Kernel-based Motion-blurred Target Tracking

Yi Wu[1 2 3 *], Jing Hu[5], Feng Li[4], Erkang Cheng[3], Jingyi Yu[4] and Haibin Ling[3]

[1] Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing, 210044
[2] School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, 210044
[3] Center for Information Science and Technology, Computer and Information Science Department, Temple University, Philadelphia, PA, USA
{wuyi,hbling,tuc33610}@temple.edu
[4] Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA
{feli,yu}@cis.udel.edu
[5] Network Center, Nanjing University of Information Science & Technology, Nanjing, 210044
hujing@nuist.edu.cn

Abstract. Motion blurs are pervasive in real captured video data, especially for hand-held cameras and smartphone cameras because of their low frame rate and material quality. This paper presents a novel Kernel-based motion-Blurred target Tracking (KBT) approach to accurately locate objects in motion blurred video sequence, without explicitly performing deblurring. To model the underlying motion blurs, we first augment the target model by synthesizing a set of blurred templates from the target with different blur directions and strengths. These templates are then represented by color histograms regularized by an isotropic kernel. To locate the optimal position for each template, we choose to use the mean shift method for iterative optimization. Finally, the optimal region with maximum similarity to its corresponding template is considered as the target. To demonstrate the effectiveness and efficiency of our method, we collect several video sequences with severe motion blurs and compare KBT with other traditional trackers. Experimental results show that our KBT method can robustly and reliably track strong motion blurred targets.

1 Introduction

Object tracking is one of the most important tasks within the field of computer vision. It plays an important role in many applications, such as surveillance, robotics, human computer interaction, and medical image analysis [17]. Most previous work on object tracking have focused on robustly handling noise [13], illumination [1, 15], and occlusions [11, 14]. A common assumption in these algorithms is that the video frames are blur-free. With the prevalence of cheap

* This work was done when Yi Wu was with Temple University.

consumer cameras and smartphone cameras, this assumption is not valid for most of the video data captured using these devices, due to the low frame rate, fast motion of the target and/or hand-shake. Because the visual features of the target and the observation models of trackers are destroyed, this degradation in appearance makes the target inference very challenging in motion blurred sequences. An extensive literature exists deblurring, visual tracking and motion-blurred target tracking.

Debulrring. Intuitively, we could handle severe motion blurs in visual tracking by explicitly deblurring each frame. Previous approaches are usually based on regularization [16], image statistics [5, 8], edge priors [9]. Recently sparse representation is applied to deblurring [10, 3]. Since image deconvolution is a highly ill-posed problem, the latent image reconstructed would have many visual artifacts, such as ringing effects, which destroy the visual features of the target and complicate the object tracking process. Moreover, the deblurring process is computationally expensive and therefore not suitable for real-time visual tracking tasks.

Visual tracking. Many tracking algorithms have been proposed to overcome the tracking difficulties, such as occlusion, background clutter, and illumination changes. In [2], the mean-shift algorithm was adopted to find the optimal location for the target. Isard and Blake [6] treat tracking as a state sequence estimation problem and use the sequential Bayesian inference coupled with Monte Carlo sampling for the solution. Pérez et.al. [12] proposed to integrate the HSV color histogram into the sequential Bayesian inference tracking framework.

Motion-blurred target tracking. The motion-blurred target tracking problem was first addressed in [7] and then further investigated in [4]. In [7], the blurred target regions are estimated by computing the matching score in terms of the region deformation parameters and motion vectors, and then a local gradient descent technique is employed to find the optimal solution. Jin et. al. [7] assume that the blurred target appears highly coherent in the video sequence and the motion between frames is relatively small. In [4], mean-shift tracker with motion-blurred temples is adopted for motion-blurred target tracking.

Although our KBT and [4] share some similarity in using the mean shift tracking with blurred templates, our method has several advantages: 1) [4] has to do local blur classification before handle local motion blurs, while our KBT method not only does not need blur classification but also can effectively deal with both local blurs and global blurs. 2) Our KBT method does not need the off-line training process for blur estimation, while in [4] they have to collect and align a large amount of blurred and non-blurred patches for complicated SVM training. They also suffer from an ambiguous problem about homogeneous regions from the training set. 3) Our KBT does not need a blur direction estimation process, while [4] needs the steerable filter to estimate the blur direction.

In this paper we present a novel *Kernel-based motion-Blurred target Tracking* (KBT) approach without explicitly performing deblurring. Our method incorporates the *blur templates* into the appearance space to model the blur degradations. Specifically, to model the underlying blurs , we augment the the target model by synthesizing various blurred templates of the target with different blur directions and strengths. We represent the templates using color histograms which are regularized by spatial masking with an isotropic kernel. Then we adopt the mean shift procedure to find the optimal the location optimization for each template iteratively. Finally, the optimized region with maximum similarity to its corresponding template is considered as the target.

To evaluate our method, we have collected several video sequences with significant motion blurs. We tested the proposed approach on these sequences and observed promising tracking performances in comparison with several other trackers.

The rest of the paper is organized as follows. In the next section the kernel-based tracking approach is reviewed. After that, the blur modeling approach is proposed in Section 3. Experimental results are reported in Section 4. We conclude this paper in Section 5.

2 Kernel-based tracking

Kernel-based tracking [2] has been proved to be very efficient. Inspired by this work, we use the mean shift procedure to optimize the target location. To handle the blur effects in the target's appearance, we introduce blur templates to augment the template set. This expanded set is useful to handle the underlying blurs.

2.1 Target representation

To characterize the target, the target model is represented by its pdf (m-bin color histogram) q in the feature space. In the subsequent frame, a target candidate at location $\mathbf{y}$ is characterized by the pdf $p(\mathbf{y})$. Thus, the target model and candidate are represented by $\hat{\mathbf{q}} = \{\hat{q}_u\}_{u=1}^{m}, \ \sum_{u=1}^{m} \hat{q}_u = 1$ and $\hat{\mathbf{p}}(\mathbf{y}) = \{\hat{p}_u(\mathbf{y})\}_{u=1}^{m}, \ \sum_{u=1}^{m} \hat{p}_u = 1$, respectively.

A similarity function between $\hat{\mathbf{p}}$ and $\hat{\mathbf{q}}$ is denoted by $\hat{\rho}(\mathbf{y}) \equiv \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}]$, whose local maxima in the image indicate the presence of objects, having representations similar to target model. To find the maxima of such functions, gradient-based optimization procedures are difficult to apply and only an expensive exhaustive search can be used. In the kernel-based tracking [2], the similarity function is regularized by masking the objects with an isotropic kernel in the spatial domain. The kernel weights carry continuous spatial information. When the kernel weights are used in defining the feature space representations, $\hat{\rho}(\mathbf{y})$ becomes a smooth function in $\mathbf{y}$. Thus, gradient-based optimization procedures can be applied to search the target location efficiently.

2.2 Kernel regularization

A differentiable kernel profile, $k(x)$, yields a differentiable similarity function and efficient gradient-based optimizations procedures can be used for finding its maxima. An isotropic kernel can assign smaller weights to pixels farther from the center. Due to the peripheral pixels are less reliable and often affected by clutters, using these weights increases the robustness of the density estimation.

Let $\{\mathbf{x}_i\}_{i=1}^{n}$ be the pixel locations of the target model, centered at 0. Let $b(\mathbf{x}_i)$ be the bin index of the pixel at location $\mathbf{x}_i$ in the quantized feature space. The probability of the feature u in the target model is then computed as

$$\hat{q}_u = C \sum_{i=1}^{n} k\left(\|\mathbf{x}_i\|^2\right) \delta\left[b\left(\mathbf{x}_i\right) - u\right], \tag{1}$$

where δ is the Kronecker delta function and the normalization constant C is

$$C = \frac{1}{\sum_{i=1}^{n} k\left(\|\mathbf{x}_i\|^2\right)}$$

Let the center of the target candidate is at location $\mathbf{y}$ in the current frame. Using the same kernel profile $k(x)$, but with bandwidth h, the probability of the feature u in the target candidate is given by

$$\hat{p}_u\left(\mathbf{y}\right) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{y} - \mathbf{x}_i}{h}\right\|^2\right) \delta\left[b\left(\mathbf{x}_i\right) - u\right]$$

where

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2\right)}$$

is the normalization constant.

2.3 Bhattacharyya Metric

In the kernel-based tracking, Bhattacharyya metric is adopted to accommodate comparisons among various targets. The distance between two discrete distributions is defined as

$$d(\mathbf{y}) = \sqrt{1 - \rho\left[\hat{\mathbf{p}}\left(\mathbf{y}\right), \hat{\mathbf{q}}\right]} \tag{2}$$

where

$$\hat{\rho}\left(\mathbf{y}\right) \equiv \rho\left[\hat{\mathbf{p}}\left(\mathbf{y}\right), \hat{\mathbf{q}}\right] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\mathbf{y})\hat{q}_u}$$

is the sample estimate of the Bhattacharyya coefficient between $\mathbf{p}$ and $\mathbf{q}$.

2.4 Target Localization

To find the location corresponding to the target in the current frame, the distance (2) should be minimized as a function of $\mathbf{y}$. The localization procedure starts from the position of the target in the previous frame and searches in the

Algorithm 1 Kernel-based tracking

1: Given: The target model $\{\hat{q}_u\}_{u=1}^{m}$ and its location $\hat{\mathbf{y}}_0$ in the previous frame.
2: Initialize the location of the target in the current frame with $\hat{\mathbf{y}}_0$, compute $\{\hat{p}_u(\hat{\mathbf{y}}_0)\}_{u=1}^{m}$.
3: Derive the weights $\{w_i\}_{i=1}^{n_h}$ according to (4).
4: Use (3) to get the new location $\hat{\mathbf{y}}_1$ of the target candidate.
5: Compute $\{\hat{p}_u(\hat{\mathbf{y}}_1)\}_{u=1}^{m}$.
6: If $\|\hat{\mathbf{y}}_1 - \hat{\mathbf{y}}_0\| < \varepsilon$ Stop. Otherwise Set $\hat{\mathbf{y}}_0 \leftarrow \hat{\mathbf{y}}_1$ and go to Step 2.

neighborhood. Since the distance function is smooth, the procedure uses gradient information provided by the mean shift vector.

The mode of this density in the local neighborhood can be found by employing the mean shift procedure, where the kernel is recursively moved from the current location $\hat{\mathbf{y}}_0$ to the new location $\hat{\mathbf{y}}_1$ according to

$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i w_i g\left(\left\|\frac{\hat{\mathbf{y}}_0 - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\hat{\mathbf{y}}_0 - \mathbf{x}_i}{h}\right\|^2\right)} \tag{3}$$

where

$$g(x) = -k^{'}(x) \ ,$$

$$w_i = \sum_{u=1}^{m} \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{y}}_0)}}\, \delta\left[b(\mathbf{x}_i - u)\right] \tag{4}$$

The complete target localization algorithm is presented in Algorithm 1.

In our implementation, kernel with Epanechnikov profile

$$k(x) = \begin{cases} \frac{1}{2} c_d^{-1}(d+2)(1-x) & if\ x \le 1 \\ 0 & otherwise \end{cases} \tag{5}$$

is used. In this case, the derivative of the profile, $g(x)$, is constant and (3) is reduced to a simple weighted average:

$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i w_i}{\sum_{i=1}^{n_h} w_i} \tag{6}$$

3 Blur Modeling

To model the underlying blurs for visual tracking, the target model is augmented by synthesizing various blurred templates of the target with different blur directions and strengths.

Let I and I_b be the blur-free and blurred image of a tracking target, respectively. I_b can be modeled as convolving I with a Gaussian blur kernel $\mathbf{k_V}$,

Algorithm 2 KBT tracking

1: Given: The target model set $\{\hat{\mathbf{q}}^n\}_{n=1}^N$, where $\hat{\mathbf{q}}^n = \{\hat{q}_u^n\}_{u=1}^m$ and its location $\hat{\mathbf{y}}$ in the previous frame.
2: **for** each target model n **do**
3: Using algorithm 1 to search its optimized target location $\hat{\mathbf{y}}_n$
4: Set the likelihood $\rho_n = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\hat{\mathbf{y}}_n)\hat{q}_u^n}$
5: **end for**
6: Find the maximum value and corresponding index n^* of $\{\rho_n\}_{n=1}^N$
7: Set the current target location to be $\hat{\mathbf{y}} = \hat{\mathbf{y}}_{n^*}$.

$I_b(\mathbf{p}) = \mathbf{k_V} \otimes I(\mathbf{p})$, 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 $\mathbf{k_{-V}}$.

To capture different blur effects, the manually selected blur-free target template $\mathbf{t}$ in the first frame is convolved with various blur kernels to generate blurred templates. Let the potential motion blurs are governed by the parameter pair θ and l, where θ is used for the motion direction and l for speed. In our implementation, $n_\theta = 8$ different directions $\Theta = \{\theta_1, \cdots, \theta_{n_\theta}\}$ and $n_l = 8$ different speeds $\mathcal{L} = \{l_1, \cdots, l_{n_l}\}$ are used. Thus, we have $n_b = n_\theta \times n_l$ blur kernels $\{\mathbf{k}_{\theta,l} : \theta \in \Theta, l \in \mathcal{L}\}$ and the $(i,j)^{th}$ blur template is defined as $\mathbf{t}_{i,j} = \mathbf{t} \otimes \mathbf{k}_{\theta_i,l_j}$. Consequently, the target template set is augmented from one single template to $N = n_b + 1$ templates.

For each template, kernel regularized color histogram $\hat{\mathbf{q}}^n = \{\hat{q}_u^n\}_{u=1}^m$ is extracted according to (1). Then the mean shift procedure is adopted to perform the location optimization for each template. Finally, the optimized region with maximum similarity to its corresponding template is considered as the target. The complete blurred target localization algorithm is presented in Algorithm 2.

4 Experiments

Our KBT tracker was applied to many sequences. Here, we just present some representative results. In all the sequences, motion blurs are severe and result in the blending of the adjacent colors. We use the Epanechnikov profile for histogram computations and the mean shift iterations were based on weighted average (6).

We compared the proposed KBT algorithm with other traditional trackers: Mean Shift tracker (MS) [2] and Color-based Particle Filtering tracker (CPF) [12]. All the three trackers adopt the RGB color space as feature space which is quantized into $16 \times 16 \times 16$ bins. In our experiments, for each tracker we used the same parameters for all of the test sequences.

We first test our algorithm on the sequence *owl*. The target in sequence *owl* is a plane object, which is frequently and severely blurred. Fig. 1 shows a sampling tracking results using different schemes on the *owl* sequence. We can

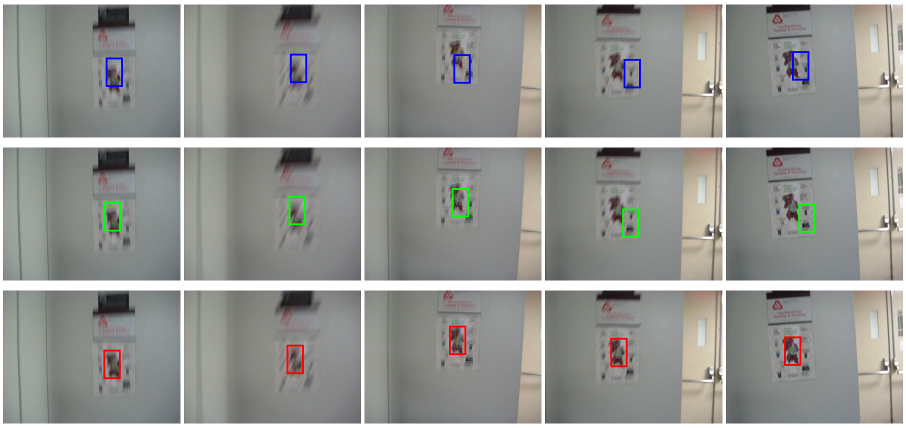

Fig. 1. Tracking comparison results of different algorithms on sequence *owl* (#22, #54, #68, #117, #151). Three examples of CPF, MS and KBT are shown in the rows from top to bottom respectively.

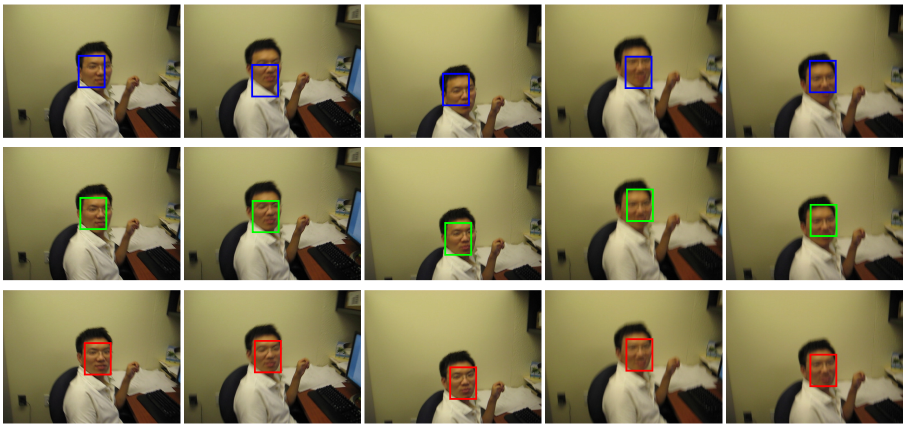

Fig. 2. Tracking comparison results of different algorithms on sequence *face* (#64, #77, #89, #152, #170). Three examples of CPF, MS and KBT are shown in the rows from top to bottom respectively.

see that when target moves fast and blurs severely, MS and CPF trackers could not follow it. While our proposed KBT can track the target throughout the sequence. The image results of *face* are illustrated in Fig. 2. Our proposed KBT achieves better results than the other two tracker. Fig. 3 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.

For all the sequences, we manually labeled the ground truth bounding box of the target in each frame for quantitative evaluation. The error is measured using

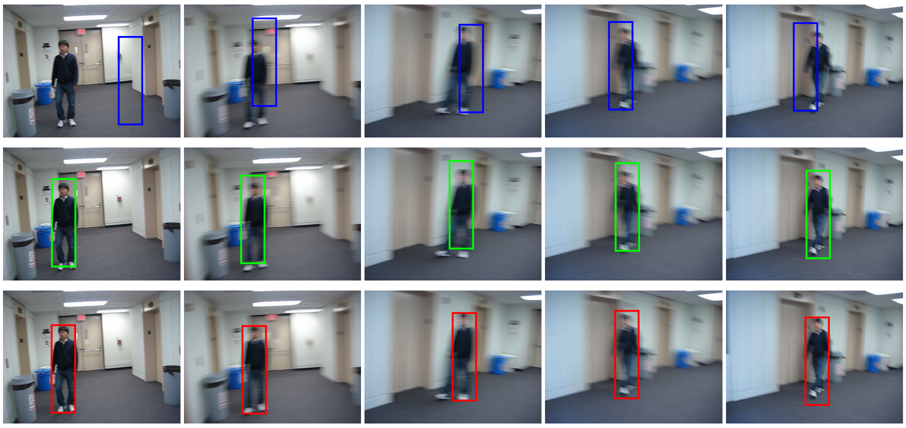

Fig. 3. Tracking comparison results of different algorithms on sequence *body* (#161, #163, #216, #240, #241). Three examples of CPF, MS and KBT are shown in the rows from top to bottom respectively.

the Euclidian distance of two center points, which has been normalized by the size of the target from the ground truth. Fig. 4 illustrates the tracking error plot for each algorithm. From this figure we can see that although all the compared tracking approaches cannot track the blurred target well, our proposed KBT can track the blurred target robustly.

The reason that KBT performs well is that KBT uses blur templates to model the underlying blurs. This improves the appearance representation in the presence of motion blurs.

5 Conclusion

We have presented a novel kernel-based tracker for tracking motion-blurred targets. KBT achieves this challenging tracker task without performing deblurring. Specifically, the target model is augmented by synthesizing various blurred templates of the target with different blur directions and speeds to model the underlying blurs. Each template is represented by a kernel regularized color histogram. Then the mean shift procedure is adopted to perform the location optimization for each template. Finally, the optimized region with maximum similarity to its corresponding template is considered as the target. Experimental results on several challenging video sequences have shown that KBT can robustly track motion-blurred targets and outperforms others traditional trackers.

Acknowledgment This work is supported in part by NSF Grants IIS-0916624 and IIS-1049032. Wu is supported in part by National Natural Science Foundation of China (Grant No. 61005027) and Priority Academic Program Development of Jiangsu Higer Education Institutions.

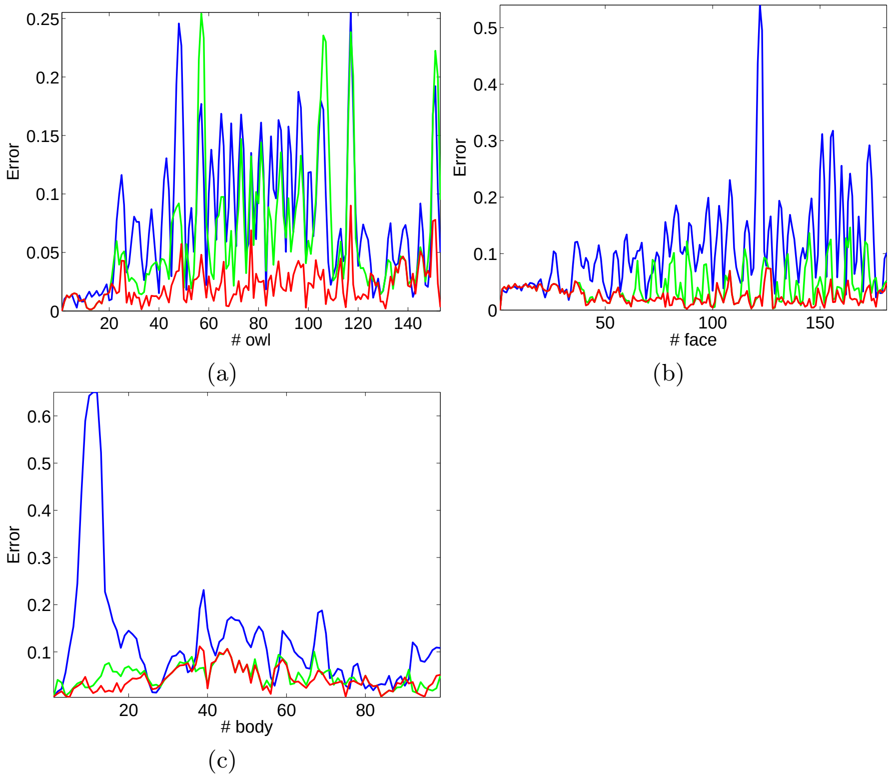

Fig. 4. The tracking error plot for each sequence we tested on. The error is measured using the Euclidian distance of two center points, which has been normalized by the size of the target from the ground truth. Blue: CPF, Green: MS, Red: KBT.

References

1. V. Badrinarayanan, P. Pérez, F. L. Clerc, and L. Oisel. "Probabilistic Color and Adaptive Multi-Feature Tracking with Dynamically Switched Priority Between Cues", *IEEE International Conference on Computer Vision (ICCV)*, 2007.
2. D. Comaniciu, and V. Ramesh, and P. Meer. "Kernel-based object tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25:564-577, 2003.
3. J. Cai, H. Ji, C. Liu and Z. Shen, "Blind motion deblurring from a single image using sparse approximation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
4. S. Dai, M. Yang, Y. Wu, and A. Katsaggelos. "Tracking Motion-Blurred Targets in Video", in *IEEE International Conference on Image Processing (ICIP)*, 2006.
5. R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. Freeman. "Removing camera shake from a single photograph", *ACM T. on Graphics (SIGGRAPH)*, 2006.
6. M. Isard and A. Blake. "Condensation-Conditional Density Propagation for Visual Tracking", *International Journal of Computer Vision (IJCV)*, 29:5-28, 1998.

7. H. Jin, P. Favaro, and R. Cipolla. "Visual Tracking in the Presence of Motion Blur", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.
8. A. Levin. "Blind motion deblurring using image statistics Advances", *Advances in Neural Information Processing Systems (NIPS)*, 2007.
9. 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.
10. Y. Lou, A. Bertozzi, and S. Soatto. "Direct Sparse Deblurring", *Int'l J. Math. Imaging and Vision*, 2010.
11. X. Mei and H. Ling. "Robust Visual Tracking using ℓ_1 Minimization", in *IEEE International Conference on Computer Vision (ICCV)*, 2009.
12. P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. "Color-Based Probabilistic Tracking", in *European Conference on Computer Vision (ECCV)*, 2002.
13. Y. Wu, B. Wu, J. Liu, and H. Q. Lu, "Probabilistic Tracking on Riemannian Manifolds," in *IEEE International Conference on Pattern Recognition (ICPR)*, 2008.
14. Y. Wu, J. Q. Wang, and H. Q. Lu, "Robust Bayesian tracking on Riemannian manifolds via fragments-based representation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009.
15. Y. Wu, J. Cheng, J. Q. Wang, and H. Q. Lu, "Real-time visual tracking via incremental covariance tensor learning," in *IEEE International Conference on Computer Vision (ICCV)*, 2009.
16. W. Richardson. "Bayesian-Based Iterative Method of Image Restoration", *Journal of the Optical Society of America (JOSA)*, 62:55-59, 1972.
17. A. Yilmaz, O. Javed, and M. Shah. "Object tracking: A survey", *ACM Computing Surveys*, 38(4), 2006.