Visual Tracking based on Log-Euclidean Riemannian Sparse Representation

Yi Wu[1], Haibin Ling[1], Erik Blasch[2], Li Bai[3], and Genshe Chen

[1] Center for Data Analytics and Biomedical Informatics, Computer and Information Science Department, Temple University, Philadelphia, PA, USA
[2] Air Force Research Lab/SNAA, OH, USA
[3] Electrical and Computer Engineering Department, Temple University, Philadelphia, PA, USA
{wuyi,lbai,hbling}@temple.edu, erik.blasch@gmail.com

Abstract. Recently, sparse representation has been utilized in many computer vision tasks and adapted for visual tracking. Sparsity-based visual tracking is formulated as searching candidates with minimal reconstruction errors from a template subspace with sparsity constraints in the approximation coefficients. However, an intensity template is easily corrupted by noise and not robust for target tracking under a dynamic environment. The recently proposed covariance region descriptor has been proven robust and versatile for a modest computational cost. Further, the covariance matrix enables efficient fusion of different types of features, where the spatial and statistical properties as well as their correlation are characterized, and its dimension is small. Although the covariance matrix lies on Riemannian manifolds, its log-transformation can be measured on a Euclidean subspace. Based on the covariance region descriptor and using the sparse representation, we propose a novel tracking approach on the Log-Euclidean Riemannian subspace. Specifically, the target region is characterized by a covariance matrix which is then log-transformed from the Riemannian manifold to the Euclidean subspace. After that, the target tracking problem is integrated under a sparse approximation framework, where the sparsity is achieved by solving an ℓ_1-regularization problem. Then the candidate with the smallest approximation is taken as the tracked target. For target propagation, we use the Bayesian state inference framework, which propagates sample distributions over time using the particle filter algorithm. To evaluate our method, we have collected several video sequences and the experimental results show that our tracker can achieve robustly and reliably target tracking.

1 Introduction

Visual tracking is an important problem in computer vision and has many applications in surveillance, robotics, human computer interaction, and medical image analysis [9]. The challenges in designing a robust visual tracking algorithm are mainly caused by the contamination of target appearance and shapes by the presence of noise, occlusion, varying viewpoints, background clutter, and illumination changes. Many tracking algorithms have been proposed to overcome these difficulties. For example, in [27], the sum of squared difference (SSD) has been used as a cost function for the tracking problem. Later, a gradient descent algorithm was used to find the target template with the minimum cost [29]. Also, the mean-shift algorithm was adopted to find the optimal solution [28]. Another view is to treat tracking as a state sequence estimation problem and use the sequential Bayesian inference coupled with Monte Carlo sampling for the solution [1] and extensions included the incorporation of an appearance-adaptive model [31]

Recently, motivated by the breakthrough advances in compressive sensing [26, 15], Mei and Ling [3] introduced the sparse representation for robust visual tracking. The tracking problem is formulated as finding the ℓ_1 minimization of a sparse representation of the target candidate using templates. The advantage of using the sparse representation lies in its robustness to background clutter and occlusions. A tracking candidate is approximated sparsely as a linear combination of target templates and trivial templates, each of them with only one nonzero element. The sparsity is achieved by solving an ℓ_1 minimization problem with non-negativity constraints during tracking. Inspired by this work, further investigation has been conducted in [18, 19] for improvement in different aspects. For example, in [18] the group sparsity is integrated and high-dimensional image features are used for improving tracking robustness. In [19], the less expensive ℓ_2 minimization is used to bound the ℓ_1 approximation error and then speed up the particle resampling process without sacrificing accuracy. Other improvements support multisensor fusion [30] and wide-area imagery [16].

The sparse representation appearance models, adopted in the above mentioned tracking approaches, while straightforward, are sometimes sensitive to the environmental variations as well as pose changes. As a result, these trackers lack a competent object description criterion that captures both statistical and spatial properties of the object appearance. The covariance region descriptor [13] is proposed to characterize the object appearance. Such a descriptor captures the correlations among extracted features inside an object region and is robust to the variations in illumination, view, and pose. A covariance descriptor has been applied to many computer vision tasks, such as object classification [20, 21], human detection [22, 23], face recognition [24], action recognition [25], and tracking [14, 8, 17]. In the recently proposed covariance tracking approach [14], a brute force search approach is adopted and the model is updated by the Riemannian mean under the affine-invariant metric [12]. Then the probabilistic covariance tracking approach is proposed in [6] and extended to multi-part based representation in [7]. Further, to reduce the computational cost for covariance model, update an incremental covariance tensor learning approach is proposed in [8]. Based on the Log-Euclidean Riemannian metric [10], Li *et al.* [11] presented an online subspace learning algorithm which models the appearance changes by incrementally learning an eigenspace representation for each mode of the target through adaptively updating the sample mean and eigenbasis.

There are three main motivations for our work: (1) the prowess of covariance descriptor as appearance models [13] to address feature correlations, (2) the effectiveness of particle filters [1, 4] to account for environmental variations, and (3) the flexibility of sparse representation [3] to utilize compressed sensing. Based on the covariance region descriptor and using the sparse representation, we propose a novel tracking approach on the Log-Euclidean Riemannian subspace. Specifically, the target region is characterized by a covariance matrix which is then log-transformed from the Riemannian manifold to the Euclidean subspace. After that, the target tracking problem is integrated under a sparse approximation framework. The sparsity is achieved by solving an ℓ_1-regularized least squares problem. Then the candidate with the smallest projection error is taken as the tracking target. Finally, tracking is continued using a Bayesian state inference framework in which a particle filter is used to propagate sample distributions over time. To evaluate our method, we tested the proposed approach on several sequences and observed promising tracking performances (i.e. minimum squared error) in comparison with several other trackers.

The rest of the paper is organized as follows. In the next section the proposed Log-Euclidean Riemannian Sparse Representation (LRSR) is discussed. After that, the particle filter algorithm is reviewed in Section 3. Experimental results are reported in Section 4. We conclude this paper in Section 5.

2 Log-Euclidean Riemannian Sparse Representation

The covariance matrix enables efficient fusion of different types of features in small dimensionality. The spatial and statistical properties as well as their correlation are characterized in this descriptor. Although the covariance matrix lies on Riemannian manifolds, its log-transformation can be measured on a Euclidean subspace. Based on the covariance region descriptor and using the sparse representation, we present a novel tracking approach based on the proposed Log-Euclidean Riemannian Sparse Representation (LRSR).

2.1 Covariance descriptor

The covariance region descriptor [13] efficiently fuses heterogeneous features while residing in a low dimensional feature space. In particular, for a window that contains an object of interest, the object is described using the covariance matrix of features collected in the window. Such a descriptor naturally captures the statistical properties is the feature distribution as well as the interactions between different features. In the following we first review the covariance descriptor used in the LRSR tracking method.

Let I be an image defined on a grid Λ of size $W \times H$, and $F \in \mathbb{R}^{W \times H \times d}$ be the d-dimensional feature image extracted from I:

$$F(x, y) = \Phi(I, x, y),$$

where Φ extracts from image I various features such as color, gradients, filter responses, etc. For a given rectangle $\mathcal{R} \subset \Lambda$, let $\{f_i\}_{i=1}^{N}$ be the d-dimensional feature points inside $\mathcal{R}$. Then, the appearance of $\mathcal{R}$ is represented by the $d \times d$ covariance matrix by the following calculation

$$C(\mathcal{R}) = \frac{1}{N-1} \sum_{n=1}^{N} (f_i - \mu)(f_i - \mu)^T,$$

where N is the number of pixels in the region $\mathcal{R}$ and μ is the mean of the feature points. In our work on visual tracking, we define the feature extraction function $\Phi(I, x, y)$ as

$$\Phi(I, x, y) = (x, y, R(x, y), G(x, y), B(x, y), I_x(x, y), I_y(x, y))^\top ,$$

where (x, y) is the pixel location, R, G, B indicate three color channels and I_x, I_y are the intensity gradients. As a result, we achieve a covariance descriptor as a 7×7 symmetric matrix, (i.e., $d = 7$ in our case).

The element (i, j) of C represents the correlation between feature i and feature j. When the extracted d-dimensional feature includes the pixel's coordinate, the covariance descriptor encodes the spatial information of features.

With the help of integral images, the covariance descriptor can be calculated efficiently [13]. When $d(d+1)/2$ integral images are constructed, the covariance descriptor of any rectangular region can be computed independent of the region size.

2.2 Riemannian geometry for covariance matrix

The covariance matrices are well known to form a connected Riemannian manifold[4], which can be locally approximated by a hyperplane. The Log-Euclidean Riemannian (LR) metric [10] was recently introduced for Symmetric Positive-Definite matrices (SPD). From the LR metric, SPD matrices lie in a Lie group $\mathcal{G}$ and the tangent space at the identity element in $\mathcal{G}$ forms a Lie algebra $\mathcal{H}$, which forms a vector space. As a result, the distance between two points $\mathbf{X}$ and $\mathbf{Y}$ on the manifold under the Log-Euclidean Riemannian metric can be easily calculated by $\|\log(\mathbf{X}) - \log(\mathbf{Y})\|$. The Riemannian mean of several elements on the manifold is simply an arithmetic mean of matrix logarithms.

Such a representation has been previously used for visual tracking [14] and has been combined with particle filter [8] for further robustness. In the following, we further extend the framework by integrating the sparsity constraint.

2.3 Target Representation

Inspired by recent work on sparse representation for visual tracking [3], we use a linear subspace representation to model the appearance of a tracking target. We follow notations in [3] whenever applicable. However, instead working on appearance directly, we use the covariance representation described above.

Let $\mathbf{y} \in \mathbb{R}^D, D = d \times d$ (we concatenate elements of the log-transformed covariance matrix into a vector) be the appearance of a tracking target. The appearance is approximated by using a low dimensional subspace spanned by a set of target templates $\mathbf{T}$,

$$\mathbf{y} \approx \mathbf{T}\mathbf{a} = a_1\mathbf{t_1} + a_2\mathbf{t}_2 + \cdots + a_n\mathbf{t}_n \ , \tag{1}$$

where $\mathbf{T} = [\mathbf{t}_1, \cdots, \mathbf{t}_n] \in \mathbb{R}^{D \times n}$ containing n target templates. In addition, $\mathbf{a} = (a_1, a_2, \cdots, a_n)^\top \in \mathbb{R}^n$ are approximation coefficients. At initialization, the first target template is manually selected from the first frame and the rest target templates are created by perturbation of one pixel in four possible directions at the corner points of the first template in the first frame. Thus, we can create all the target templates (10 for our experiments) at initialization.

[4] We enforce non-singularity for these matrices, which is common for image patches since feature vectors from different regions are rarely identical.

2.4 Target Inference through ℓ_1 Minimization

There are many ways to solve the linear approximation in (1). Traditional solutions using least squares approximation have been shown to be less impressive [3, 18] than the sparsity constrained version. Intuitively, sparsity has been recently intensively exploited for discriminability and robustness against appearance corruption [5]. Then, following the work in [3], (1) is reformulated to take into account approximation residuals,

$$\mathbf{y} = [\mathbf{T},\ \mathbf{I}] \begin{bmatrix} \mathbf{a} \\ \mathbf{e} \end{bmatrix} \triangleq \mathbf{T}^{+}\mathbf{c}\ , \tag{2}$$

where $\mathbf{I}$ is a $D \times D$ identity matrix containing D so called *trivial templates*. Each trivial template has only one nonzero element, which encodes the corruption at the corresponding pixel location. Accordingly, $\mathbf{e} = (e_1, e_2, \cdots, e_D)^\top \in \mathbb{R}^D$ are called *trivial coefficients*, $\mathbf{T}^{+} = [\mathbf{T}, \mathbf{I}] \in \mathbb{R}^{D\times(n+D)}$ and $\mathbf{c} = \begin{bmatrix} \mathbf{a} \\ \mathbf{e} \end{bmatrix} \in \mathbb{R}^{n+D}$. The trivial templates and coefficients are included to deal with image contaminations such as occlusion. Note that although we use the same term "template" as in [3], it means the D-dimensional a covariance representation which is different than the original appearance template.

In the above trivial coefficients approximation, the residual error implies how likely a candidate $\mathbf{y}$ comes from previously learned knowledge (i.e. $\mathbf{T}$). In particular, the approximation can represent a target candidate through a linear combination of the template set composed of both target templates and trivial templates. Usually the number of target templates is much smaller than D, which is the number of trivial templates. The intuition is that, a good candidate should be have a sparse representation, which is not usually true for bad tenmplates. The sparsity leads to a sparse coefficient vector and the coefficients corresponding to trivial templates are close to zero. These trivial coefficients can also be used to model occlusion and cluttering, as previously experimented for face recognition. Now the task is to solve the system in (2) with sparse solutions. Toward this end, a ℓ_1-regularization term is added to achieve the sparsity. In other words, the problem turns out to be a ℓ_1-regularized least squares problem

$$\min_{\mathbf{c}} \parallel \mathbf{T}^{+}\mathbf{c} - \mathbf{y} \parallel_2^2 + \lambda \parallel \mathbf{c} \parallel_1\ \ , \tag{3}$$

where $\|\cdot\|_1$ and $\|\cdot\|_2$ indicate the ℓ_1 and ℓ_2 norms respectively. The solution to the above regularization, denoted as $\hat{\mathbf{c}} = \begin{bmatrix} \hat{\mathbf{a}} \\ \hat{\mathbf{e}} \end{bmatrix}$, can then be used to infer the likelihood of $\mathbf{y}$ being a tracking target. In particular, the candidate $\varepsilon(\mathbf{y})$ with the minimum reconstruction error is chosen:

$$\varepsilon(\mathbf{y}) = \|\mathbf{y} - \mathbf{T}\hat{\mathbf{a}}\|_2^2\ . \tag{4}$$

Note in the above error measurement, the coefficients from trivial templates are ignored, since they represent corruptions such as noise or occlusion. Such reconstruction errors will also be used for calculating the observation likelihood used for propagating tracking over frames (see the next section).

To solve (3), we use the recently proposed approach in [2] that sequentially minimizes a quadratic local surrogate of the cost. With an efficient Cholesky-based implementation, the algorithm has been shown to be at least as fast as approaches based on

Algorithm 1 LRSR tracking

At $t = 0$, initialize template set $\mathbf{T}$
Initialize particles
for $t = 1, 2, \ldots$ **do**
  for each sample i **do**
    Propagate particles $\mathbf{x}_t^i$ with respect to the proposal $q\left(\mathbf{x}_t \left| \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t}\right.\right) = p\left(\mathbf{y}_t \left| \mathbf{x}_t\right.\right)$.
    Compute the transformed target candidate $\mathbf{y}_t^i$ from $\mathbf{x}_t^i$.
    Compute the covariance representation of for each candidate.
    Calculate the likelihood $p(\mathbf{y}_t^i|\mathbf{x}_t^i)$ via (4)(5).
  end for
  Locate the target based on the Maximum Likelihood estimation.
  Resample particles.
end for

soft thresholding, while achieving a higher accuracy. Mairal et. al.'s algorithm is based on stochastic approximations and converges almost surely to a stationary point of the objective function and is significantly faster than previous approaches, such as the one used in [3].

3 Combination with the Particle Filter Framework

Following the work in [3] and [8], we combine the proposed sparse covariance representation with the particle filter (PF) [1] for visual tracking. PF models in visual tracking, such as a Bayesian sequential inference problem, where the task is to find the tracking state sequences based on observation sequences. PF uses a Bayesian sequential importance sampling technique to approximate the posterior distribution of state variables for a dynamic system. Such distributions are usually too complicated to be modeled easily with a Gaussian assumption. A PF provides a convenient framework for estimating and propagating the posterior probability density function of state variables regardless of the underlying distribution. The framework mainly composes of two steps: prediction and update, as described below.

We use $\mathbf{x}_t$ to denote target states that describe the location and pose of a tracking target at time t, and we use $\mathbf{y}_t$ for the observations at time t. Furthermore, we denote $p\left(\mathbf{x}_t \left| \mathbf{y}_{1:t-1}\right.\right)$ as the predicting distribution of $\mathbf{x}_t$ given all available observations (i.e., appearances for tracking) $\mathbf{y}_{1:t-1} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{t-1}\}$ up to time $t-1$. The distribution can be recursively computed as

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

Once the prediction and the observation $\mathbf{y}_t$ at time t are available, the state distribution can be updated using the Bayes rule

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

Fig. 1. Tracking comparison results of different algorithms on sequence *VIVID* (#8, #65, #89, #114, #158). The results of CPF, L1 and LRSR are shown in the rows from top to bottom, respectively.

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 $\{(\mathbf{x}_t^i, w_t^i) : i = 1, \cdots, n_p\}$, 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:

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

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

In the above two steps, we need to model the observation likelihood and the proposal distribution, which are based on the proposed sparse covariance representation. Specifically, for the observation likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$, the reconstruction error $\varepsilon(\mathbf{y}_t)$ is used and we have

$$p(\mathbf{y}_t|\mathbf{x}_t) \propto \exp(-\gamma\varepsilon(\mathbf{y}_t)), \tag{5}$$

where γ is constant controlling the shape of the distribution. A common choice of proposal density is by taking $q(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{y}_{1:t}) = p(\mathbf{y}_t|\mathbf{x}_t)$. Consequently, the weights become the local likelihood associated with each state $w_t^i \propto p(\mathbf{y}_t|\mathbf{x}_t^i)$. Finally, the Maximum Likelihood (ML) is performed to estimate current target state. An outline of our tracking algorithm is shown in Algorithm 1. Note that the difference with previously proposed ℓ_1-tracker is mainly on the target representation.

4 Experiments

Our LRSR tracker was applied to many sequences. Here, we just present some representative results. We compared the proposed tracker with other two trackers: L1

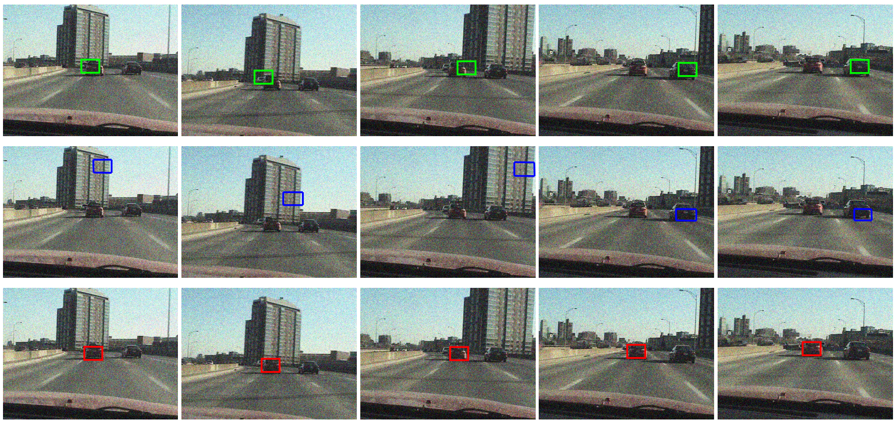

Fig. 2. Tracking comparison results of different algorithms on sequence *carNoise* (#34, #60, #130, #236, #290). The results of CPF, L1 and LRSR are shown in the rows from top to bottom, respectively.

tracker [3] and Color-based Particle Filtering tracker (CPF) [4]. 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 from DARPA VIVID data collection [32], named as *VIVID*. The car in sequence *VIVID* is moving out of and into the shadow of trees frequently. Fig. 1 shows sampling tracking results using different schemes on this sequence. We can see that the appearance of the car is frequently changed and thus the L1 and CPF trackers could not follow the target; however, our proposed LRSR tracker can track the target throughout the sequence. To test the robustness to noise, sequence *carNoise*, which is corrupted by Gaussian noise, is used. The comparative results are shown in Fig. 2. We can see that L1 and CPF cannot follow the target. The poor performance results from their adopted appearance models which are not robust to the noise. Note that the covariance descriptor is robust to the Gaussian noise and the improves the performance the LRSR tracker.

To quantitatively evaluate our proposed tracker, we manually labeled the ground truth bounding box of the target in each frame for the sequence *carNoise*. 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. Fig. 3 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 LRSR tracker can track the target robustly. The reason that LRSR tracker performs well is that it uses covariance descriptor to fuse different types of features, which improves the representation in the presence appearance corruption.

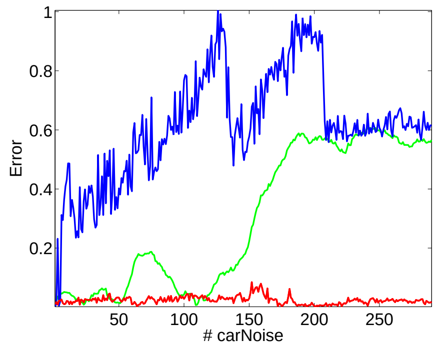

Fig. 3. The tracking error plot for the sequence *carNoise*. The error is measured using the Euclidian distance of two center points, which has been normalized by the size of the target from the ground truth. Green: CPF; Blue: L1; Red: LRSR.

5 Conclusion

In this paper, we have introduced a novel tracking approach on the proposed Log-Euclidean Riemannian sparse representation (LRSR). Specifically, the target region is characterized by a covariance matrix which is then log-transformed from the Riemannian manifold to the Euclidean subspace. After that, the target tracking problem is integrated under a sparse approximation framework. The sparsity is achieved by solving an ℓ_1-regularized least squares problem. Then the candidate with the smallest projection error is taken as the tracking target. Finally, tracking is continued using a Bayesian state inference framework in which a particle filter is used to propagate sample distributions over time. The experimental results show that our tracker can robustly track targets through occlusions, pose changes, and template variations.

Acknowledgment This work is supported in part by NSF Grants IIS-0916624 and IIS-1049032.

References

1. M. Isard and A. Blake. "Condensation-Conditional Density Propagation for Visual Tracking", *Int'l Journal of Computer Vision*, 29:5-28, 1998.
2. J. Mairal, F. Bach, J. Ponce, and G. Sapiro. "Online Learning for Matrix Factorization and Sparse Coding", *J. Machine Learning Research*, 11:19-60, 2010.
3. X. Mei and H. Ling. "Robust Visual Tracking using ℓ_1 Minimization", in *ICCV*, 2009.
4. P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. "Color-Based Probabilistic Tracking", in *ECCV*, 2002.
5. J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. "Robust Face Recognition via Sparse Representation", *IEEE T. Pattern Analysis and Machine Intelligence*, 31(1):210-227, 2009.
6. Y. Wu, B. Wu, J. Liu, and H. Q. Lu, "Probabilistic Tracking on Riemannian Manifolds," in *ICPR*, 2008.
7. Y. Wu, J. Q. Wang, and H. Q. Lu, "Robust Bayesian tracking on Riemannian manifolds via fragments-based representation," in *ICASSP*, 2009.

8. Y. Wu, J. Cheng, J. Wang, and H. Lu. “Real-time Visual Tracking via Incremental Covariance Tensor Learning”, *ICCV*, 2009.
9. A. Yilmaz, O. Javed, and M. Shah. “Object tracking: A survey”, *ACM Computing Surveys*, 38(4), 2006.
10. V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. “Geometric means in a novel vector space structure on symmetric positive-definite matrices”, *SIAM J. on Matrix Analysis and Applications*, 29(1), 2008.
11. X. Li, W. Hu, Z. Zhang, X. Zhang, M. Zhu, and J. Cheng. “Visual tracking via incremental Log-Euclidean Riemannian subspace learning”, in *CVPR*, 2008.
12. X. Pennec, P. Fillard, and N. Ayache. “A Riemannian framework for tensor computing”, *Int’l Journal of Computer Vision*, 66(1):41–66, 2006.
13. O. Tuzel, F. Porikli, and P. Meer. “Region covariance: A fast descriptor for detection and classification.” in *ECCV*, 2006.
14. F. Porikli, O. Tuzel, and P. Meer. “Covariance tracking using model update based on Lie Algebra”.in *CVPR*, pages 728–735, 2006.
15. D. Donoho, “Compressed sensing,” *IEEE T. Information Theory*, 52(4):1289–1306, 2006.
16. H. Ling, Y. Wu, E. Blasch, G. Chen, H. Lang, and L. Bai. “Evaluation of Visual Tracking in Extremely Low Frame Rate Wide Area Motion Imagery.” *Fusion*, 2011.
17. M. Chen, S. K. Pang, T. J. Cham, and A. Goh. “Visual Tracking with Generative Template Model based on Riemannian Manifold of Covariances.” *Fusion*, 2011.
18. 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.
19. X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. “Minimum Error Bounded Efficient ℓ_1 Tracker with Occlusion Detection.” in *CVPR*, 2011.
20. X. Hong, H. Chang, S. Shan, X. Chen, and W. Gao. “Sigma set: A small second order statistical region descriptor.” in *CVPR*, pages 1802–1809, 2009.
21. D. Tosato, M. Farenzena, M. Spera, V. Murino, and M. Cristani. “Multi-class classification on Riemannian manifolds for video surveillance.” in *ECCV*, pages 378–391, 2010.
22. O. Tuzel, F. Porikli, and P. Meer. “Human detection via classification on Riemannian manifolds.” in *CVPR*, 2007.
23. S. Paisitkriangkrai, C. Shen, and J. Zhang. “Fast pedestrian detection using a cascade of boosted covariance features.” *IEEE T. Circuits & Systems for Video Technology*, 18(8):1140–1151, 2008.
24. Y. Pang, Y. Yuan, and X. Li. “Gabor-based region covariance matrices for face recognition.” *IEEE T. Circuits & Systems for Video Technology*, 18(7):989–993, 2008.
25. K. Guo, P. Ishwar, and J. Konrad. “Action change detection in video by covariance matching of silhouette tunnels.” in *ICASSP*, 1110–1113, 2010.
26. E. Candès, J. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Commun. on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.
27. S. Baker and I. Matthews. “Lucas-kanade 20 years on: A unifying framework.” *Int’l Journal of Computer Vision*, 56:221-255, 2004.
28. D. Comaniciu, and V. Ramesh, and P. Meer. “Kernel-based object tracking.” *IEEE T. Pattern Analysis and Machine Intelligence*, 25:564-577, 2003.
29. G. Hager and P. Belhumeur. “Real-time tracking of image regions with changes in geometry and illumination.” in *CVPR*, 403-410, 1996.
30. Y. Wu, E. Blasch, G. Chen, L. Bai, and H. Ling. “Multiple Source Data Fusion via Sparse Representation for Robust Visual Tracking.” in *Fusion*, 2011.
31. S. K. Zhou, R. Chellappa, and B. Moghaddam. “Visual tracking and recognition using appearance-adaptive models in particle filters.” *IEEE T. Image Processing*, 11:1491-1506, 2004.
32. https://www.sdms.afrl.af.mil/index.php?collection=video_sample_set_2.