Multi-Cue Visual Tracking Using Robust Feature-Level Fusion Based on Joint Sparse Representation

Xiangyuan Lan Andy J Ma Pong C Yuen

Department of Computer Science, Hong Kong Baptist University
{xylan, jhma, pcyuen}@comp.hkbu.edu.hk

Abstract

The use of multiple features for tracking has been proved as an effective approach because limitation of each feature could be compensated. Since different types of variations such as illumination, occlusion and pose may happen in a video sequence, especially long sequence videos, how to dynamically select the appropriate features is one of the key problems in this approach. To address this issue in multi-cue visual tracking, this paper proposes a new joint sparse representation model for robust feature-level fusion. The proposed method dynamically removes unreliable features to be fused for tracking by using the advantages of sparse representation. As a result, robust tracking performance is obtained. Experimental results on publicly available videos show that the proposed method outperforms both existing sparse representation based and fusion-based trackers.

1. Introduction

Effective modeling of the object's appearance is one of the key issues for the success of a visual tracker [14] and many visual features have been proposed for handling illumination, pose, occlusion and scaling variations [10, 11, 24, 27]. However, because the appearance of target and the environment are dynamically changed, especially in long term videos, a single feature is difficult to deal with all such variations. As such, the use of multiple cues/features to model object appearance has been proposed and proved as a more robust approach for better performance [21, 8, 3, 20, 15]. Many algorithms based on multi-cue appearance model have been proposed for tracking in the past years. Generally, existing multi-cue tracking algorithms can be roughly divided into two categories: score level and feature level. Score-level approach combines classification score corresponding to different visual cues to perform the foreground and background classification. Methods such as online boosting [8, 9], multiple kernel boosting [21] and online multiple instance learning [3]have been proposed. However, the Data Processing Inequality (DPI) [5] indicates that the feature level contains more information than that in the classifier level. Therefore, feature level fusion should be performed to take advantage of the more informative cues for tracking. A typical approach is to concatenate different feature vectors to form a single vector [20]. But such method may result in a high dimensional feature vector which may degrade the tracking efficiency. Moreover, combining all features may not be necessary to improve the tracking performance because not all cues/features are reliable. As such, dynamically selection/combination of visual cues/features is required.

Recently, multi-task joint sparse representation (MTJSR) [19, 22] has been proposed for feature-level fusion in visual classification and promising results have been reported. In MTJSR, the class-level joint sparsity patterns among multiple features are discovered by using a joint sparsity-inducing norm. Therefore, the relationship between different visual cues can be discovered by the joint sparsity constraint. Moreover, high-dimensional features are represented by low-dimensional reconstruction weights for efficient fusion. However, directly applying the MTJSR for object tracking may not achieve convincing performance, since MTJSR was derived based on the assumption that all representation tasks are closely related and share the same sparsity pattern, which may not be valid in tracking application due to unreliable features.

In order to overcome the above-mentioned problem, this paper proposes to remove the negative effect from the unreliable visual cues (outlier) that do not share the same sparsity pattern. Based on joint sparse representation, we propose and develop a new robust feature-level fusion method for visual tracking. It is important to point out that the existing joint sparse representation based tracking algorithms cannot make use of multiple features. For example, Zhang et

al. [26] applied joint sparsity to model relationship between particles to enhance the robustness to significant variations. To the best of our knowledge, this is the first joint sparse representation based multiple feature-level fusion method for visual tracking.

The contributions of this paper are as follows:

- This paper develops a new visual tracking algorithm based on feature-level fusion using joint sparse representation. The proposed method possess all the advantages of joint sparse representation and is able to fuse multiple features for object tracking.
- We propose to detect the unreliable visual cues for the robustness in the feature-level fusion process. By removing the unreliable (outlier) features which introduce negative effect in fusion, the tracking performance can be improved.

2. Related Work

In this section, we give an overview on existing sparse representation based trackers and multi-task joint sparse representation methods related to our proposed method.

Sparse Representation based Tracker Based on the intuition that the appearance of a tracked object can be sparsely represented by its appearance in previous frames, sparse representation based tracker was introduced in [16], which is robust to occlusion and noise corruption. Beyond [16], lots of algorithms have been proposed to improve the tracking accuracy and reduce the computational complexity [25]. Li et al. [13] exploit compressive sensing theory to reduce the template dimension to improve the computational efficiency. Zhang et al. [26] proposed a multi-task joint sparse learning method to exploit the relationship between particles such that the accuracy of L_1 tracker can be improved. Xu et al. [12] developed a local sparse appearance model to enhance the robustness to occlusion. All these sparse representation based trackers utilized a single cue for appearance modeling. To fuse multiple features, Wu et al. [20] concatenated multiple features into a high-dimensional feature vector to construct a template set for sparse representation. However, the high dimensionality of the combined feature vector increases the computational complexity of this method. And, fusion via concatenation may not improve the performance when some source data are corrupted.

Multi-task Joint Sparse Representation In transfer learning, multi-task learning aims to improve the overall performance of related tasks by exploiting the cross-task relationships. Yuan et al. [22] formulated linear representation models from multiple visual features as a multi-task joint sparse representation problem in which multiple features are fused via class-level joint sparsity regularization. Zhang et al. [23] proposed a novel joint dynamic sparsity prior and applied for multi-observation visual recognition. Shekhar et al. [19] proposed a novel multimodal multivariate sparse representation method for multimodal biometrics recognition.

3. Robust Feature-Level Fusion for Multi-Cue Tracking

This section presents the details of the proposed tracking algorithm using robust feature-level fusion based on joint sparse representation. The proposed method consists of two major components: feature-level fusion based on joint sparse representation and detecting unreliable visual cues for robust fusion.

3.1. Multi-Cue Tracking Using Joint Sparse Representation

In the particle filter based multi-cue tracking framework, we are given K types of visual cues, e.g. color, shape and texture, to represent the tracking result in the current frame and template images of the target object. Denote the k-th visual cues of the current tracking result and the n-th template image as y^k and x_n^k, respectively. Inspired by the sparse representation based tracking algorithm [16], the tracking result in the current frame can be sparsely represented by a linear combination of the target templates added by an error vector ε^k for each visual cue, i.e.

$$y^k = X^k w^k + \varepsilon^k, k = 1, \cdots, K \tag{1}$$

where w^k is a weight vector with dimension N to reconstruct the current tracking result with visual cue y^k based on the template set $X^k = [x_1^k, ..., x_N^k]^T$ and N is the number of templates.

In Eq.(1), the weight vectors $w^1, \cdots, w^K$ can be considered as an underlying representation of the tracking result in the current frame with visual cues $y^1, \cdots, y^K$. In other words, the feature-level fusion is given by discovering the relationship between visual cues $y^1, \cdots, y^K$ to determine weight vectors $w^1, \cdots, w^K$ dynamically. To learn the optimal fused representation, we define the objective function by minimizing the reconstruction error and a regularization term, i.e.

$$\min_W \frac{1}{2} \sum_{k=1}^{K} \|y^k - X^k w^k\|_2^2 + \lambda \Omega(W) \tag{2}$$

where $\|\cdot\|_2$ represents L_2 norm, λ is a non-negative parameter, $W = (w^1, ..., w^K) \in \mathbb{R}^{C \times K}$ is the matrix of the weight vectors and Ω is the regularization function on W.

To derive the regularization function Ω, we assume that the current tracking result can be sparsely represented by the same set of chosen target templates with index $n_1, \cdots, n_c$ for each visual cue, i.e.

$$y^k = w_{n_1}^k x_{n_1}^k + \cdots + w_{n_c}^k x_{n_c}^k + \varepsilon^k, k = 1, \cdots, K \tag{3}$$

Under the joint sparsity assumption, the number of chosen target templates $c = \|(\|w_1\|_2, \cdots, \|w_N\|_2)\|_0$ is a small number. Therefore, we can minimize the sparsity measurement as the regularization term in optimization problem (2). Since the L_0 norm can be relaxed by L_1 norm to make the optimization problem tractable, we define Ω as the following equation similar to that in [22] measuring the class-level sparsity for classification applications,

$$\Omega(W) = \|(\|w_1\|_2, \cdots, \|w_N\|_2)\|_1 = \sum_{n=1}^{N} \|w_n\|_2 \quad (4)$$

where w_n denotes the n-th row in matrix W corresponding to the weights of visual cues for the n-th target template. With this formulation, the joint sparsity across different visual cues can be discovered, i.e. w_n becomes zero for a large number of target templates when minimizing optimization problem (2). This ensures that all the selected templates (with non-zero weights) play more important roles in reconstructing the current tracking result for all the visual cues.

3.2. Detecting Unreliable Visual Cues for Robust Feature-Level Fusion

Since some visual cues may be sensitive to illumination or viewpoint change, the assumption about shared sparsity may not be valid for tracking. Such unreliable visual cues of the target cannot be sparsely represented by the same set of the selected target templates. That means, for the unreliable visual cue $y^{k'}$, all the target templates are likely to have non-zero weighting for small reconstruction error, i.e.

$$y^{k'} = w_1^{k'} x_1^{k'} + \cdots + w_N^{k'} x_N^{k'} + \varepsilon^{k'} \quad (5)$$

where $w_1^{k'}, ..., w_N^{k'}$ are non-zero weights. In this case, we cannot obtain robust fusion result by minimizing optimization problem (2) with the regularization function (4).

Although unreliable features cannot satisfy Eq.(3), reliable features can still be sparsely represented by Eq.(3) and used to choose the most informative target templates for reconstruction. With the selected templates of index $n_1, \cdots, n_c$, we rewrite Eq.(5) as follows,

$$y^{k'} - \sum_{i=1}^{c} w_{n_i}^{k'} x_{n_i}^{k'} = \sum_{j=1}^{N-c} w_{m_j}^{k'} x_{m_j}^{k'} + \varepsilon^{k'} \quad (6)$$

where m_j denotes the index for the template which is not chosen to reconstruct the current tracking result. Suppose we have K' unreliable visual cues. Without loss of generality, let visual cues $1, \cdots, K-K'$ be reliable, while $K-K'+1, \cdots, K$ be unreliable. To detect the K' unreliable visual cues, we employ the sparsity assumption for the unreliable features, i.e. the number of unreliable visual cues $K' = \|(\sum_{j=1}^{N-c} |w_{m_j}^1|^2, \cdots, \sum_{j=1}^{N-c} |w_{m_j}^K|^2)\|_0$ is a small number, which can be used to define the regularization function. Similar to Eq.(4), L_1 norm is used instead of L_0 norm. Combining with the regularization function for discovering the joint sparsity among reliable features, Ω becomes

$$\Omega(W) = \theta_1 \sum_{n=1}^{N} \sum_{k=1}^{K-K'} |w_n^k|^2 + \theta_2 \sum_{k=1}^{K} \sum_{j=1}^{N-c} |w_{m_j}^k|^2 \quad (7)$$

where θ_1 and θ_2 are non-negative parameters to balance the joint sparsity across the selected target templates and unreliable visual cues.

However, we have no information about the selected templates and unreliable features before learning, so we cannot define the regularization function like Eq.(7) practically. Inspired by robust multi-task feature learning [7], the weight matrix W can be decomposed into two terms R and S with $W = R+S$. Suppose the non-zero weights of the reliable features be encoded in R, while the non-zero weights of the unreliable features encoded in S. The current tracking result of the reliable visual cue k can be reconstructed by the information in R only, i.e. Eq.(3) is revised as

$$y^k = r_{n_1}^k x_{n_1}^k + \cdots + r_{n_c}^k x_{n_c}^k + \varepsilon^k, k = 1, \cdots, K-K' \quad (8)$$

On the other hand, Eq.(6) for the unreliable feature k' is changed to

$$y^{k'} - \sum_{i=1}^{c} s_{n_i}^{k'} x_{n_i}^{k'} = \sum_{j=1}^{N-c} s_{m_j}^{k'} x_{m_j}^{k'} + \varepsilon^{k'}, \quad k' = K-K'+1, \cdots, K \quad (9)$$

According to the above analysis, the final regularization function can be defined analogous to Eq.(7), i.e.

$$\Omega(W) = \theta_1 \sum_{n=1}^{N} \|r_n\|_2 + \theta_2 \sum_{k=1}^{K} \|s^k\|_2 \quad (10)$$

Denote $\lambda_1 = \lambda\theta_1$ and $\lambda_2 = \lambda\theta_2$. Substituting $\Omega(W)$ by Eq.(10) into optimization problem (2), the proposed robust joint sparse representation based feature-level fusion (RJSR-FFT) model for visual tracking is developed as,

$$\min_{W,R,S} \frac{1}{2} \sum_{k=1}^{K} \|y^k - X^k w^k\|_2^2 + \lambda_1 \sum_{n=1}^{N} \|r_n\|_2 + \lambda_2 \sum_{k=1}^{K} \|s^k\|_2 \quad \text{s.t. } W = R + S \quad (11)$$

The procedures to solve optimization problem (11) will be given in the following section. The optimal fused representation is given by R and S, which encode the information about important target templates and unreliable visual

cues, respectively. With S, we determine the index set O of the unreliable features as

$$O = \{k', \text{s.t.}, \frac{\|s^{k'}\|_2}{\max\{\sum_{k=1}^{K} \|s^k\|_2, \epsilon\}} \geq T\} \tag{12}$$

where ϵ is a positive number to avoid zero division for reliable features. This scheme detects the unreliable visual cues when the norm of some column of matrix S is larger than a pre-defined threshold T.

On the other hand, the likelihood function is defined by R and S as follows. The representation coefficients of different visual cues are estimated and the unreliable features are detected by solving optimization problem (11). Then, the observation likelihood function is defined by

$$p(z_t|l_t) \propto EXP(-\frac{1}{K-K'} \sum_{j \notin O} \|y^j - X^j \cdot r^j\|_2^2) \tag{13}$$

where l_t is the latent state and z_t is the observation in particle filer framework, and the right side of this equation denotes the average reconstruction error of reliable visual cues. Since the proposed model can detect the unreliable cues, the likelihood function can combine the reconstruction error of reliable cues to define the final similarity between the target candidate and the target templates.

3.3. Optimization Procedures

The objective function in optimization problem (11) is given by a smooth function plus a non-smooth one. This kind of optimization problem can be solved efficiently by employing Accelerated Proximal Gradient Method (APG) [4]. Let

$$\begin{aligned} F(R,S) &= \frac{1}{2}\sum_{k=1}^{K} f(r^k, s^k) = \frac{1}{2}\sum_{k=1}^{K} \|y^k - \sum_{n=1}^{N} x_n^k (r_n^k + s_n^k)\|_2^2 \\ G(R,S) &= \lambda_1 \sum_{n=1}^{N} \|r_n\|_2 + \lambda_2 \sum_{k=1}^{K} \|s^k\|_2 \end{aligned} \tag{14}$$

where $F(R,S)$ and $G(R,S)$ are differential and non-differential terms in the objective function, respectively. In the $(t+1)$-th iteration, given the aggregation matrices U^t and V^t, the proximal matrices R^{t+1} and S^{t+1} are given by solving the following minimization problem:

$$\begin{aligned} &\min_{R,S} \frac{1}{2}\sum_{k=1}^{K} \{f(u^{k,t}, v^{k,t}) + \nabla f_{u^{k,t}}^T (r^k - u^{k,t}) \\ &+ \nabla f_{v^{k,t}}^T (s^k - v^{k,t}) + \frac{\mu^{t+1}}{2}\|r^k - u^{k,t}\|_2^2 \\ &+ \frac{\mu^{t+1}}{2}\|s^k - v^{k,t}\|_2^2\} + \lambda_1 \sum_{n=1}^{N} \|r_n\|_2 + \lambda_2 \sum_{k=1}^{K} \|s^k\|_2 \end{aligned} \tag{15}$$

where μ^{t+1} is the Lipschitz constant [4]. Expanding the objective function in optimization problem (15) and neglecting the constant terms, optimization problem (15) can be separated into two independent sub-problems about R and S, respectively, i.e.

$$\begin{aligned} &\min_R \frac{1}{2}\sum_{k=1}^{K} \|r^k - (u^{k,t} - \frac{1}{\mu^{t+1}}\nabla_u^{k,t})\|_2^2 + \frac{\lambda_1}{\mu^{t+1}}\sum_{n=1}^{N} \|r_n\|_2 \\ &\min_S \frac{1}{2}\sum_{k=1}^{K} \|s^k - (v^{k,t} - \frac{1}{\mu^{t+1}}\nabla_v^{k,t})\|_2^2 + \frac{\lambda_2}{\mu^{t+1}}\sum_{k=1}^{K} \|s^k\|_2 \end{aligned} \tag{16}$$

where the gradient operators of f are given by $\nabla_u^{k,t} = -(X^k)^T y^k + (X^k)^T (X^k) u^{k,t} + (X^k)^T (X^k) v^{k,t}$, $\nabla_v^{k,t} = -(X^k)^T y^k + (X^k)^T (X^k) v^{k,t} + (X^k)^T (X^k) u^{k,t}$. The above subproblems in each iteration can be solved in two steps:

Gradient Mapping Step: According to the proved proposition in [18], we updated the proximal matrices R^{t+1} and S^{t+1} by Eq.(17) and Eq.(18), respectively.

$$\begin{aligned} r^{k,t+\frac{1}{2}} &= u^{k,t} - \frac{1}{\mu^{t+1}}\nabla_u^{k,t}, k = 1, \cdots, K, \\ r_n^{t+1} &= \max(0, 1 - \frac{\lambda_1}{\mu^{t+1}\|r_n^{t+\frac{1}{2}}\|_2}) \cdot r_n^{t+\frac{1}{2}}, n = 1, \cdots, N \end{aligned} \tag{17}$$

$$\begin{aligned} s^{k,t+\frac{1}{2}} &= v^{k,t} - \frac{1}{\mu^{t+1}}\nabla_v^{k,t}, k = 1, \cdots, K, \\ s^{k,t+1} &= \max(0, 1 - \frac{\lambda_2}{\mu^{t+1}\|s^{k,t+\frac{1}{2}}\|_2}) \cdot s^{k,t+\frac{1}{2}}, k = 1, \cdots, K \end{aligned} \tag{18}$$

It should be noticed that the update schemes (17) for R and (18) for S are different from each other, since R and S have different sparsity properties grouping according to columns and rows, respectively.

Aggregation Step: We adopt the aggregation matrix update scheme in [4] as follows.

$$\begin{aligned} U^{t+1} &= R^{t+1} + \frac{a_t - 1}{a_{t+1}}(R^{t+1} - R^t), \\ V^{k+1} &= S^{t+1} + \frac{a_t - 1}{a_{t+1}}(S^{t+1} - S^t) \end{aligned} \tag{19}$$

where $a_{t+1} = \frac{1+\sqrt{1+a_t^2}}{2}$, and $a_0 = 1$.

3.4. Template Update Scheme

The proposed tracker is sparse-based. Thus, we adopt the template update scheme in [16] with a small modification because the proposed tracker is also fusion-based tracker with outlier detection scheme. Similar to [16], we associate each template in different visual cues with a weight, and the weight is updated in each frame. Once the similarity

between the template with the largest weight from the reliable visual cue and the target sample of the corresponding visual cue is larger than a predefine threshold, the proposed tracker will replace the template which has the least weight with the target sample. The difference between [16] and the proposed method is that the update scheme in this paper is performed simultaneously for template sets in different visual cues. Once one template of the template set in a visual cue is replaced, the template in other visual cues will be replaced because the proposed model performs multi-cue fusion on feature level. As such, all the cues of the same template should be updated simultaneously.

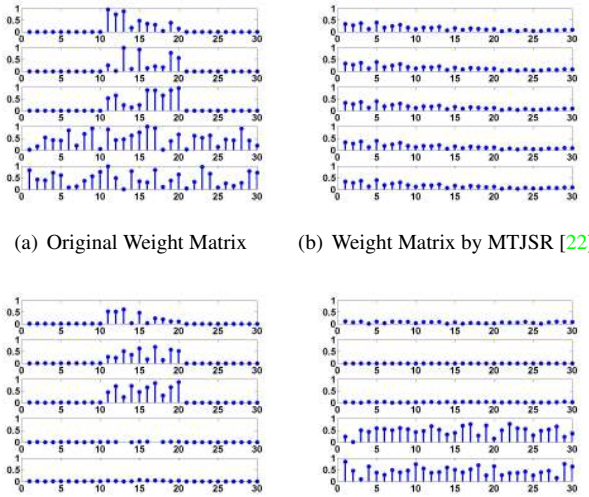
4. Experiment

In this section, we evaluate the proposed robust joint sparse representation based feature-level fusion (RJSR-FFT) tracking algorithm using both synthetic data and real videos for experiments.

4.1. Unreliable Feature Detection on Synthetic Data

To demonstrate that the proposed method can detect unreliable features, we compare the RJSR-FFT with the weight matrices obtained by solving optimization problem (2) with the regularization term (4) as in the multi-task joint sparse representation (MTJSR) method [22]. In this experiment, we simulated the multi-cue tracking problem by randomly generating five kinds of ten dimensional normalized features with 30 templates, i.e. $X^k \in \mathbb{R}^{10\times 30}, k = 1, \cdots, 5$ are the template sets. Two kinds of features are set as unreliable with sparsity patterns. For the other three kinds of reliable features, we divide the template sets into three groups and randomly generate the template weight vector $w^k \in \mathbb{R}^{30}$, such that the elements in w^k corresponding to only one group of templates are non-zero. The testing sample of the k-th feature y^k to represent the current tracking result is computed by $X^k w^k$ plus a Gaussian noise vector with zero mean and variance 0.2 to represent the reconstruction error ε^k. For fair comparison with the MTJSR [22], we extend our model to impose the group lasso penalty by simply using a group sparsity term in optimization problem (11). We empirically set parameters λ, λ_1, λ_2 as 0.001 and the step size μ as 0.002 and repeated this experiment 100 times.

We use the average normalized mean square error between the original weight matrix and recovered one for evaluation. Our method achieves a much lower average recover error of 4.69% compared with that of the MTJSR with 12.29%. This indicates that our method can better recover the underlying weight matrix by detecting the unreliable features successfully. To further demonstrate the ability for unreliable feature detection, we give a graphical illustration of one out of the 100 experiments in Fig.1. The original weight matrix is shown in Fig.1(a) with each row representing a weight vector w^k. The horizontal axis records the sample indexes, while the vertical gives the values of weights. From Fig.1(a), we can see that the first three share the same sparsity patterns over the samples with indexes in the middle range, while all the weights of the last two features are non-zeros, thus non-sparse. In this case, the MTJSR cannot discover the sparsity patterns as shown in Fig.1(b), while the proposed RJSR-FFT can find out the shared sparsity of the reliable features and detect unreliable features as shown in Fig.1(c) and (d). This also explains the reason why our method can better recover the underlying matrix as shown in Fig.1(a).



(a) Original Weight Matrix (b) Weight Matrix by MTJSR [22]

(c) Matrix R by RJSR-FFT (d) Matrix S by RJSR-FFT

Figure 1. Graphical illustration of unreliable feature detection

4.2. Visual Tracking Experiments

While the simulated experiment showed that the proposed method can detect unreliable features, the tracking results with real videos are reported in this section.

4.2.1 Experimental Settings

We evaluate our tracking algorithm on fifteen challenging video sequences with large illumination variations, partial occlusion, pose variations and/or cluttered background. Most videos and its corresponding ground truth data can be found in the website[1]. We compare our tracker with state-of-art tracking algorithms including multi-cue trackers: OAB [8], COV [11], sparse representation based trackers: MTT [26], L1T [16] and other state-of-the-art methods: IVT [17], CT [24], Frag [1]. We use the source code provided by the authors of these papers and adjust the parameters in these methods for better performance.

[1] http://visual-tracking.net/

For our tracking method, we extract seven kinds of local and global features for fusion. For local visual cues, we divide the tracking bounding box into 4 blocks and extract covariance descriptor [11] in each block. For global visual cues, we use HOG [6], LBP [2] and GLF [27] to represent the whole bounding box. The parameters are selected as follows. The number of templates is set as 12. The Lipschitz constant μ is automatically determined according to [7]. We empirically found that the regularization parameters λ_1 and λ_2 are related to μ for robust performance, so we set $\lambda_1 = 0.0027\mu$ and $\lambda_2 = 0.022\mu$. The template size is set to 32×32, while the number of particles is 200.

4.2.2 Quantitative Comparison

Two evaluation criteria are used for quantitative comparison: center location error and success rate. The overlap ratio is define as $\frac{area(B_T \cap B_G)}{area(B_T \cup B_G)}$, where B_T and B_G are the bounding boxes of the tracker and ground-truth. A frame is successfully tracked means that the overlap ratio is larger than 0.5. The center location error is the Euclidean distance between the centers of bounding boxes B_T and B_G. Table 1 and 2 report the center location error and overlapping rate on the 15 videos. With limited space available, we list out frame-by-frame center error comparison results for 8 out of the 15 videos in Fig.2 and more frame-by-frame comparison result can be found in supplementary materials. The best results are shown in red, and the second ones are marked in green. These results show that the proposed method outperforms both multi-cue and sparse representation based trackers as well as state-of-the-art methods in most videos. And, the average center location error of our method is about 7.5 pixels much lower than those of existing trackers, while the successful tracking rate of the proposed tracking algorithm is 90.9% much higher than those of existing methods.

4.2.3 Qualitative Comparison

The video sequences of the tracked results of all trackers in our experiment are provided in supplementary materials and some frames are shown in Fig.3. We qualitatively evaluate the tracking results in four different aspects as follows:

Cluttered Background We test the 8 trackers on several videos(Deer,Football,MountainBike) with cluttered background as shown in Fig.3(a). When the tracked target comes into the dense group of players(Football♯0149), similar pattern of the background distract some trackers from the target, e.g., COV, OAB. Football also pose partial occlusion(Football♯0295), all trackers except our proposed tracker lost the target. This mainly attribute to the fusion of local information in our proposed method so its less sensitive to partial occlusion.

Partial Occlusion FaceOcc1, Girl, David3 pose partial occlusion as shown in Fig.3(b). All tracker can successfully handle the partial occlusion except OAB has small drift from the target(FaceOcc1♯0057). David3 also pose cluttered background and deformation challenge. David3♯0051 show cluttered background distract the tracker, e.g., L1T, COV, Frag, OAB from target. In plan rotation also appears in Girl sequence. CT, Frag, IVT lost the target(Girl♯0246), and CT has small drift.

Non-rigid Target Skating1, Basketball, Crossing show the performance of these trackers when the target is non-rigid as shown in Fig.3(c). Skating1 is the most challenging one with other variation, e.g., in plane rotation(♯0064), partial occlusion(♯0176), Illumination(♯0310)). only our proposed method can track through the sequence.

Illumination Variation Trellis, Car, Shaking, David1, CarDark, Car4 test these trackers under illumination and pose variaton as shown in Fig.3(d). Only our tracker can successfully tracked the target in Trellis and Shaking in all frames.

5. Conclusion

In this paper, we have successfully formulated a feature-level fusion visual tracker based on joint sparse representation. This paper has demonstrated that using proposed robust feature-level fusion of multiple features can improve the tracking accuracy. Experimental results on publicly available videos show that the performance of the proposed tracker using robust joint sparse representation based feature-level fusion model outperforms seven state-of-the-art tracking methods.

Acknowledgements

This project was partially supported by the Science Faculty Research Grant of Hong Kong Baptist University, Hong Kong Research Grants Council General Research Fund 212313 and National Science Foundation of China Research Grant 61172136. The authors would like to thank the reviewers for their helpful comments and thank Mr. K.-Y. Zhao and Mr. G.-C. Mai for their help in debugging the codes of other trackers for comparison experiment.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, 2006. 5, 7

[2] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *ECCV*, 2004. 6

[3] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *TPAMI*, 33(8):1619–1632, 2011. 1

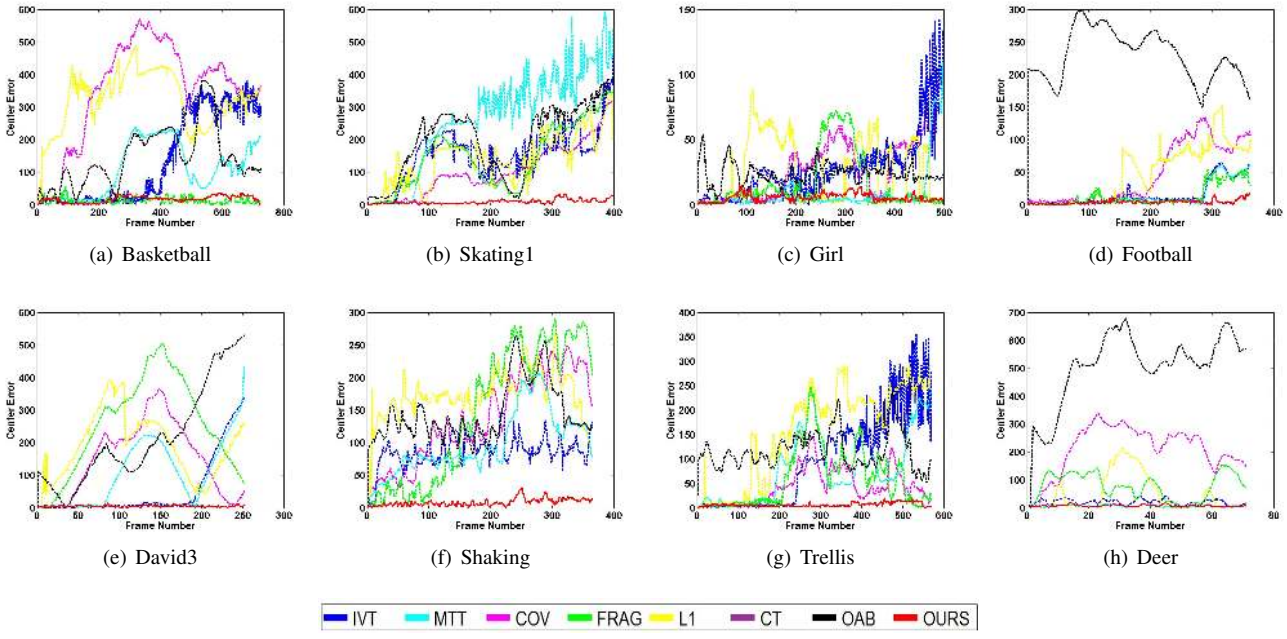
[4] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. 4

	IVT [17]	COV [11]	OAB [8]	CT [24]	Frag [1]	L1T [16]	MTT [26]	Proposed Method
CarDark	1.3	2.7	2.8	120.3	70.7	**1.2**	**1.1**	1.3
Car	42.9	8.0	**3.8**	41.1	42.2	3.9	31.7	**2.8**
Car4	**2.1**	62.5	88.5	81.2	15.0	118.9	238.5	**9.6**
Trellis	97.0	41.8	54.5	48.3	58.3	**31.1**	78.3	**6.4**
David1	**11.2**	57.8	21.5	53.3	63.1	18.5	17.2	**11.1**
Shaking	87.2	137.5	**23.3**	104.1	145.6	145.9	98.0	**8.4**
FaceOcc1	16.6	16.2	40.9	18.4	**11.0**	14.8	22.2	**13.6**
Girl	27.4	27.6	**3.8**	18.4	16.2	9.6	9.1	**5.6**
David3	**52.2**	149.4	193.1	90.0	252.5	189.5	105.0	**5.1**
Deer	20.6	212.2	**6.2**	236.1	78.0	160.9	**5.8**	7.3
Football	15.7	45.6	19.6	**12.8**	13.3	27.6	13.6	**4.4**
MountainBike	21.8	9.4	13.8	213.3	21.1	10.4	**5.8**	**5.6**
Basketball	134.8	351.7	153.3	122.4	**13.1**	106.1	108.6	**17.9**
Crossing	25.4	72.7	**3.0**	4.3	50.1	**3.8**	53.1	4.1
Skating1	154.0	104.5	**48.5**	175.8	147.4	83.6	262.3	**9.5**
Average	47.3	86.6	**45.1**	89.3	66.5	61.7	70.0	**7.5**

Table 1. Quantitative comparison of 8 trackers in 15 videos in terms of center location error (in pixels). The best two results are shown in red and green.

	IVT [17]	COV [11]	OAB [8]	CT [24]	Frag [1]	L1T [16]	MTT [26]	Proposed Method
CarDark	**100**	98.0	90.6	1.0	5.1	**100**	**100**	**100**
Car	44.8	58.6	80.5	5.8	50.6	**82.8**	54.0	**92.0**
Car4	**100**	31.0	27.8	27.9	37.8	27.8	23.1	**99.9**
Trellis	**42.4**	32.0	24.6	32.7	39.9	23.4	28.8	**97.9**
David1	**64.5**	19.3	29.5	24.8	20.2	48.6	25.1	**79.0**
Shaking	3.6	1.1	**48.8**	15.3	14.0	0.8	1.1	**94.5**
FaceOcc1	97.5	**100**	61.1	97.5	**100**	**100**	99.8	**100**
Girl	17.8	31.0	**92.6**	14.0	61.0	70.0	**83.0**	76.4
David3	**69.1**	19.4	15.1	31.0	7.1	4.0	34.5	**99.2**
Deer	45.1	5.6	95.8	4.2	18.3	4.2	**100**	**100**
Football	65.5	42.0	69.1	77.4	71.0	78.2	79.3	**95.9**
MoutainBike	85.1	69.7	71.9	16.7	69.3	86.4	**100**	**100**
Basketball	6.2	5.4	1.1	23.7	**69.0**	25.7	16	**56.1**
Crossing	43.3	9.2	**95.8**	92.5	35.8	**95**	24.1	81.7
Skating1	8.8	15.5	**29.8**	10.5	9.3	20.5	17	**90.5**
Average	52.9	35.9	**55.6**	31.6	40.6	51.1	52.4	**90.9**

Table 2. Quantitative comparison of 8 trackers on 15 videos in terms of success rate (%). The best two results are shown in red and green.



(a) Basketball (b) Skating1 (c) Girl (d) Football

(e) David3 (f) Shaking (g) Trellis (h) Deer

Figure 2. Quantitative frame-by-frame comparison of 8 trackers on 8 Challenging videos in terms of center location error

Figure 3. Qualitative results on some typical frames including some challenging factors.(a) Cluttered background. (b) Partial occlusion. (c) Non-rigid object. (d) Illumination variation.

[5] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley Interscuebce, 2006. 1

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 6

[7] P. Gong, J. Ye, and C. Zhang. Robust multi-task feature learning. In *SIGKDD*, 2012. 3, 6

[8] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR*, 2006. 1, 5, 7

[9] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*. 2008. 1

[10] S. He, Q. Yang, R. W. Lau, J. Wang, and M.-H. Yang. Visual tracking via locality sensitive histograms. In *CVPR*, 2013. 1

[11] W. Hu, X. Li, W. Luo, X. Zhang, S. Maybank, and Z. Zhang. Single and multiple object tracking using log-euclidean riemannian subspace and block-division appearance model. *T-PAMI*, 34(12):2420–2440, 2012. 1, 5, 6, 7

[12] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, 2012. 2

[13] H. Li, C. Shen, and Q. Shi. Real-time visual tracking using compressive sensing. In *CVPR*, 2011. 2

[14] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. v. d. Hengel. A survey of appearance models in visual object tracking. *TIST, in press*, 2013. 1

[15] A. J. Ma, P. C. Yuen, and J.-H. Lai. Linear dependency modeling for classifier fusion and feature combination. *TPAMI*, 35(5):1135–1148, 2013. 1

[16] X. Mei and H. Ling. Robust visual tracking and vehicle classification via sparse representation. *TPAMI*, 33(11):2259–2272, 2011. 2, 4, 5, 7

[17] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1-3):125–141, 2008. 5, 7

[18] M. W. Schmidt, E. Berg, M. P. Friedlander, and K. P. Murphy. Optimizing costly functions with simple constraints: A limited-memory projected quasi-newton algorithm. In *AISTATS*, 2009. 4

[19] S. Shekhar, V. M. Patel, N. M. Nasrabadi, and R. Chellappa. Joint sparsity-based robust multimodal biometrics recognition. In *ECCV*, 2012. 1, 2

[20] 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. 1, 2

[21] F. Yang, H. Lu, and M.-H. Yang. Robust visual tracking via multiple kernel boosting with affinity constraints. *TCSVT, in press*, 2013. 1

[22] X.-T. Yuan, X. Liu, and S. Yan. Visual classification with multitask joint sparse representation. *TIP*, 21(10):4349–4360, 2012. 1, 2, 3, 5

[23] H. Zhang, N. M. Nasrabadi, Y. Zhang, and T. S. Huang. Multi-observation visual recognition via joint dynamic sparse representation. In *ICCV*, 2011. 2

[24] K. Zhang, L. Zhang, and M.-H. Yang. Real-time compressive tracking. In *ECCV*. 2012. 1, 5, 7

[25] S. Zhang, H. Yao, X. Sun, and X. Lu. Sparse coding based visual tracking: Review and experimental comparison. *Pattern Recognition*, 46(7):1772–1788, 2013. 2

[26] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via structured multi-task sparse learning. *IJCV*, 101(2):367–383, 2013. 2, 5, 7

[27] W. W. Zou, P. C. Yuen, and R. Chellappa. A low resolution face tracker robust to illumination variations. *TIP*, 22(5):1726–1739, 2013. 1, 6