Multiple Source Data Fusion via Sparse Representation for Robust Visual Tracking

Yi Wu[†] **Erik Blasch**[‡] **Genshe Chen** **Li Bai**[♯] **Haibin Ling**[†]

[†]Computer & Information Science Department, Temple University, Philadelphia, PA, USA
[‡]Defense R&D Canada, QC, Canada
[♯]Electrical & Computer Engineering Department, Temple University, Philadelphia, PA, USA

wuyi@temple.edu, erik.blasch@gmail.com, genshe.chen@ieee.org
lbai@temple.edu, hbling@temple.edu

Abstract—**Information from multiple heterogenous data sources (e.g. visible and infrared) or representations (e.g. intensity and edge) have become increasingly important in many video-based applications. Fusion of information from these sources is critical to improve the robustness of related visual information processing systems. In this paper we propose a data fusion approach via sparse representation with applications to robust visual tracking. Specifically, the image patches from different sources of each target candidate are concatenated into a one-dimensional vector that is then sparsely represented in the target template space. The template space representation, which naturally fuses information from different sources, brings several benefits to visual tracking. First, it inherits robustness to appearance contaminations from the previously proposed sparse trackers. Second, it provides a flexible framework that can easily integrate information from different data sources. Third, it can be used for handling various number of data sources, which is very useful for situations where the data inputs arrive at different frequencies. The sparsity in the representation is achieved by solving an ℓ_1-regularized least squares problem. The tracking result is then determined by finding the candidate with the smallest approximation error. To propagate the results over time, the sparse solution is combined with the Bayesian state inference framework using the particle filter algorithm. We conducted experiments on several real videos with heterogeneous information sources. The results show that the proposed approach can track the target more robustly than several state-of-the-art tracking algorithms.**

Keywords: Tracking, particle filter, sparse representation, information fusion.

I. Introduction

For many applications, multiple source data (e.g. visible and infrared) for the real world or different representations of the visual features (e.g. intensity and edge) are available. The fusion of these data is critical to make the visual information processing system more robust. A visual tracking system augmented with visual feature information is especially useful to robustly follow the target.

In [4], Davis and Sharma presented a background subtraction technique fusing contours from thermal and visible imagery for object detection. In [13] Pérez et. al. proposed generic importance sampling mechanisms for fusing color with either stereo sound for tele-conferencing, or with motion for surveillance.

There are some works focusing on target appearance representation based on the fusion of different types of visual features. Stanley et al. [14] proposed a novel histogram named spatiogram in which each bin is spatially weighted by the mean and covariance of the locations of the pixels that contribute to that bin. Spatiogram captures not only the values of the pixels but their spatial relationships as well. The covariance region descriptor [17]–[20] enables efficient fusion of different types of features and its dimensionality is small. An object window is represented as the covariance matrix of features; the spatial and statistical properties as well as their correlation are characterized within the same representation.

Taking advantage of advances in compressive sensing ([3], [6]), Mei and Ling [10] recently introduced the sparse representation for visual tracking. In the method, the tracking problem is formulated as finding a sparse representation of the target candidate using templates. The advantage of using the sparse representation lies in the robustness to a wide range of image corruptions, such as noises, background clutter and occlusion. 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 an ℓ_1 minimization problem with non-negativity constraints to solve the inverse intensity pattern problem during tracking. Inspired by this work, further investigation has been conducted in ([8], [9], [11]) for improvement in different aspects. For example, in [9] the group sparsity is integrated and very high dimensional image features are used for improving tracking robustness. In [11], 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.

Inspired by the work mentioned above, we propose a data fusion approach via sparse representation with application to visual tracking. Specifically, we extend the original ℓ_1 tracker in [10] by enriching particles or candidates with information from multiple data sources. Each target candidate is described by concatenating patches from different sources, resulting in a one dimensional vector which is longer than that used in

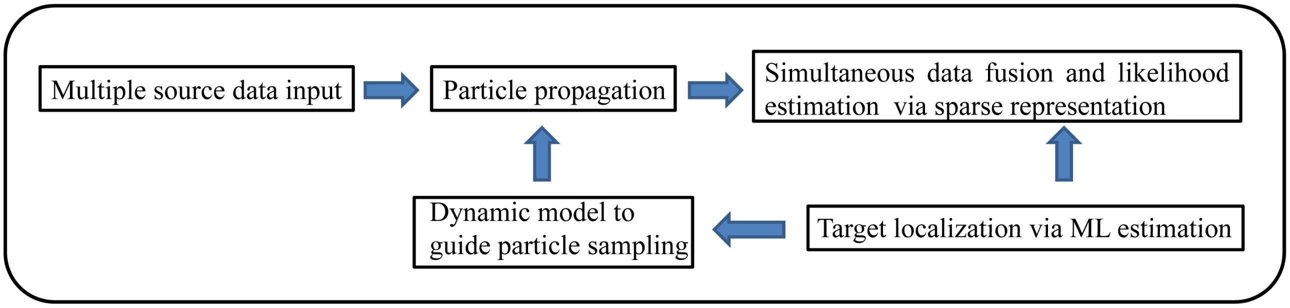
Multiple source data input
Particle propagation
Simultaneous data fusion and likelihood estimation via sparse representation
Dynamic model to guide particle sampling
Target localization via ML estimation

Figure 1. Framework of the proposed tracker.

the original L1 tracker [10]. The vector is then sparsely represented in the target template space, where each template is formed similarly. To achieve the sparsity, the approximation is formulated by solving an ℓ_1-regularized least squares problem. The candidate with the smallest projection error is then taken as the tracking target. After that, tracking is driven by a Bayesian state inference framework in which a particle filter is used to propagate sample distributions across frames. Figure 1 illustrates the idea in the proposed tracker.

The proposed visual tracker enjoys several advantages. First, it inherits from the previously proposed sparse trackers the robustness against appearance contaminations. Such robustness has been shown in previous work for tracking in color videos ([9], [11]) as well as in infrared sequences [8]. Second, it provides a flexible framework that can easily integrate information from different data sources. For example, the inputs from difference sources can have different resolutions, as in many real world applications. Third, it can be used for handling various number of data sources, which is very useful for situations where the data inputs arrive at different frequencies. This is achieved by using dummy input when a data source is unavailable for certain frames. Intuitively, this corresponds to an occlusion in the concatenated candidate representation and such occlusion can be addressed in the sparsity approximation.

To show the benefit of the proposed approach, we conducted experiments on two different tracking scenarios. In the first scenario the task is to track targets in sequences where both color and infrared data are available. In the second one the task involves integrating both color and contour images to improve visual tracking. The effectiveness of the proposed algorithm is demonstrated through these experiments in comparison with state-of-the-art tracking algorithms.

The rest of this paper is organized as follows. The proposed formulation of data fusion for visual tracking is described in Section II. Then, the particle filter for tracking is introduced in Section III. After that, the experimental evaluation is given in Section IV. Finally, we draw conclusions in Section V.

II. Sparse Visual Tracking with Data Fusion

To use the sparse representation for data fusion in the tracking context, we first extend the sparse representation used in [10], where a target is approximated sparsely by a linear subspace. The framework is extended by concatenating information from multiple sources in the representation and then follow a similar Bayesian tracking framework. In the rest of the paper, we follow notations used in [10] whenever applicable.

A. Data Fusion via Sparse Representation

Suppose our data are generated from M different sources $source_1, \ldots, source_M$. At a given time frame, let the appearance[1] of the target from each source represented by $\mathbf{y}_m \in \mathbb{R}^{d_m}$ (we concatenate pixel intensities into a column vector), for $m = 1, \cdots, M$. One solution to fuse these appearances is to model them separately using the sparse representation in [10], i.e.,

$$\mathbf{y}_m \approx \mathbf{T}_m \mathbf{a}_m = a_{1,m}\mathbf{t}_{1,m} + a_{2,m}\mathbf{t}_{2,m} + \cdots + a_{n,m}\mathbf{t}_{n,m} \ , \quad (1)$$

where $\mathbf{T}_m = [\mathbf{t}_{1,m}, \cdots, \mathbf{t}_{n,m}] \in \mathbb{R}^{d_m \times n}$ is the template set for the m-th data source containing n templates $\{\mathbf{t}_{1,m}, \mathbf{t}_{2,m}, \ldots, \mathbf{t}_{n,m}\}$, and $\mathbf{a}_m = (a_{1,m}, a_{2,m}, \ldots, a_{n,m})^\top$ is the approximation coefficient. Such representation can then be fed into the ℓ_1 regularization for sparse solution and then the approximation for different sources are fused to tracking inference.

While such a solution may work to some extent, it requires explicit balancing between different approximation errors, which can be tricky in some applications. In addition, by delaying the fusion step to the final inference stage, only a weak consideration is taken to address the correlations between different data sources. Furthermore, the solution requires solving the ℓ_1 minimization M times, which can be slow for large M.

We propose an alternative solution by concatenating $\mathbf{y}_1, \ldots, \mathbf{y}_M$ into a long vector, named $\mathbf{y}$, as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_M \end{bmatrix} \in \mathbb{R}^D \ , \quad (2)$$

[1] Here we use "appearance" to indicate the responses of the target region in each data source, not necessarily limited to the color appearance as in the color videos.

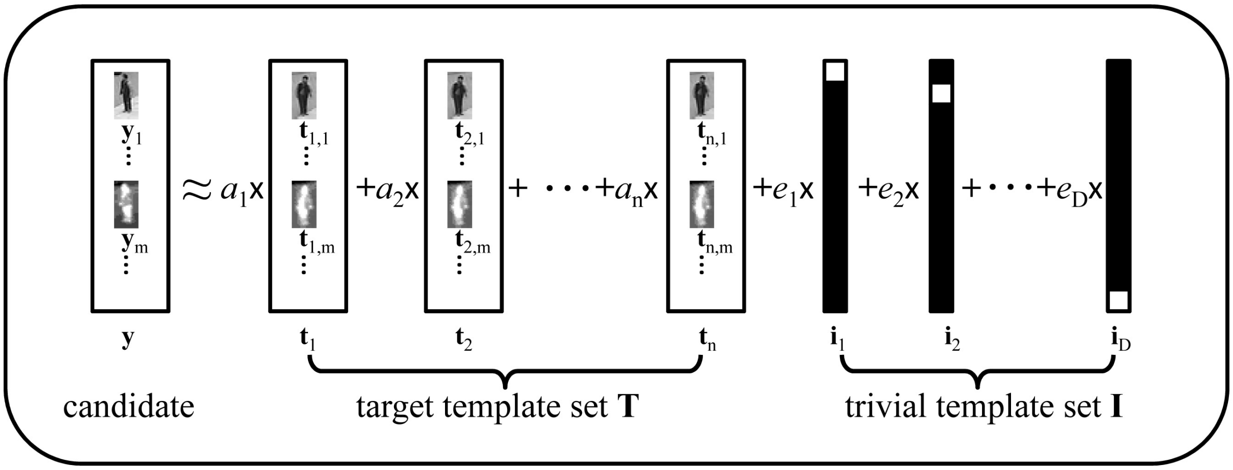

Figure 2. Illustration of the proposed data fusion formulation.

where $D = \sum_{i=m}^{M} d_m$ is the final dimension. Similar operations have been conducted for templates and coefficients. In particular, the i-th template $\mathbf{t}_i$ and the coefficient $\mathbf{a}$ have the form

$$\mathbf{t}_i = \begin{bmatrix} \mathbf{t}_{i,1} \\ \vdots \\ \mathbf{t}_{i,M} \end{bmatrix} \in \mathbb{R}^D \ , \quad \mathbf{a} = \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_M \end{bmatrix} \in \mathbb{R}^n \ . \tag{3}$$

The new target template set is then defined as $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_n] \in \mathbb{R}^{D \times n}$. Finally, we can approximate the candidate $\mathbf{y}$ by the low dimensional subspace spanned by $\mathbf{T}$,

$$\mathbf{y} \approx \mathbf{Ta} = a_1\mathbf{t}_1 + a_2\mathbf{t}_2 + \cdots + a_n\mathbf{t}_n \ , \tag{4}$$

where the indices for a and $\mathbf{t}$ are reordered from 1 to n for notation convenience, and the approximation coefficients $\mathbf{a}$ is called the *target coefficient vector*. This formulation is illustrated in Figure 2.

This representation, not surprisingly, avoids two of the disadvantages in Equation (1). First, only one approximation needs to be solved instead of M of them. Second, the information from different sources is integrated tightly. In addition, the representation also benefits from the implicit balance between different data sources, which is achieved by implicitly weighs them equally (maybe after normalization). While this may not be the optimal strategy, it provides a natural and concise way to address the issue.

One interesting benefit of our representation lies in its ability to handle non-synchronized data sources. For example, at certain frames, there may not be any input from $souce_k$. In our solution, all we need to do is to create a dummy input $\mathbf{y}_k = 0$. This naive solution, by itself, may be arguably ineffectiveness. However, once sparsity is enforced in the approximation, it actually becomes reasonable. Intuitively, the missing elements can be viewed as an occlusion, which can be modeled by the sparse representation ([10], [16]). This point will be further explained in the following section.

B. Target Inference through ℓ_1 Minimization

We are now ready to solve the linear system in (4). A traditional solution is to use least squares approximation, which has been shown in ([9], [10]) to be less impressive than the sparsity constrained version. In fact, sparsity has been recently intensively exploited for discriminability and robustness against appearance corruption [16].

Inspired by these studies, we rewrite (4) to take into account approximation residuals,

$$\mathbf{y} = [\mathbf{T}, \ \mathbf{I}] \begin{bmatrix} \mathbf{a} \\ \mathbf{e} \end{bmatrix} \triangleq \mathbf{T}^{+}\mathbf{c} \ , \tag{5}$$

where $\mathbf{I}$ is the $D \times D$ identity matrix containing D so called *trivial templates*, namely each trivial template has only one nonzero element, $\mathbf{e} = (e_1, e_2, \cdots, e_D)^\top \in \mathbb{R}^D$ are *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.

Then, during tracking, a target candidate is represented as a linear combination of the template set composed of both target templates and trivial templates. The number of target templates are far fewer than the number of trivial templates. Intuitively, a good target candidate can be efficiently represented by the target templates. This leads to a sparse coefficient vector, since coefficients corresponding to trivial templates, namely the trivial coefficients $\mathbf{e}$, tend to vanish. In the case of occlusion (and/or other unpleasant issues such as noise corruption or background clutter), a limited number of trivial coefficients will be activated, but the whole coefficient vector remains sparse. A bad target candidate, on the contrary, often leads to a dense representation. The sparse representation is achieved through solving an ℓ_1-regularized least squares problem, which can be done efficiently through convex optimization. Then the candidate with the smallest target template projection error is chosen as the tracking result.

The system in (5) is underdetermined. To pursue a sparse solution, we add an ℓ_1-regularization term, which is known to typically yield sparse solutions [3], [6]. This leads to the following ℓ_1-regularized least squares problem

$$\min_{\mathbf{c}} \parallel \mathbf{T}^{+}\mathbf{c} - \mathbf{y} \parallel_2^2 + \lambda \parallel \mathbf{c} \parallel_1 \ , \tag{6}$$

where $\| \cdot \|_1$ and $\| \cdot \|_2$ denote for the ℓ_1 and ℓ_2 norms respectively. The solution to (6), denoted as $\hat{\mathbf{c}} = \begin{bmatrix} \hat{\mathbf{a}} \\ \hat{\mathbf{e}} \end{bmatrix}$, is then used to find the tracking result. Specifically, we choose the candidate with the minimum reconstruction error

$$\varepsilon(\mathbf{y}) = \|\mathbf{y} - \mathbf{T}\hat{\mathbf{a}}\|_2^2 \tag{7}$$

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

In this work, we adopt the recent proposed approach [15] for the minimization task (6). In their formulation, the problem is efficiently solved by sequentially minimizing a quadratic local surrogate of the cost. With an efficient Cholesky-based implementation, it has been proven experimentally at least as fast as approaches based on soft thresholding, while providing the solution with a higher accuracy and being more robust as well since it does not require an arbitrary stopping criterion. Their 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 [10].

III. Particle Filter

The particle filter framework ([5], [7]) is adopted for the proposed tracker to propagate sample distributions over time. 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. It provides a convenient framework for estimating and propagating the posterior probability density function of state variables regardless of the underlying distribution. 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) $\mathbf{y}_{1:t-1} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{t-1}\}$ up to time $t-1$, denoted by $p\left(\mathbf{x}_t \left| \mathbf{y}_{1:t-1}\right.\right)$, 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

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

where $p\left(\mathbf{y}_t \left| \mathbf{x}_t\right.\right)$ denotes the observation likelihood. The posterior $p\left(\mathbf{x}_t \left| \mathbf{y}_{1:t}\right.\right)$ 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 x_t^i. The samples are drawn from the so called *proposal distribution* $q\left(\mathbf{x}_t \left| \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t}\right.\right)$ 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.

To use the particle filter framework, we need to model the observation likelihood and the proposal distribution. For the observation likelihood $p\left(\mathbf{y}_t \left| \mathbf{x}_t\right.\right)$, we use the reconstruction error $\varepsilon(\mathbf{y}_t)$ defined in (7)

$$p\left(\mathbf{y}_t \left| \mathbf{x}_t\right.\right) \propto \exp(-\gamma\varepsilon(\mathbf{y}_t)) \tag{8}$$

for a constant γ. The common choice of proposal density is by taking $q\left(\mathbf{x}_t \left| \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t}\right.\right) = p\left(\mathbf{y}_t \left| \mathbf{x}_t\right.\right)$. As a result, the weights become the local likelihood associated with each state $w_t^i \propto p\left(\mathbf{y}_t \left| \mathbf{x}_t^i\right.\right)$. Finally, the Maximum Likelihood (ML) estimation is taken as the current state of the target.

IV. Experimental results

In our framework, we model the state variable $\mathbf{x}_t$ using the 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. Each candidate patch is scaled to be the same size as the target templates. The observation model $p\left(\mathbf{y}_t | \mathbf{x}_t\right)$ reflects the similarity between a target candidate and the target templates. In this paper, $p\left(\mathbf{y}_t | \mathbf{x}_t\right)$ is formulated from the error approximated by the target templates using ℓ_1 minimization.

We compared the proposed algorithm with four state-of-the-art visual trackers, namely, the original sparse representation based tracker without data fusion (L1) [10], Manifold-based Particle Filtering tracker (MPF) [18], Color-based Particle Filtering tracker (CPF) [12], and Mean-Shift tracker (MS) [2]. All the compared algorithms only use color or gray information to perform tracking. During our experiments with all the tested algorithms we used the same parameters for all of the test sequences. The exemplary tracking results for all the sequences along with the information (etc. edge or infrared images) fused by our tracker (L1-PF) are illustrated in Figure 3, Figure 4, Figure 5 and Figure 6. Below is a more detailed discussion of the comparison tracking results.

We first test our algorithm to fuse the intensity and edge information using two sequences, namely *cup* and *crossing*. The Canny edge detector [21] is used to generate the edge map. It is tested on two sequences: *cup* and *crossing*. In the sequence *cup*, which is taken by ourselves, the target is moving against the background with the same color as the target. From the color images of Figure 3, it is very difficult for us to distinguish the cup from the background. However, the edge map gives us the useful information to differentiate the target. Our proposed tracker can effectively fuse intensity with edge and follows the target throughout the sequence. While all the other trackers lose the target when it moves to the background with similar color.

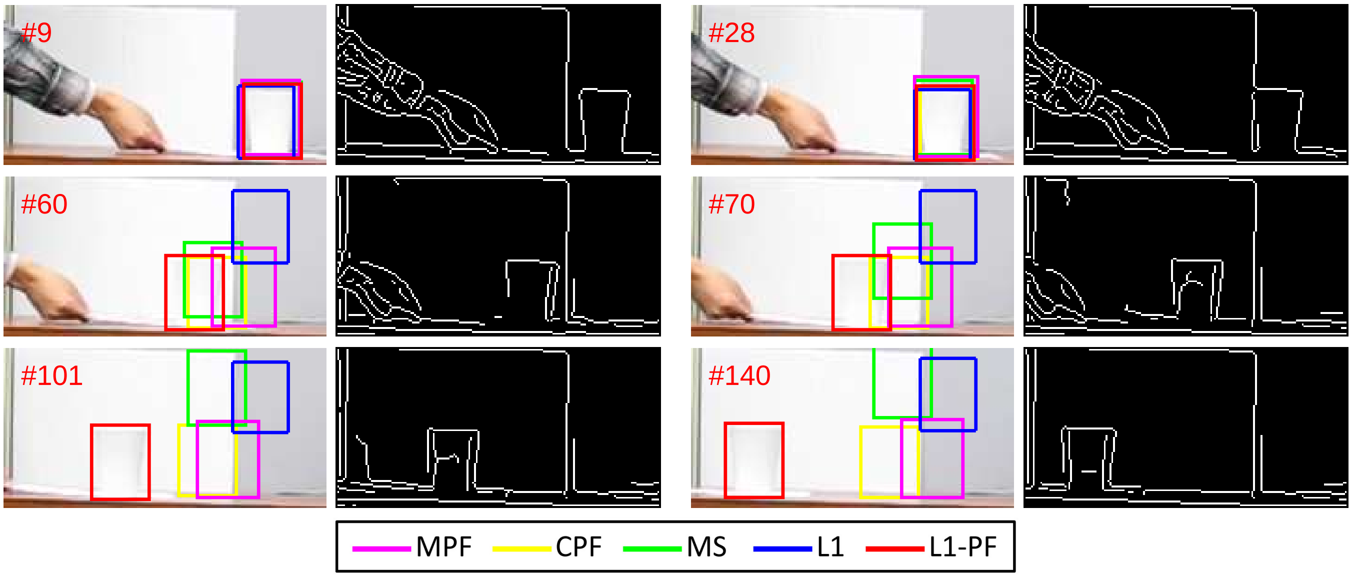

Figure 3. Exemplary results from the sequence *cup*. Canny edge images, which are fused into our tracker, are shown on the right of each color image.

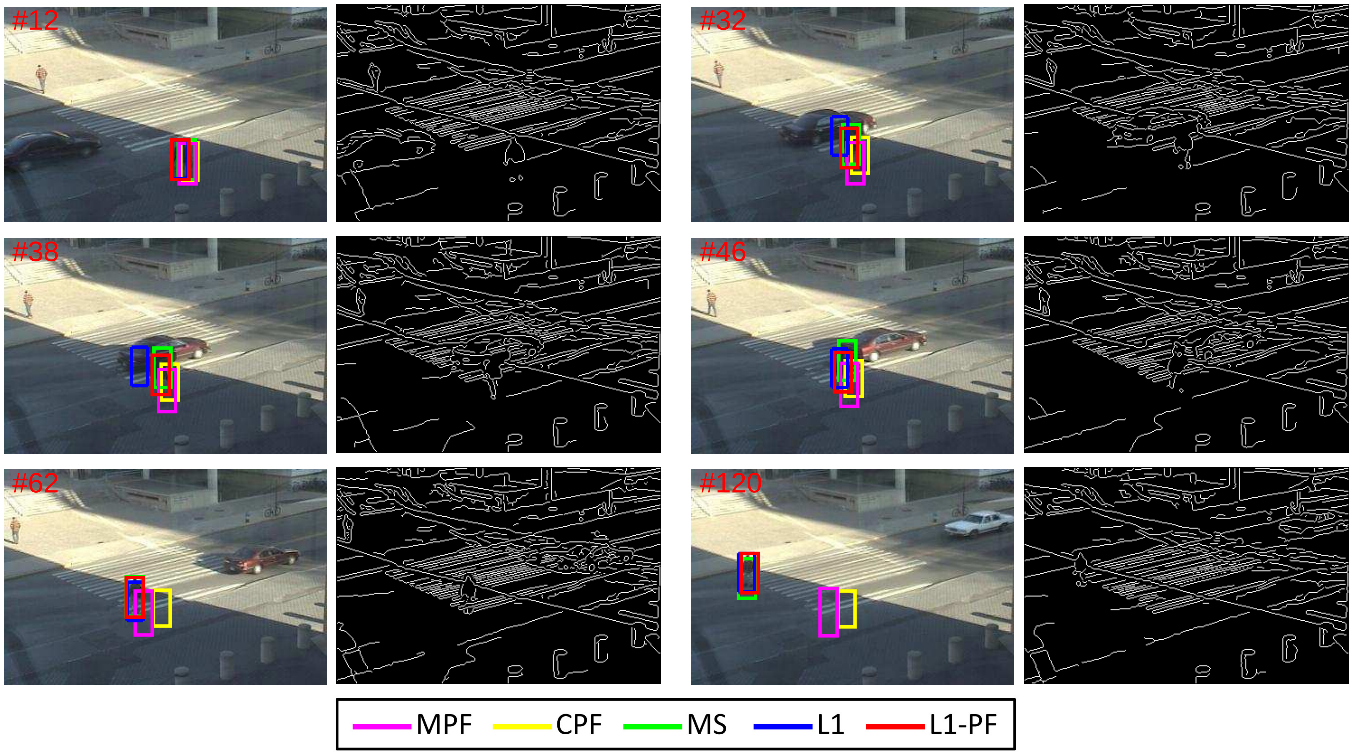

Figure 4. Exemplary results from the sequence *crossing*. Canny edge images, which are fused into our tracker, are shown on the right of each color image.

The sequence *crossing* was used in [1]. In the sequence, the target is crossing the street and the contrast between the target and the background is very low. When intensity is fused with edge information our tracker can track the target throughout the sequence. While the original L1 tracker is drifting off the target when a car with similar color passes the target, eg. #32 and #38. The MPF tracker and CPF tracker lose the target after #46 because the representation they adopted cannot distinguish the target from the background.

Finally, we test our algorithm to fuse the intensity and infrared information on two sequences, *inf1* and *inf2*, from the OSU Color-Thermal Database [4]. The sequence *inf1* is more than one thousand frames. In this sequence, the target is walking and stops to talk to someone else, and then continues walking until out the view of the camera. As illustrated in Figure 5, MS and CPF are attracted to the black trash can

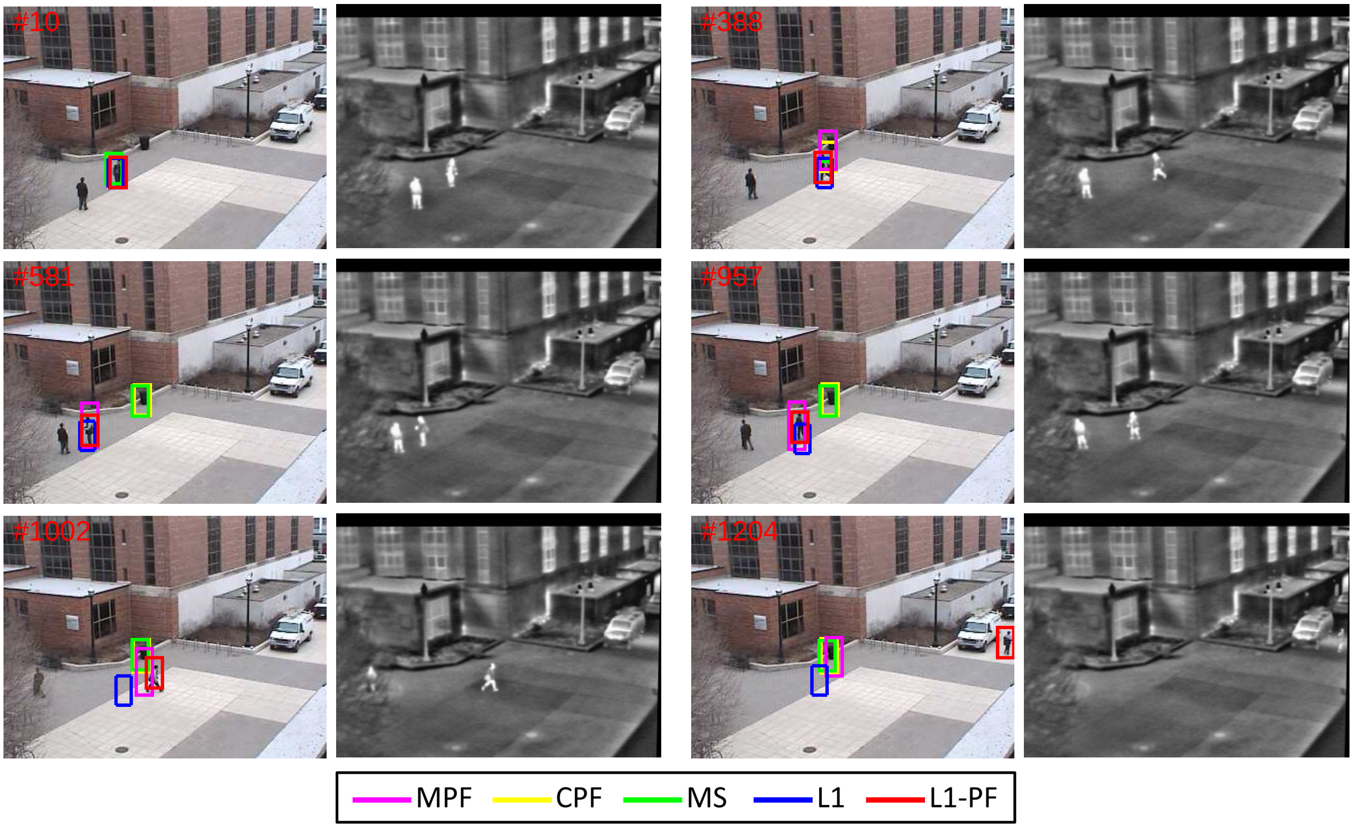

Figure 5. Exemplary results from the sequence *inf1*. Infrared images, which are fused into our tracker, are shown on the right of color images.

in #388 and lose the target after that. From #1002 L1 and MPF begin to drift off the target and lose it after that. While, our tracker fuses the intensity with infrared information and robustly track the target throughout the sequence.

In sequence *inf2*, the target is walking passing a trash can with similar color and then occluded by the branches of a tree. The CPF lose the target first (#27) and then all the other compared trackers are attracted by the can and lose the target after #55. At the end of this sequence, the target is occluded by a tree and it is very difficult to identify it only from the intensity information. However, the infrared information tells us useful information and our proposed approach can effectively fuse it with the intensity information, and thus our tracker follows the target throughout the sequence.

V. Conclusion

In this paper we propose a data fusion approach via sparse representation with application to visual tracking. This representation inherits robustness to appearance contaminations from the previously proposed L1 tracker. It provides a flexible framework that can easily integrate information from different data sources. Further, it can be used for handling various number of data sources, which is very useful for situations where the data inputs arrive at different frequencies. Experiments conducted on several real videos with heterogeneous information sources showed that the proposed approach can track the target more robustly than many state-of-the-art tracking algorithms without data fusion.

Acknowledgment

This work is supported in part by NSF Grant IIS-1049032. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] S. Avidan. “Ensemble Tracking”, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 494–501, 2005.

[2] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[3] E. Candès, J. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[4] J. W. Davis and V. Sharma, “Background-subtraction using contour-based fusion of thermal and visible imagery,” *Computer Vision and Image Understanding*, vol. 106, no. 2-3, pp. 162–182, 2007.

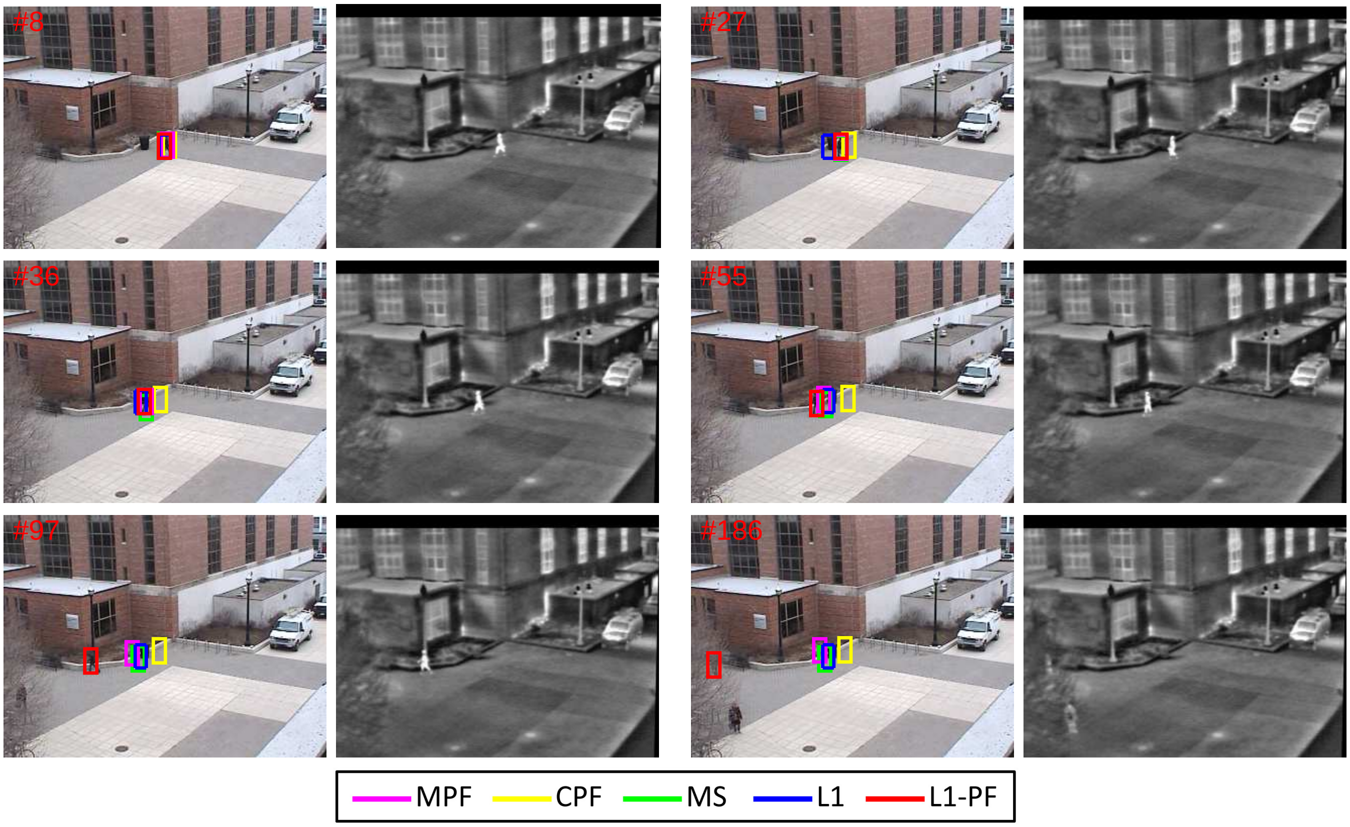

Figure 6. Exemplary results from the sequence *inf2*. Infrared images, which are fused into our tracker, are shown on the right of color images.

[5] A. Doucet, N. de Freitas, and N. Gordon. “Sequential Monte Carlo Methods in Practice”. *Springer-Verlag*, 2001, New York.

[6] D. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[7] M. Isard and A. Blake, “Condensation—conditional density propagation for visual tracking,” *International Journal of Computer Vision*, 29(1):5–28, 1998.

[8] H. Ling, L. Bai, E. Blasch, and X. Mei. “Robust Infrared Vehicle Tracking across Target Pose Change using ℓ_1 Regularization,” *International Conference on Information Fusion*, 2010.

[9] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski, “Robust and fast collaborative tracking with two stage sparse optimization,” in *IEEE European Conference on Computer Vision (ECCV)*, 2010.

[10] X. Mei and H. Ling, “Robust visual tracking using ℓ_1 minimization,” in *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[11] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. “Minimum Error Bounded Efficient ℓ_1 Tracker with Occlusion Detection.” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[12] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, “Color-based probabilistic tracking,” in *IEEE European Conference on Computer Vision (ECCV)*, 2002.

[13] P. Pérez, J. Vermaak, and A. Blake, “Data fusion for visual tracking with particles,” *Proceedings of the IEEE*, vol. 92, no. 3, pp. 495–513, 2005.

[14] S. T. Birchfield and S. Rangarajan, “Spatiograms versus histograms for region-based tracking,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[15] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online Learning for Matrix Factorization and Sparse Coding,” *Journal of Machine Learning Research*, 11:19–60, 2010.

[16] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. “Robust Face Recognition via Sparse Representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):210–227, 2009.

[17] 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.

[18] Y. Wu, B. Wu, J. Liu, and H. Q. Lu, “Probabilistic Tracking on Riemannian Manifolds,” in *IEEE International Conference on Pattern Recognition (ICPR)*, 2008.

[19] 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.

[20] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *IEEE European Conference on Computer Vision (ECCV)*, 2006.

[21] J. F. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 679-698, 1986.