $\mathbf{T}_b$  are automatically generated from normal templates to take into account different blur effects. This representation is illustrated in Fig. 2.

## 2.2. Blur Template

Let  $I$  be the blur-free (latent) image of a tracking target. A blurred version  $I_b$  of the target can be modeled as convolving  $I$  with a Gaussian kernel  $\mathbf{k}_v$  representing a 2D motion, i.e.:

$$I_b(\mathbf{p}) = \mathbf{k}_v \otimes I(\mathbf{p}), \quad (2)$$

where vector  $\mathbf{v}$  encodes both the direction and the magnitude of the motion. Since the kernel  $\mathbf{k}_v$  is symmetric, the motion blur kernel  $\mathbf{k}_v$  is therefore equivalent to  $\mathbf{k}_{-\mathbf{v}}$ . Equation (2) essentially interprets motion blurs at a pixel  $\mathbf{p}$  as an integration over  $\mathbf{p}$ 's neighboring pixels.

To capture different blur effects, we consider different potential motion blurs governed by the parameter pair  $\theta$  and  $l$ , such that  $\theta$  is used for the motion direction and  $l$  for speed. In our implementation, we sampled on  $n_\theta = 8$  different directions  $\Theta = \{\theta_1, \dots, \theta_{n_\theta}\}$  and  $n_l = 8$  different speeds  $\mathcal{L} = \{l_1, \dots, l_{n_l}\}$ . Consequently we have  $n_\theta \times n_l$  blur kernels  $\mathcal{K} = \{\mathbf{k}_{\theta,l} : \theta \in \Theta, l \in \mathcal{L}\}$ .

Theoretically we can apply the kernels to all normal templates in  $\mathbf{T}_a$ . In our implementation, however, we only use the normal template  $\mathbf{t}_1$  that was manually selected in the first frame. (other normal templates are shifted from it in the first frame). In practice we found that this choice performed well and was computationally efficient. In summary, the blur template set  $\mathbf{T}_b$  is now defined as

$$\mathbf{T}_b = [\mathbf{t}_{1,1}, \dots, \mathbf{t}_{1,n_l}, \mathbf{t}_{2,1}, \dots, \mathbf{t}_{2,n_l}, \dots, \mathbf{t}_{n_\theta,1}, \dots, \mathbf{t}_{n_\theta,n_l}],$$

where  $\mathbf{t}_{i,j} = \mathbf{t}_1 \otimes \mathbf{k}_{\theta_i,l_j}$  is the  $(i,j)^{th}$  blur template and we have in total  $n_b = n_\theta \times n_l$  such templates. Accordingly, for the blur coefficients we have  $\mathbf{b} = [\mathbf{b}_1^\top, \dots, \mathbf{b}_{n_\theta}^\top]^\top$ , where  $\mathbf{b}_i = (b_{i,1}, b_{i,2}, \dots, b_{i,n_l})^\top \in \mathbb{R}^{n_l}$  are coefficients for the  $i^{th}$  direction.

## 2.3. Simultaneous Target Searching and Blur Estimation Through $\ell_1$ Minimization

We are now ready to solve the linear system in (1). A traditional solution is to use least squares approximation, which has been shown in [19, 16] to be less impressive than the sparsity constrained version. In fact, sparsity has been recently intensively exploited for discriminability and robustness against appearance corruption [28].

Inspired by these studies, we rewrite (1) to take into account approximation residuals,

$$\mathbf{y} = [\mathbf{T}_a, \mathbf{T}_b, \mathbf{I}] \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{e} \end{bmatrix} \triangleq \mathbf{T}\mathbf{c}, \quad (3)$$

where  $\mathbf{I}$  is the  $d \times d$  identity matrix containing  $d$  so called *trivial templates*,  $\mathbf{e} = (e_1, e_2, \dots, e_d)^\top \in \mathbb{R}^d$  are *trivial coefficients*,  $\mathbf{T} = [\mathbf{T}_a, \mathbf{T}_b, \mathbf{I}] \in \mathbb{R}^{d \times (n_a + n_b + d)}$  and  $\mathbf{c} = [\mathbf{a}^\top, \mathbf{b}^\top, \mathbf{e}^\top]^\top$ . The trivial templates and coefficients are included to deal with image contaminations such as occlusion.

To achieve a sparse solution, we add an  $\ell_1$ -regularization term [7], which leads to the following  $\ell_1$ -regularized least squares problem

$$\min_{\mathbf{c}} \|\mathbf{T}\mathbf{c} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{c}\|_1, \quad (4)$$

Here, we adopt the recent proposed approach [18] for the minimization task. The solution to (4), denoted as  $\hat{\mathbf{c}} = [\hat{\mathbf{a}}^\top, \hat{\mathbf{b}}^\top, \hat{\mathbf{e}}^\top]^\top$ , is then used to find the tracking result. Specifically, we choose the candidate with the minimum reconstruction error

$$\varepsilon(\mathbf{y}) = \left\| \mathbf{y} - \mathbf{T}_a \hat{\mathbf{a}} - \mathbf{T}_b \hat{\mathbf{b}} \right\|_2^2 \quad (5)$$

as the tracking target. We also use the error to derive the observation likelihood which helps propagate the tracking to next frame (§3).

The blur templates are used to give our approach blur insensitivity. Furthermore, the blur coefficients  $\mathbf{b}$  provide rich information about the distribution of motion blur. This can be attributed to the sparse basis selectivity of the  $\ell_1$  minimization. Intuitively, when there is little blur, a target is clear and therefore only normal templates give a good response in the  $\ell_1$  minimization. On the other hand, when there is motion blur, say along direction  $\theta$ , blur templates along the direction respond actively. Such phenomenon is illustrated in Fig. 2.

Based on this observation, we use distribution of blur coefficients for blur detection and for estimating the motion distribution. We then use the results, in probabilistic fashion, to guide the sample propagation in the particle filter framework as described in the next subsection.

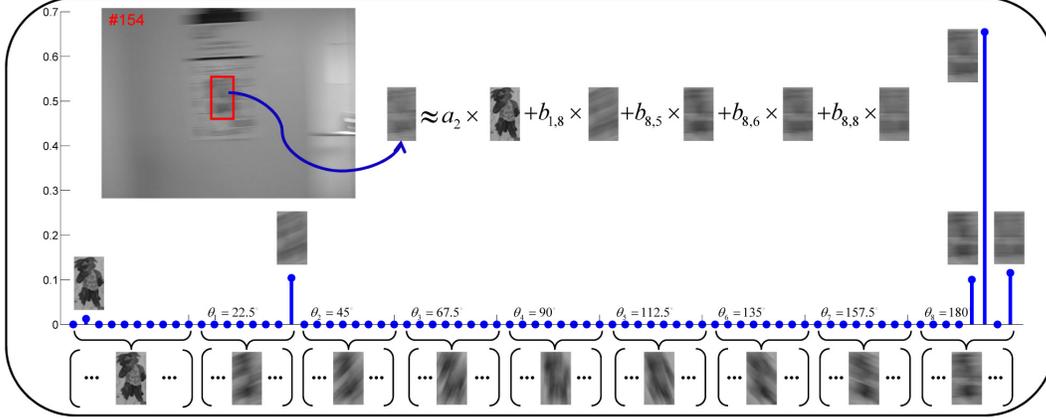


Figure 2. Subspace representation for blur estimation using blurred templates.

### 3. Blur Driven Visual Tracker

#### 3.1. Particle Filter

We use the particle filter framework [11] for the proposed BLUT tracker to integrate the blur insensitivity and motion estimation, achieved using the blur template subspace. The particle filter is a Bayesian sequential importance sampling technique which is widely used to approximate the posterior distribution of state variables for a dynamic system. The framework contains two major steps: prediction and update. In the tracking scenario, we use a state vector  $\mathbf{x}_t$  to describe the location and pose of the tracking target at time  $t$ . The predicting distribution of  $\mathbf{x}_t$  given all available observations (i.e., appearances for tracking)  $\mathbf{y}_{1:t-1} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{t-1}\}$  up to time  $t-1$ , denoted by  $p(\mathbf{x}_t | \mathbf{y}_{1:t-1})$ , is recursively computed as

$$p(\mathbf{x}_t | \mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1}.$$

At time  $t$ , the observation  $\mathbf{y}_t$  is available and the state distribution is updated using Bayes rule

$$p(\mathbf{x}_t | \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t | \mathbf{y}_{1:t-1})},$$

where  $p(\mathbf{y}_t | \mathbf{x}_t)$  denotes the observation likelihood. The posterior  $p(\mathbf{x}_t | \mathbf{y}_{1:t})$  is approximated by a finite set of  $n_p$  weighted samples  $\{\mathbf{x}_t^i, w_t^i : i = 1, \dots, n_p\}$ , where  $w_t^i$  is the importance weight for sample  $\mathbf{x}_t^i$ . The samples are drawn from the so called *proposal distribution*  $q(\mathbf{x}_t | \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t})$  and the weights of the samples are updated according to the following formula:

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

To avoid degeneracy, resampling is applied to generate a set of equally weighted particles according to their importance weights.

To use the particle filter framework, we modelled the observation likelihood and the proposal distribution. For the observation likelihood  $p(\mathbf{y}_t | \mathbf{x}_t)$ , we use the reconstruction error  $\varepsilon(\mathbf{y}_t)$  defined in (5)

$$p(\mathbf{y}_t | \mathbf{x}_t) \propto \exp(-\gamma \varepsilon(\mathbf{y}_t)) \quad (6)$$

for a constant  $\gamma$ . We model the proposal distribution  $q(\mathbf{x}_t | \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t})$ , by fusing information from different sources described in the next subsection.

#### 3.2. Blur-driven Proposal Distribution

It is well known that a good proposal distribution can make the sampled particles more efficient [27, 21]. In this paper, we propose to use the estimated motion information from the  $\ell_1$  minimization to guide the particle sampling process. The idea is to integrate estimated motion information from different sources into the proposal distribution. Specifically, we use the following model:

$$q(\mathbf{x}_t | \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t}) = (w_1 + w_a) p(\mathbf{x}_t | \mathbf{x}_{t-1}) + w_2 p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_{t-2}) + \sum_{i=1}^{n_\theta} w_{b,i} q_i(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_{t-1}), \quad (7)$$

where  $p(\mathbf{x}_t | \mathbf{x}_{t-1}) = \Phi(\mathbf{x}_{t-1}, \sigma_1)$  is used for the first-order Markov transition ( $\Phi(\cdot, \sigma)$  for Gaussian with variance  $\sigma$ );  $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_{t-2}) = \Phi(\mathbf{x}_{t-1} + \mathbf{u}_{t-1}, \sigma_2)$  encodes the second-order Markov transition ( $\mathbf{u}_{t-1} = \mathbf{x}_{t-1} - \mathbf{x}_{t-2}$ ); and  $q_i(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_{t-1})$  is based on the blur motion estimation along direction  $\theta_i$ . The weights  $w_1, w_2 > 0$  are predefined to avoid degenerate cases. Other weights are derived from the  $\ell_1$  minimization:  $w_{b,i} = \sum_{j=1}^{n_i} b_{i,j}$  and  $w_a = \sum_{i=1}^{n_a} a_i$ . The weights are normalized such that  $w_1 + w_2 + w_a + \sum_{i=1}^{n_\theta} w_{b,i} = 1$ .

For  $q_i(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_{t-1})$ , we use the blur coefficients  $\mathbf{b}_i$  along direction  $\theta_i$ ,

$$q_i(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_{t-1}) = \Phi(\mathbf{x}_{t-1} + \mathbf{v}_i, \Sigma_i),$$

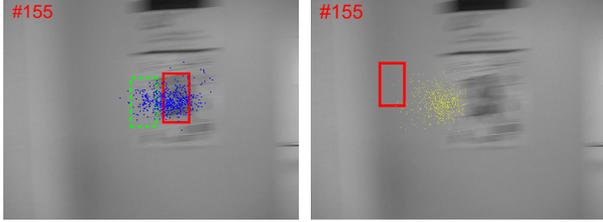


Figure 3. Blur-driven particle sampling. Left: Tracking success with the blur-driven particle sampling. Right: Tracking failure with the traditional first-order Markov transition. Green: tracking result of previous frame; Red: tracking result of current frame; Blue: particles sampled by the blur-driven proposal. Yellow: particles sampled by the traditional first-order Markov transition.

---

**Algorithm 1** BLUT: BLUR-driven Tracker

---

- 1: At  $t = 0$ , initialize template set  $\mathbf{T}_a$  and  $\mathbf{T}_b$
  - 2: Initialize particles
  - 3: **for**  $t = 1, 2, \dots$  **do**
  - 4:   **for** each sample  $i$  **do**
  - 5:     Propagate particles  $\mathbf{x}_t^i$  with respect to the blur-driven proposal  $q(\mathbf{x}_t | \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t})$  via (7).
  - 6:     Compute the transformed target candidate  $\mathbf{y}_t^i$  from  $\mathbf{x}_t^i$ .
  - 7:     Calculate the likelihood  $p(\mathbf{y}_t^i | \mathbf{x}_t^i)$  via (5)(6).
  - 8:   **end for**
  - 9:   Locate the target based on the Maximum Likelihood estimation.
  - 10:   Estimate the blur and motion via the blur coefficients of the estimated target.
  - 11:   Resample particles.
  - 12: **end for**
- 

where  $\mathbf{v}_i = h_i(\cos \theta_i, \sin \theta_i)^\top$  denotes the estimated motion along direction  $\theta_i$  such that  $h_i = \sum_{j=1}^{n_l} b_{i,j} l_j$  is the average motion magnitude in the direction with degree  $\theta_i$ ; and  $\Sigma_i = \Sigma(\theta_i, \sigma_x, \sigma_y)$  is the covariance matrix with orientation  $\theta_i$  and eigen-values  $\sigma_x, \sigma_y$ . In our experiment, we set  $\sigma_x = h_i$  and  $\sigma_y = \sigma_x/2$ .

Note that the direction of  $\mathbf{v}_i$  should be consistent with  $\mathbf{u}_{t-1}$  and the original angle range estimated by our sparse approximation is  $(0, \pi]$ .  $\theta_i$  is adjusted to  $\theta_i + \pi$  according to which angle has the minimum difference to the direction of  $\mathbf{u}_{t-1}$ .

It is worth noting that for a blur-free frame, the estimated  $w_{b,i}$  tends to vanish and our proposed proposal distribution (7) degrades to the common proposal distribution used in many tracking approaches. This way, our tracker unifies the tracking over blur-free frames and blurred frames.

## 4. Results and Discussion

In our framework, we model the state variable  $\mathbf{x}_t$  using three parameters  $\mathbf{x}_t = (t_x, t_y, s)$ , where  $(t_x, t_y)$  are the 2D

translation parameters and  $s$  is the scale variation parameter. The region of interest  $\mathbf{y}_t$  is cropped from the image and scaled to be the same size as the target templates. The observation model  $p(\mathbf{y}_t | \mathbf{x}_t)$  reflects the similarity between a target candidate and the target templates. In this paper,  $p(\mathbf{y}_t | \mathbf{x}_t)$  is formulated from the error approximated by the target templates using  $\ell_1$  minimization.

In order to evaluate the performance of the proposed tracking approach, we collected seven blur video sequences, among which three are taken indoors and the others are of an outdoor traffic scene. We created a tracking groundtruth for quantitative evaluation by manually annotating the data. In total there are 3522 frames used in the experiments; among them around 10% are clear images and 70% have significant blurring effects. Sample clear and blurred frames can be seen in Fig. 5.

We compared the proposed BLUT algorithm with eight state-of-the-art visual trackers. For seven of these trackers, VTD tracker [13], GKT tracker [26], L1 tracker [19], IVT tracker [25], MIL tracker [1], ICTL tracker [29], OAB tracker [9], we use the publicly available code or the code from the original authors. For the other tracker, the classical hsvPF tracker [22], we implemented it ourselves with careful parameter tuning. Note that IVT, MIL, L1, OAB and our proposed BLUT only use grayscale information to track target, while the other algorithms need color information to perform tracking. In our experiments using the public trackers we used the same parameters as the authors. During our experiments with our proposed BLUT we used the same parameters for all of the test sequences. The quantitative results are summarized in Table 1. The tracking results for all the trackers are illustrated in Fig. 5. More tracking results can be found in the supplementary material. Below is a more detailed discussion of the comparison tracking results.

### 4.1. Qualitative Evaluation

We first test our algorithm on the sequences from the indoor scene, where three sequences, *owl*, *face* and *body* are used. The target in sequence *owl* is a plane object, which is frequently and severely blurred. Fig. 5(a) shows a sampling tracking results using different schemes on the *owl* sequence. From the results of #154 and #155, we can see that when target moves fast and blurs severely, most traditional trackers could not follow it. While our proposed BLUT can track the target throughout the sequence. This is because our blur-driven proposal could obtain efficient samples (illustrated in Fig. 3) effectively approximating the blurred target using the proposed blur template subspace representation.

In sequence *face*, the target is blurred together with a slight pose variation (#429). The image results are illustrated in Fig. 5(b). We can see from #302 and #303 that the L1-tracker loses the target due to the large translation of the

	GKT [26]	MIL [1]	OAB [9]	hsvPF [22]	ICTL [29]	VTD [13]	IVT [25]	L1 [19]	BLUT
owl	0.121	0.467	0.544	0.088	0.027	0.683	0.358	0.479	<b>0.011</b>
face	0.081	0.523	0.960	0.069	0.044	0.123	0.039	0.560	<b>0.027</b>
body	0.082	0.613	0.244	0.170	0.105	0.388	0.334	0.331	<b>0.033</b>
car1	0.317	0.626	1.012	0.594	0.193	0.862	0.639	0.622	<b>0.019</b>
car2	0.697	0.775	0.510	0.272	0.297	0.563	1.679	0.555	<b>0.023</b>
car3	0.697	0.569	0.345	0.145	0.152	0.328	0.321	0.538	<b>0.013</b>
car4	0.871	0.974	0.516	0.670	0.304	0.260	0.248	0.452	<b>0.050</b>
Ave.	0.409	0.650	0.590	0.287	0.160	0.458	0.517	0.505	<b>0.025</b>

Table 1. The average tracking errors. The error is measured using the Euclidian distance of two center points, which has been normalized by the size of the target from the ground truth. The last row is the average error for each tracker over all the test sequences.

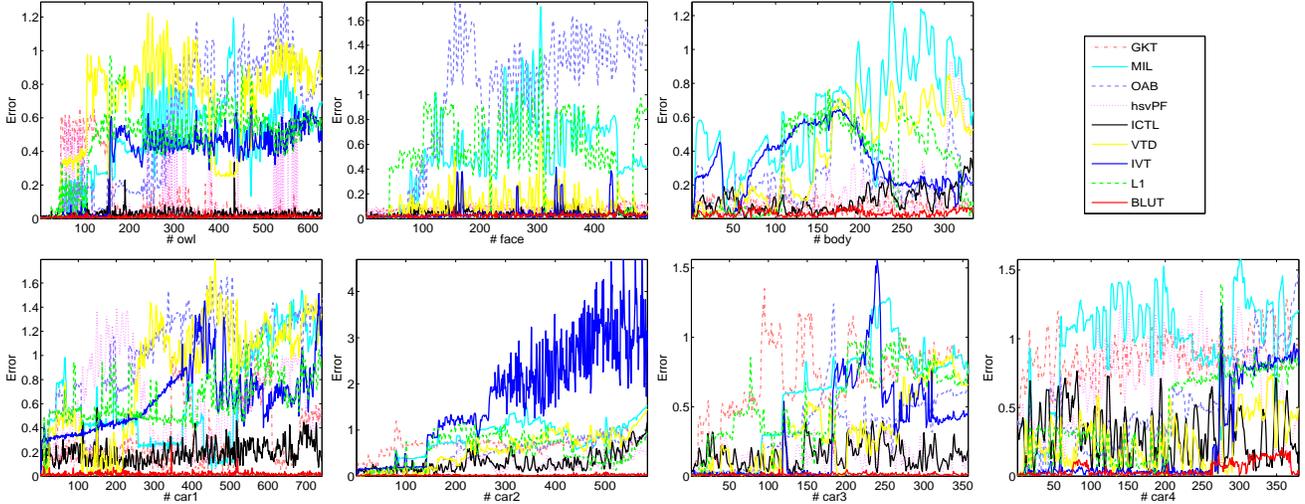


Figure 4. The tracking error plot for each sequence we tested on. The error is measured the same as in Table 1.

target, while our blur-driven proposal could obtain efficient samples to catch the target.

Fig. 5(c) illustrates the tracking results in sequence *body*. The target is moving and is severely blurred. Again, our tracker successfully tracks the target throughout the sequence. We can see from #301 and #302 that our BLUT handles the fast motion together with severe blurring very well, but most other methods perform poorly.

Finally, all the trackers were tested on the sequences captured from outdoor traffic scene, including four sequences named as *car1*, *car2*, *car3* and *car4* respectively. The results for *car1* are shown in Fig. 5(d), as we can see from #521, due to the fast motion and blur the IVT tracker drifts from the target and loses the target in the next frame #522, while our tracker follows the target successfully. Fig. 5(e) shows the results for *car2*. Our tracker can track the target throughout the sequence as it experiences blur and a large change in scale (#581). However, in #79 all other trackers are distracted by the sky whose color is similar to the car. The results for *car3* are shown in Fig. 5(f), from which we notice that the VTD tracker (yellow bounding box) is drifting in #91 and lost after #119. Fig. 5(g) illustrates the

tracking results for *car4*. Again, our BLUT tracks the target throughout the sequences.

## 4.2. Quantitative Evaluation

To quantitatively evaluate all the trackers, we manually labeled the ground truth bounding box of the target in each frame. In Table 1 we give the average tracking errors of each approach in all sequences. From this statistical result we can see that although all the state-of-the-art traditional tracking approaches cannot track the blurred target well, our proposed BLUT can track the blurred target robustly. We can also see that all the trackers which update the target model online, e.g. MIL, OAB, IVT, VTD and L1, give the less reliable results. We argue that the traditional model update scheme does not work on the blurred target. This is why we do not update the tracker for our proposed BLUT. However, model update is very important when the target has other deformations, such as pose variations and illumination changes. Updating the target model in presence of blur is our future work. Fig. 4 illustrates the tracking error plot for each algorithm on each testing sequence. Each subfigure corresponds to one testing sequence, and in



Figure 5. Tracking results of different algorithms. Legend is the same as in Fig. 4. Sequence names are *owl(a)*, *face(b)*, *body(c)*, *car1(d)*, *car2(e)*, *car3(f)* and *car4(g)*.

each subfigure, nine different colored lines represent different trackers. Our proposed BLUT performs better than other state-of-the-art trackers in all examples as shown in the figure and in Table 1.

The reason that BLUT performs well is two-folded: (1) BLUT uses blur templates in addition to the normal templates. This improves the appearance representation in the presence motion blurs; and (2) BLUT employs motion information estimated from the blur to improve the results. Under the  $\ell_1$  minimization framework, BLUT simultaneously tracks the target and estimates the motions.

To further evaluate the effectiveness of the blur template

and blur-driven proposal, we designed two trackers by exclude related information from BLUT. The first one, named LnP (L1 no proposal), is constructed by replacing blur-driven proposal with the common used proposal  $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ . The second one, named LnB (L1 no blur template), excludes blur templates from LnP. Note that LnB can also be viewed as the L1 tracker without model update. We compare the proposed BLUT tracker with the two trackers along with the L1 tracker on all seven blurred sequences. Some example results are shown in Fig. 6. From the results we can see that our BLUT tracker gives better results, which can be attributed to both blur templates and blur-driven proposal.

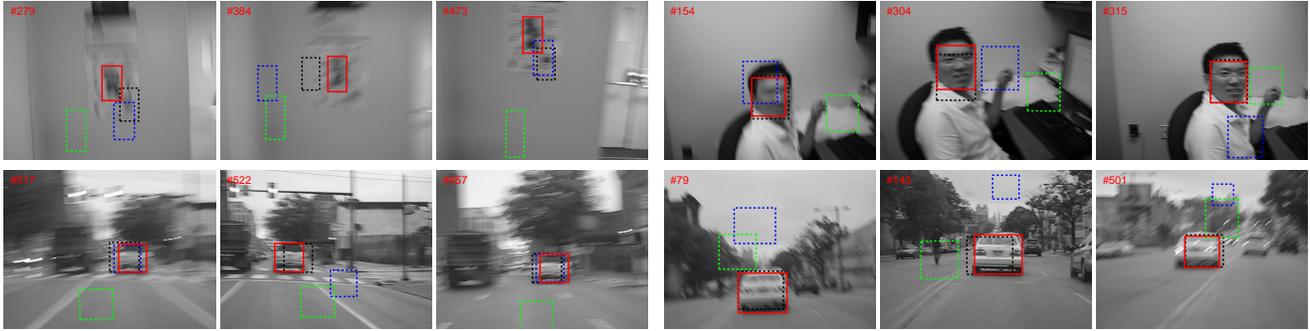


Figure 6. Effects of blur templates and blur-driven proposal. Red: BLUT; Green: L1 tracker; Blue: LnB tracker; Black: LnP tracker.

## 5. Conclusion

We have presented a novel BLUR-driven Tracker (BLUT) framework for tracking motion-blurred targets. BLUT actively uses the information from blurs without performing deblurring. Specifically, we have introduced the blur template subspace and integrated it with the sparse tracking framework. To further improve robustness, we have used blur-driven distribution to guide particle sampling in the particle-filter based tracking framework. Experimental results on a large number of data have shown that BLUT can robustly track motion-blurred targets and outperforms eight state-of-the-art trackers. In the future we plan to exploit other discriminative models for blur-insensitivity. In addition, using the blur estimation to intelligently update template set is also worth investigating.

**Acknowledgment** The work is supported partly by NSF Grants IIS-0916624 and IIS-1049032. Wu is supported partly by National Natural Science Foundation of China (Grant No. 61005027). Yu and Li were partially supported by the AFOSR YIP Award.

## References

- [1] B. Babenko, M. Yang, and S. Belongie. “Visual Tracking with Online Multiple Instance Learning”, in *CVPR*, 2009. 5, 6
- [2] V. Badrinarayanan, P. Pérez, F. L. Clerc, and L. Oisel. “Probabilistic Color and Adaptive Multi-Feature Tracking with Dynamically Switched Priority Between Cues”, *ICCV*, 2007. 1
- [3] J. Cai, H. Ji, C. Liu and Z. Shen, “Blind motion deblurring from a single image using sparse approximation,” *CVPR*, 2009. 1, 2
- [4] S. Dai, M. Yang, Y. Wu, and A. Katsaggelos. “Tracking Motion-Blurred Targets in Video”, in *ICIP*, 2006. 1, 2
- [5] S. Dai and Y. Wu. “Motion from blur”, in *CVPR*, 2008. 1
- [6] Y. Ding, S. McCloskey, and J. Yu. “Analysis of Motion Blur with a Flutter Shutter Camera for Non-linear Motion”, in *ECCV*, 2010. 1
- [7] D. Donoho. “Compressed Sensing”, *IEEE T. on Information Theory*, 52(4):1289-1306, 2006. 3
- [8] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. Freeman. “Removing camera shake from a single photograph”, *ACM T. on Graphics (SIGGRAPH)*, 2006. 1
- [9] H. Grabner, M. Grabner, and H. Bischof. “Real-time tracking via online boosting”, in *BMVC*, 2006. 5, 6
- [10] W. Hu, X. Li, X. Zhang, X. Shi, S.J. Maybank, and Z. Zhang. “Incremental Tensor Subspace Learning and Its Applications to Foreground Segmentation and Tracking.” *IJCV*, 91(3):303-327, 2011. 1
- [11] M. Isard and A. Blake. “Condensation-Conditional Density Propagation for Visual Tracking”, *IJCV*, 29:5-28, 1998. 2, 4
- [12] H. Jin, P. Favaro, and R. Cipolla. “Visual Tracking in the Presence of Motion Blur”, in *CVPR*, 2005. 1
- [13] J. Kwon and K. M. Lee. “Visual Tracking Decomposition”, in *CVPR*, 2010. 5, 6
- [14] A. Levin. “Blind motion deblurring using image statistics Advances”, *NIPS*, 2007. 1
- [15] A. Levin, R. Fergus, F. Durand, and W. Freeman. “Image and depth from a conventional camera with a coded aperture”, *ACM T. on Graphics (SIGGRAPH)*, 2007. 1
- [16] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski. “Robust and Fast Collaborative Tracking with Two Stage Sparse Optimization”, in *ECCV*, 2010. 3
- [17] Y. Lou, A. Bertozzi, and S. Soatto. “Direct Sparse Deblurring”, *Int’l J. Math. Imaging and Vision*, 2010. 1, 2
- [18] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. “Online Learning for Matrix Factorization and Sparse Coding”, *JMLR*, 11:19-60, 2010. 3
- [19] X. Mei and H. Ling. “Robust Visual Tracking using  $\ell_1$  Minimization”, in *ICCV*, 2009. 2, 3, 5, 6
- [20] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. “Minimum Error Bounded Efficient  $\ell_1$  Tracker with Occlusion Detection.” in *CVPR*, 2011. 1
- [21] K. Okuma, A. Taleghani, N. Freitas, J. Little, and D. Lowe. “A boosted particle filter: Multitarget detection and tracking”, in *ECCV*, 2004. 4
- [22] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. “Color-Based Probabilistic Tracking”, in *ECCV*, 2002. 5, 6
- [23] F. Porikli, O. Tuzel, and P. Meer. “Covariance tracking using model update based on Lie algebra”, in *CVPR*, 2006. 1
- [24] W. Richardson. “Bayesian-Based Iterative Method of Image Restoration”, *JOSA*, 62:55-59, 1972. 1
- [25] D. A. Ross, J. Lim, R. Lin and M. Yang. “Incremental learning for robust visual tracking”, *IJCV*, 77:125-141, 2008. 5, 6
- [26] C. Shen, J. Kim and H. Wang. “Generalized Kernel-based Visual Tracking”, *IEEE T. CSVT*, 20(1):119-130, 2010. 5, 6
- [27] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. Wan. “The unscented particle filter”, *NIPS*, 2001. 4
- [28] 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. 3
- [29] Y. Wu, J. Cheng, J. Wang, and H. Lu. “Real-time Visual Tracking via Incremental Covariance Tensor Learning”, *ICCV*, 2009. 5, 6
- [30] A. Yilmaz, O. Javed, and M. Shah. “Object tracking: A survey”, *ACM Computing Surveys*, 38(4), 2006. 1