Weighted Sparse Representation Regularized Graph Learning for RGB-T Object Tracking

Chenglong Li
School of Computer Science and Technology, Anhui University
Hefei, China 230601
lcl1314@foxmail.com

Nan Zhao
School of Computer Science and Technology, Anhui University
Hefei, China 230601
zhn1528@gmail.com

Yijuan Lu
Department of Computer Science, Texas State University
San Marcos, USA
lu@txstate.edu

Chengli Zhu
School of Computer Science and Technology, Anhui University
Hefei, China 230601
zcl912@foxmail.com

Jin Tang*
School of Computer Science and Technology, Anhui University
Key Lab of Industrial Image Processing and Analysis of Anhui Province
Hefei, China 230601
jtang99029@foxmail.com

ABSTRACT

In this paper, we propose a novel graph model, called weighted sparse representation regularized graph, to learn a robust object representation using multispectral (RGB and thermal) data for visual tracking. In particular, the tracked object is represented with a graph with image patches as nodes. This graph is dynamically learned from two aspects. First, the graph affinity (i.e., graph structure and edge weights) that indicates the appearance compatibility of two neighboring nodes is optimized based on the weighted sparse representation, in which the modality weight is introduced to leverage RGB and thermal information adaptively. Second, each node weight that indicates how likely it belongs to the foreground is propagated from others along with graph affinity. The optimized patch weights are then imposed on the extracted RGB and thermal features, and the target object is finally located by adopting the structured SVM algorithm. Moreover, we also contribute a comprehensive dataset for RGB-T tracking purpose. Comparing with existing ones, the new dataset has the following advantages: 1) Its size is sufficiently large for large-scale performance evaluation (total frame number: 210K, maximum frames per video pair: 8K). 2) The alignment between RGB-T video pairs is highly accurate, which does not need pre- and post-processing. 3) The occlusion levels are annotated for analyzing the occlusion-sensitive performance of different methods. Extensive experiments on both public and newly created datasets demonstrate the effectiveness of the proposed tracker against several state-of-the-art tracking methods.

KEYWORDS

Tracking, thermal infrared, dataset, sparse representation

1 INTRODUCTION

Visual tracking is to estimate the states of target object in subsequent frames, given the initial ground-truth bounding box. It has drawn a lot of attention due to its wide range of practical applications, such as video surveillance, self-driving cars, and robotics. Despite many recent breakthroughs in visual tracking, it still faces many challenging problems especially tracking target objects in various environmental conditions (e.g., low illumination, rain, haze and smog, etc.), which significantly limit the imaging quality of visible spectrum.

Integrating visible and thermal (called RGB-T in this paper) spectrum data has been proven to be effective in boosting tracking performance, and also allow tracking the target objects in day and night [15]. Visible and thermal information complement each other and contribute to visual tracking in different aspects. On one hand, thermal infrared camera can capture infrared radiation (0.75-13μm) emitted by subjects with a temperature above absolute zero. Thus they are insensitive to lighting conditions and have a strong ability to penetrate haze and smog. These kind of sensors, therefore, are more effective in capturing objects than visible spectrum cameras under poor lighting conditions and bad weathers. On the other hand, visible spectrum cameras are more effective in separating two moving subjects, which are crossing or moving side (called thermal crossover [31, 34]).

Many efforts have been devoted to RGB-T object tracking. Such as, Conaire *et al.* [5, 6] and Cvejic *et al.* [7] adopt some simple weight schemes to fuse RGB and thermal data adaptively, which might easily fail in many challenging scenarios. Bunyak *et al.* [3] employ thermal information to assist RGB tracking as thermal information is less sensitive to illumination variations and shadows. But it also limits the tracking performance when thermal information is unreliable. Recently, sparse representation, which is directly

*Corresponding author.

MM'17, October 23–27, 2017, Mountain View, CA, USA.
 ISBN 978-1-4503-4906-2/17/10…$15.00
DOI: https://doi.org/10.1145/3123266.3123289

related to compressed sensing [11], has been successfully applied to RGB-T tracking [28, 30, 34, 42] due to its capability of suppressing noises. All these methods adopt raw gray values as feature representation and their weakness is usually revealed when dealing with complex challenging scenarios.

In this paper, we aim to learn a robust object representation for RGB-T tracking. In particular, we partition each target bounding box into a set of non-overlapping patches, which are described with RGB and thermal features. However, background information is inevitably included in the bounding box and likely results in model drifting. Thus, the bounding box cannot represent the target object well, and we associate a weight to each patch, which reflects its importance to describe the object. We concatenate all patch features within the target bounding box with their corresponding weights to convey structural information of the object while suppressing the pollution of background. The structured SVM algorithm [41] is then adopted to perform object tracking.

In general, the patch weight computation is performed via a semi-supervised learning algorithm, e.g., manifold ranking [50] and random walk [21]. And how to learn an optimal affinity matrix is essential. Most of methods construct a fixed-structure graph (e.g., 8-neighbor graph where each node is only connected with its 8 neighbor nodes), and thus neglect global intrinsic relationship of patches. While Li *et al.* [29] propose to learn a dynamic graph with the low-rank and sparse representations as the graph affinity to capture the global subspace structure of patches. In fact, although this kind of affinity is somehow valid, the meaning of which is already not the same as the original definition [16].

Motivated by above observations, we propose a novel graph model, which called weighted sparse representation regularized graph, to adaptively employ RGB and thermal data for learning weights. We take patches as graph nodes, and pursue a joint sparse representation [25, 34] with patch feature matrix as input. Note that other constraints on the representation matrix, such as low rank [33], can also be incorporated in our model. We only employ the sparse constraints in this paper for an emphasis on efficient performance. To deal with occasional perturbation or malfunction of individual sources, we assign a weight for each modality to represent the reliability, which allows our method to integrate different spectrum data adaptively. Instead of directly using sparse representations, we learn a more meaningful graph affinity with the assumption that patches are more likely located in the same class (the target object or the background) if their sparse representations have a small distance [16]. Based on the graph affinity, the patch weights are computed when giving some initial weights to patches [21]. It is worth noting that we jointly optimize the modality weights, the sparse representations, and the graph (including the structure, the edge weights and the node weights) by a designed efficient ADMM (Alternation Direction Method of Multipliers) algorithm [2].

In addition, we build a new comprehensive dataset for RGB-T tracking purpose, and plan to open it to public. Compared with other existing RGB-T datasets [1, 28, 40], the new one has sufficiently big size, highly-accurate alignment between RGB-T sequence pairs, and the annotated occlusion levels.

To our best knowledge, we are the first to learn robust RGB-T object features via graph learning for visual tracking. In this paper, the following contributions have been made for RGB-T tracking and its related applications. First, a novel patch-based graph model is proposed to learn robust object feature presentation for RGB-T tracking. In particular, the graph is optimized via weighted sparse representations that utilize multispectral information adaptively according to their reliabilities. Second, an ADMM algorithm is designed to jointly and efficiently optimize the modality weights, the sparse representations, and the graph with the structure, edge weights and node weights. Third, a comprehensive RGB-T tracking dataset is created for large-scale performance evaluation of different tracking algorithms. This dataset [1] and the source code will be available online for free academic usage and accessible reproducible research.

2 RELATED WORK

Here we discuss the most related visual tracking works that assign weights to different pixels (or patches) in the bounding box to suppress background effects (See [44] and [27] for recent surveys). Comaniciu *et al.* [4] assume that pixels far away from a box center should be less important, and thus assign smaller weights to boundary pixels via the kernel-based method during the histogram construction. The similar assumption is also considered by He *et al.* [18]. Based on their assumption, they construct a locality sensitive histogram at each pixel location, which takes every pixel into account. But the pixels that are far away from the center could be neglected due to the very small weights assigned. These methods, however, might be disturbed when a target object has a complicated shape or is occluded. Some other methods [12, 48] integrate segmentation results (i.e., assigning 0 or 1 to each pixel) into tracking to alleviate the effects of background, but these algorithms are sensitive to segmentation results.

Recently, associating each patch with a weight has been proven to be an effective way for suppressing the background effects in visual tracking [21, 29]. Kim *et al.* [21] employ a random walk restart algorithm on the fixed-structure graph with patches as nodes to compute patch weights within target object bounding box. The above fixed-structure graph is constructed via only local cues, neglecting global cues that are important for exploring the intrinsic relationship among patches. While Li *et al.* [29] employ the low-rank and sparse representation to learn a dynamic graph with global considerations.

RGB-T tracking receives more and more attention in the community with the popularity of thermal infrared sensors [6]. Conaire *et al.* [5] propose an automated surveillance framework that can efficiently combine visible and thermal features for robust tracking, and Cvejic *et al.* [7] investigate the impact of pixel-level fusion of videos from grayscale-thermal surveillance cameras. Leykin *et al.* [26] propose a pedestrian tracker in the particle filter framework, which represents the target object as a multi-modal distribution with the changing number of modalities for both grayscale and thermal input.

[1] RGB-T tracking dataset's webpage: http://chenglongli.cn/people/lcl/confs.html.

Recently, Sparse representation has been successfully applied to RGB-T tracking [28, 30, 34, 42]. Wu *et al.* [42] concatenate the image patches from RGB and thermal sources, and then sparsely represent each sample in the target template space. Liu *et al.* [34] fuse the resultant tracking results using min operation on the sparse representation coefficients calculated on both RGB and thermal modalities. These methods may limit the tracking performance in dealing with occasional perturbation or malfunction of individual sources as available spectrums contribute equally. Li *et al.* [28, 30] introduce a modality weight for each source to represent the imaging quality, and combine with the sparse representation in Bayesian filtering framework to perform object tracking.

There are several RGB-T video datasets for the various vision tasks. For example, OSU Color-Thermal dataset [10] contains six RGB-T video sequence pairs recorded from two different locations with only people moving, which is obviously not sufficient to evaluate tracking algorithms. Other two RGB-T datasets are collected by Torabi *et al.* [40] and Bilodeau *et al.* [1]. Most of them suffer from their limited size, low diversity, and high bias. Li *et al.* [28] release a dataset with 50 RGB-T video pairs, which are also not enough for large-scale performance evaluation. Our work addresses these issues and creates a reasonable size and more challenging RGB-T video dataset.

3 THE PROPOSED TRACKER

In this paper, the object tracking is performed based on the conventional tracking-by-detection algorithm, Struck [17]. It is worth noting that other tracking-by-detection algorithms can also be adopted.

3.1 Tracking via Structured SVM

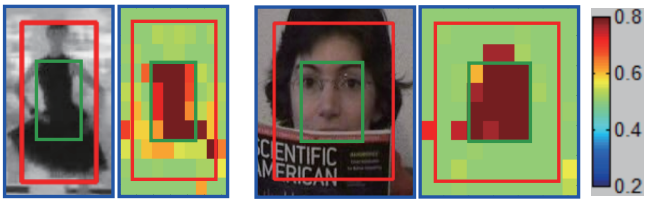
Hare *et al.* [17] employ the structured SVM framework [41] (called Struck) for tracking and achieve promising tracking performance. Struck selects the optimal target bounding box y_t^* in the t-th frame by maximizing a classification score.

Let $\Psi(\mathbf{x}_t, y_t)$ denote the target object descriptor representing a bounding box y_t in the t-th frame to maximize a classifier score $\langle \mathbf{h}_{t-1}, \Psi(\mathbf{x}_t, y_t)\rangle$, where $\mathbf{h}_{t-1}$ is the normal vector of a decision plane of $(t-1)$-th frame:

$$y_t^* = \arg\max_{y_t} \langle \mathbf{h}_{t-1}, \Psi(\mathbf{x}_t, y_t)\rangle \tag{1}$$

Instead of using binary-labeled samples, Struck employs a structured sample that consists of a target bounding box and nearby boxes in the same frame to prevent the labeling ambiguity in training the classifier. Specifically, it constraints that the confidence score of a target bounding box is larger than the confidence scores of nearby boxes, which are determined by a margin (the overlap ratio between two boxes). By this way, Struck can reduce adverse effects of false labelling.

In this work, we combine the object representations of multiple modalities with Struck to achieve robust RGB-T tracking. Given the bounding box of the target object in previous frame $t-1$, we first set a searching window in current frame t, and sample a set of candidates within the searching window. The tracking result is then predicted by selecting the candidate with maximal classification score:



(a) Deformation (b) Partial occlusion

Figure 1: Two samples for showing the optimized patch weights are shown in (a) and (b). The original, shrunk and expanded bounding boxes are represented by the red, green and blue colors, respectively. The optimized patch weights are also shown for clarity, in which the hotter color indicates the larger weight.

$$y_t^* = \arg\max_{y_t} (\nu\langle \mathbf{h}_{t-1}, \Psi(\hat{\mathbf{x}}_t, y_t)\rangle + (1-\nu)\langle \mathbf{h}_0, \Psi(\hat{\mathbf{x}}_t, y_t)\rangle), \tag{2}$$

where $\mathbf{h}_0$ is learned in initial frame, which prevents it from learning drastic appearance changes [36]. ν is a balance parameter. To prevent the effects of unreliable tracking results, we update the classifier only when the confidence score of tracking result [21] is larger than a threshold ζ.

During RGB-T tracking, we construct a target pyramid around the estimated translation location for scale estimation [35]. Let $W \times H$ be the target size in a test frame and N indicates the number of scales $B = \{a^{\bar{n}} | \bar{n} = \lfloor -\frac{N-1}{2} \rceil, \lfloor -\frac{N-3}{2} \rceil, ..., \lfloor \frac{N-1}{2} \rceil\}$. For each $b \in B$, we extract an image region of size $bW \times bH$ centered around the estimated location and compute its classification score $y_{t,b}^*$. Then, we uniformly resize all image regions with the size $W \times H$, and the optimal scale b of target can be estimated by maximizing the classification scores of all resized image regions:

$$b^* = \arg\max_b \{y_{t,b}^*, b \in B\}. \tag{3}$$

3.2 Feature Representation

In RGB-T tracking, it is important to construct a robust descriptor $\Psi(\hat{\mathbf{x}}, y)$ of the bounding box y. Therefore, as discussed in Section 1, we assign a weight $\hat{s}_i$ for each patch to suppress the background effects, and also assign a weight r^m for each modality to integrate different source data adaptively. Combining these weights with the corresponding patch feature descriptor $\mathbf{x}_i^m$ forms the final feature representation of the bounding box:

$$\begin{aligned}\Psi(\hat{\mathbf{x}}_t, y_t) = [&r_{t-1}^1 \hat{s}_{t-1,1} \mathbf{x}_{t,1}^1, ..., r_{t-1}^1 \hat{s}_{t-1,n} \mathbf{x}_{t,n}^1, \\ &..., r_{t-1}^M \hat{s}_{t-1,1} \mathbf{x}_{t,1}^M, ..., r_{t-1}^M \hat{s}_{t-1,n} \mathbf{x}_{t,n}^M]^T.\end{aligned} \tag{4}$$

From (4), we can see that $\hat{s}_i$ and r^m plays a critical role in the target feature representation, and we present the details of their computation in next section.

4 WEIGHTED SPARSE REPRESENTATION REGULARIZED GRAPH LEARNING

In this section, we introduce the proposed graph-based algorithm that infers the patch weight and the modality weight used in our tracker.

4.1 Formulation

Each bounding box of the target object is partitioned into n non-overlapping patches, and a set of low-level appearance features are extracted and further combined into a d-dimensional feature vector $\mathbf{x}_i^m$ for characterizing the i-th patch in the m-th modality. All the feature descriptors of n patches in one bounding box form the data matrix $\mathbf{X}^m = \{\mathbf{x}_1^m, \mathbf{x}_2^m, ..., \mathbf{x}_n^m\} \in \mathbb{R}^{d\times n}$, where the m indicates the index of the modality with the range between 1 and M. Herein, we discuss the general case for the scalability, and RGB-T data used in this paper is the special one with $M = 2$.

Weighted sparse representation. We take the above patches as graph nodes and dynamically learn the graph according to the intrinsic relationship of patches instead of using fixed-structure graph in conventional methods [21, 50]. Motivated by the sparse clustering algorithms [13, 47], we assume the foreground or background patches are in the same sparse subspace [13], and thus each patch descriptor can be sparsely self-represented by a linear combination of remaining patches: $\mathbf{X}^m = \mathbf{X}^m\mathbf{Z}^m$, where $\mathbf{Z}^m \in \mathbb{R}^{n\times n}$ is sparse representation coefficient matrix. Sparse constraints can automatically select most informative neighbors for each patch (higher-order relationship), making the graph more powerful and discriminative [47]. Considering the patches are often disturbed by noises and/or corruptions, we introduce a noise matrix to improve the robustness. The joint sparse representation with the convex relaxation [47] for all modalities can be formulated as:

$$\min_{\mathbf{Z},\mathbf{E}} \sum_{m=1}^{M} \|\mathbf{X}^m - \mathbf{X}^m\mathbf{Z}^m - \mathbf{E}^m\|_F^2 + \lambda\|\mathbf{E}^m\|_{2,1} + \gamma\|\mathbf{Z}\|_{2,1}, \quad (5)$$

where $||\cdot||_F$ and $||\cdot||_{2,1}$ denote the Frobenius norm and the $l_{2,1}$ norm of a matrix, respectively. λ and γ are the balanced parameters. $\mathbf{Z} = [\mathbf{Z}^1; ...; \mathbf{Z}^M] \in \mathbb{R}^{Mn\times n}$ is the joint sparse representation coefficients matrix, and $\|\mathbf{Z}\|_{2,1}$ encourages each patch to share the same pattern across different modalities. $\mathbf{E}^m \in \mathbb{R}^{d\times n}$ denotes the noise matrix, and $\|\mathbf{E}^m\|_{2,1}$ makes it as the sparse sample-specific corruptions, i.e., some patches are corrupted and others are clean. It is worth noting that other constraints on the representation matrix, such as low rank [33], can be also incorporated in (5), and we only employ the sparse constraints in this paper with an emphasis on efficient performance.

In (5), different modalities contribute equally, but usually have different imaging qualities in real-life scenarios. Therefore, we assign a weight for each modality to represent the reliability to deal with occasional perturbation or malfunction of individual sources, which allows our method to integrate different spectrum data adaptively [28]. We integrate these modality weights in (5), and have

$$\begin{aligned}\min_{\mathbf{Z},\mathbf{E},\mathbf{r}} &\sum_{m=1}^{M} \left(\frac{(r^m)^2}{2}\|\mathbf{X}^m - \mathbf{X}^m\mathbf{Z}^m - \mathbf{E}^m\|_F^2 + \lambda\|\mathbf{E}^m\|_{2,1}\right) \\ &+ \gamma\|\mathbf{Z}\|_{2,1} + \Gamma\|\mathbf{1} - \mathbf{r}\|_F^2,\end{aligned} \quad (6)$$

where $\mathbf{r} = [r^1, ..., r^M]^T$ is the modality weighting vector, in which r^m is the modality weight in the m-th modality. In general, the reconstruction error can measure all patches by how well it could be sparsely reconstructed from the other patches. Therefore, the qualities of different modalities can be reflected by their respective reconstruction errors. From the first term in (6) we can see that our method places larger weights on those modalities which have smaller reconstruction errors, resulting in a quality-aware weight optimization. The last term of (6) is the regularization of r^m, which avoids a degenerate solution of r^m while allowing them to be specified independently. Γ is an adaptive parameter, which is determined after the first iteration, see the **supplementary file** for details.

Regularized graph model. Most of methods utilize the representation coefficient matrix to define the graph affinity by $\frac{|\mathbf{Z}|+|\mathbf{Z}^T|}{2}$. Although this kind of affinity is somehow valid, its meaning is already different from the original definition [16]. Recall that we assume the patch descriptor should have a larger probability to be in the same cluster if their representations have a smaller distance. Therefore, instead of directly using sparse representations, we learn a more meaningful graph affinity $\mathbf{A}$ by the following constraints:

$$\min_{\mathbf{A}} \delta \sum_{i,j=1}^{n} \|\mathbf{Z}_i - \mathbf{Z}_j\|_F^2\mathbf{A}_{ij} + \frac{\omega}{2}\|\mathbf{A}\|_F^2, s.t. \mathbf{A}^T\mathbf{1} = \mathbf{1}, \mathbf{A} \geq 0, \quad (7)$$

where δ, ω are the balanced parameters, and $\mathbf{1}$ denotes a unit vector. $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the desired affinity matrix, $\mathbf{A}_{ij}$ reflects the probability of the patch i and j from the same class based on the distance between their joint sparse representations $\mathbf{Z}_i$ and $\mathbf{Z}_j$ across all modalities. The constraints $\mathbf{A}^T\mathbf{1} = \mathbf{1}$ and $\mathbf{A} \geq 0$ are to guarantee the probability property of $\mathbf{A}_i$. The last term is to avoid overfitting of $\mathbf{A}$.

Given the graph affinity $\mathbf{A}$, we can compute the patch node weights in a semi-supervised way. Let $\mathbf{q} = \{q_1, q_2, ..., q_n\}^T$ be an initial weight vector, in which $q_i = 1$ if q_i is a foreground patch, and $q_i = 0$ is a background patch. $\mathbf{q}$ is computed by the initial ground-truth (for first frame) or tracking results (for subsequent frames) as follows: for i-th patch, $q_i = 1$ if it belongs to the shrunk region of bounding box and the remaining patches are 0. Figure 1 shows the details. Similar to the PageRank and spectral clustering algorithm [37], the patch weights $\mathbf{s}$ can be calculated as follows:

$$\min_{\mathbf{s}} \alpha \sum_{i,j=1}^{n} (s_i - s_j)^2\mathbf{A}_{ij} + \beta\|\mathbf{s} - \mathbf{q}\|_F^2, \quad (8)$$

where α and β are the balanced parameters. The first term is the smoothness constraint and the second term is the fitting constraint.

In this paper, we aim to jointly optimize the modality weights, the sparse representations, and the graph (including the structure,

the edge weights and the node weights) for boosting their respective performance. Therefore, the final formulation of the proposed model can be written by combining (6), (7) and (8):

$$\begin{aligned}\min_{\substack{\mathbf{Z},\mathbf{E},\mathbf{r},\mathbf{s},\mathbf{A},\\ \mathbf{A}^T\mathbf{1}=\mathbf{1},\mathbf{A}\geq 0}} &\sum_{m=1}^{M}\left(\frac{(r^m)^2}{2}\|\mathbf{X}^m-\mathbf{X}^m\mathbf{Z}^m-\mathbf{E}^m\|_F^2+\lambda\|\mathbf{E}^m\|_{2,1}\right)\\ &+\gamma\|\mathbf{Z}\|_{2,1}+\delta\sum_{i,j=1}^{n}\|\mathbf{Z}_i-\mathbf{Z}_j\|_F^2\mathbf{A}_{ij}+\alpha\sum_{i,j=1}^{n}(s_i-s_j)^2\mathbf{A}_{ij}\\ &+\beta\|\mathbf{s}-\mathbf{q}\|_F^2+\Gamma\|\mathbf{1}-\mathbf{r}\|_F^2+\frac{\omega}{2}\|\mathbf{A}\|_F^2.\end{aligned} \tag{9}$$

Although (9) seems complex, as demonstrated in the experiments, its parameters are easy to adjust, and the tracking performance is insensitive to parameter variations. The final weight of the i-th patch is computed by $\hat{s}_i=\frac{1}{(1+\exp(-\sigma s_i))}$, where the parameter σ is fixed to be 37 in this work.

4.2 Optimization

Although the variables of the (9) are not joint convex, the subproblem of each variable with others fixed is convex and has a closed-form solution. The ADMM (alternating direction method of multipliers) algorithm [2] is efficient and effective solver of the problems like (9). To apply ADMM to our problem, we need make our objective function separable. Therefore, the auxiliary variables $\mathbf{P}^m\in\mathbb{R}^{n\times n}$ and $\mathbf{Q}^m\in\mathbb{R}^{n\times n}$ are introduced to make (9) separable:

$$\begin{aligned}\min_{\mathbf{Z},\mathbf{P},\mathbf{Q},\mathbf{E},\mathbf{r},\mathbf{s},\mathbf{A}} &\sum_{m=1}^{M}\left(\frac{(r^m)^2}{2}\|\mathbf{X}^m-\mathbf{X}^m\mathbf{Z}^m-\mathbf{E}^m\|_F^2+\lambda\|\mathbf{E}^m\|_{2,1}\right)\\ &+\gamma\|\mathbf{Q}\|_{2,1}+\delta\sum_{i,j=1}^{n}\|\mathbf{P}_i-\mathbf{P}_j\|_F^2\mathbf{A}_{ij}+\alpha\sum_{i,j=1}^{n}(s_i-s_j)^2\mathbf{A}_{ij}\\ &+\beta\|\mathbf{s}-\mathbf{q}\|_F^2+\Gamma\|\mathbf{1}-\mathbf{r}\|_F^2+\frac{\omega}{2}\|\mathbf{A}\|_F^2,\\ &s.t.\mathbf{Z}^m=\mathbf{P}^m,\mathbf{Z}^m=\mathbf{Q}^m,\mathbf{A}^T\mathbf{1}=\mathbf{1},\mathbf{A}\geq 0,\end{aligned} \tag{10}$$

where $\mathbf{P}=[\mathbf{P}^1;\ldots;\mathbf{P}^M]$ and $\mathbf{Q}=[\mathbf{Q}^1;\ldots;\mathbf{Q}^M]$.
The augmented Lagrange function of (10) is

$$\begin{aligned}&\mathcal{L}_{\{\mathbf{A}\mathbf{1}=\mathbf{1},\mathbf{A}\geq 0,\mathbf{s}\geq 0\}}(\mathbf{Z},\mathbf{P},\mathbf{Q},\mathbf{E},\mathbf{r},\mathbf{s},\mathbf{A})\\ &=\sum_{m=1}^{M}\left(\frac{(r^m)^2}{2}\|\mathbf{X}^m-\mathbf{X}^m\mathbf{Z}^m-\mathbf{E}^m\|_F^2+\lambda\|\mathbf{E}^m\|_{2,1}\right)+\gamma\|\mathbf{Q}\|_{2,1}\\ &+\delta\sum_{i,j=1}^{n}\|\mathbf{P}_i-\mathbf{P}_j\|_F^2\mathbf{A}_{ij}+\alpha\sum_{i,j=1}^{n}(s_i-s_j)^2\mathbf{A}_{ij}+\beta\|\mathbf{s}-\mathbf{q}\|_F^2\\ &+\Gamma\|\mathbf{1}-\mathbf{r}\|_F^2+\frac{\omega}{2}\|\mathbf{A}\|_F^2+f(\mathbf{Z},\mathbf{P},\mathbf{Q},\mathbf{Y}_1,\mathbf{Y}_2,\mu),\end{aligned} \tag{11}$$

where $\mu>0$ is the penalty parameter, and $f(\mathbf{Z},\mathbf{P},\mathbf{Q},\mathbf{Y}_1,\mathbf{Y}_2,\mu)=-\sum_{m=1}^{M}(\frac{1}{2\mu}(\|\mathbf{Y}_1^m\|_F^2+\|\mathbf{Y}_2^m\|_F^2)+\frac{\mu}{2}(\|\mathbf{Z}^m-\mathbf{P}^m+\mathbf{Y}_1^m/\mu\|_F^2+\|\mathbf{Z}^m-\mathbf{Q}^m+\mathbf{Y}_2^m/\mu\|_F^2))$. $\mathbf{Y}_1^m$ and $\mathbf{Y}_2^m$ are the Lagrangian multipliers.

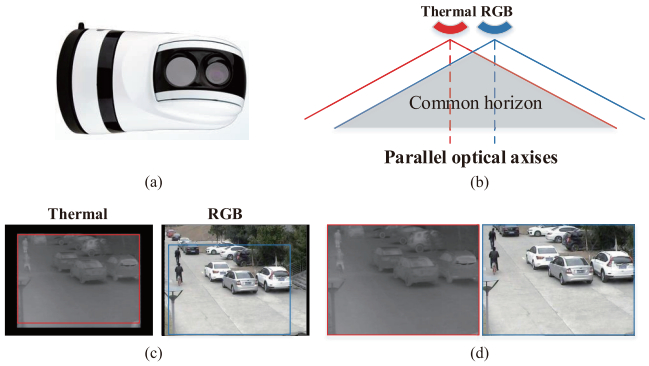


Figure 2: The mechanism of our imaging system for capturing the RGB-T video sequences. (a) Illumination of the imaging hardware. (b) The optic axes of two cameras are aligned as parallel using the collimator. (c) Sample frame pairs captured by our system. The annotated bounding boxes indicate the common horizon, in which the colors denote the correspondences to the cameras. (d) Cropped image regions that are highly aligned.

ADMM alternatively updates one variable by minimizing $\mathcal{L}$ with other variables fixed. Besides the Lagrangian multipliers, the updating schemes of $(k+1)$-th iteration could be solved as follows:

$$\begin{aligned}\mathbf{Z}^{m,k+1}&=((\mathbf{X}^m)^T\mathbf{X}^m+2\mu\mathbf{I}/(r^{m,k})^2)^{-1}((\mathbf{X}^m)^T\\ &(\mathbf{X}^m-\mathbf{E}^{m,k})+\frac{\mu(\mathbf{P}^{m,k}+\mathbf{Q}^{m,k})-\mathbf{Y}_1^{m,k}-\mathbf{Y}_2^{m,k}}{(r^{m,k})^2}),\\ \mathbf{P}^{k+1}&=(\mu\mathbf{Z}^{k+1}+\mathbf{Y}_1^k)(4\delta{\mathbf{L}_\mathbf{A}}^k+\mu_k\mathbf{I})^{-1},\\ \mathbf{Q}^{k+1}&=\mathbb{S}_{\frac{\gamma}{\mu_k}}(\mathbf{Z}^{k+1}+\mathbf{Y}_2^k/\mu_k),\\ \mathbf{E}^{m,k+1}&=\mathbb{S}_{\frac{\lambda}{(r^{m,k})^2}}(\mathbf{X}^m-\mathbf{X}^m\mathbf{Z}^{m,k+1}),\\ r^{m,k+1}&=\frac{2\Gamma}{\|\mathbf{X}^m-\mathbf{X}^m\mathbf{Z}^{m,k+1}-\mathbf{E}^{m,k+1}\|_F^2+2\Gamma},\\ \mathbf{A}_i^{k+1}&=(\frac{1+\sum_{j=1}^{\xi}\hat{\mathbf{u}}}{\xi}\mathbf{1}-\hat{\mathbf{u}}_{ij})_+,\ i=1,2,\ldots,n,\\ \mathbf{s}^{k+1}&=(2\alpha{\mathbf{L}_\mathbf{A}}^{k+1}+\beta\mathbf{I})^{-1}\beta\mathbf{q},\end{aligned} \tag{12}$$

where $\mathbf{L}_{\mathbf{A}^k}$ is the Laplacian matrix of $\mathbf{A}^k$: $\mathbf{L}_{\mathbf{A}^k}=\mathbf{D}_{\mathbf{A}^k}-\mathbf{A}^k$, where the degree matrix $\mathbf{D}_{\mathbf{A}ii}=\sum_{j=1}^{n}\mathbf{A}_{ij}$. $\mathbb{S}_{\frac{\gamma}{\mu_k}}(\mathbf{Z}^{m,k+1}+\mathbf{Y}_2^{m,k}/\mu_k)$ is the $l_{2,1}$ minimization operator [33] on $(\mathbf{Z}^{m,k+1}+\mathbf{Y}_2^{m,k}/\mu_k)$ with parameter $\frac{\gamma}{\mu_k}$. $\mathbf{u}_i\in\mathbb{R}^{n\times 1}$ is a vector whose j-th element is $\mathbf{u}_{ij}=\frac{\delta\|\mathbf{P}_i-\mathbf{P}_j\|_F^2+\alpha(s_i-s_j)^2}{\omega}$. The elements of $\hat{\mathbf{u}}\in\mathbb{R}^{\mathbf{n}\times\mathbf{1}}$ are those of $\mathbf{u}$ but with an ascending order, and the parameter $\xi\in\{1,\ldots,n\}$ is introduced to control the number of nearest neighbors of each patch. Please refer to the **supplementary file** for details. Since each subproblem of (11) is convex, we can guarantee that the limit point by our algorithm satisfies the Nash equilibrium conditions [46].

Table 1: List of the attributes annotated to RGBT210.

Attr	Description
NO	No Occlusion - the target is not occluded.
PO	Partial Occlusion - the target object is partially occluded.
HO	Heavy Occlusion - the target object is heavy occluded (over 80%).
LI	Low Illumination - the illumination in the target region is low.
LR	Low Resolution - the resolution in the target region is low.
TC	Thermal Crossover - the target has similar temperature with other objects or background surroundings.
DEF	Deformation - non-rigid object deformation.
FM	Fast Motion - the motion of the ground truth between two adjacent frames is larger than 20 pixels.
SV	Scale Variation - the ratio of the first bounding box and the current bounding box is out of the range [0.5,1].
MB	Motion Blur - the target object motion results in the blur image information.
CM	Camera Moving - the target object is captured by moving camera.
BC	Background Clutter - the background information which includes the target object is messy.

5 RGB-T TRACKING DATASET

This section introduces a new RGB-Thermal object tracking dataset (called RGBT210 in this paper).

5.1 Platform

Our imaging hardware consists of a turnable platform, a thermal infrared imager (DLS-H37DM-A) and a CCD camera (SONY EXView HAD CC), as shown in Figure 2 (a). In particular, the two cameras have same imaging parameters, and their optical axises are aligned as parallel by the collimator, as shown in Figure 2 (b). The above setup makes the common horizon of these two cameras aligned in pixel level. Figure 2 (c) and (d) show the details.

5.2 Annotation

For large scale performance evaluation of different RGB-T tracking algorithms, we collect 210 RGB-T video pairs, each containing a RGB video and a thermal video. We annotate each frame with a minimum bounding box covering the targets using more reliable source data. All annotations are done by a full-time annotator to guarantee consistency. In addition, we also annotate the attributes for each video sequence for the attribute-sensitive performance analysis, as shown in Table 1. Due to space limitation, we present the attribute distribution over the entire dataset and some sample pairs with ground truth and attribute annotations in the **supplementary file**.

5.3 Advantages over existing Datasets

Table 2 provides summary of existing tracking datasets, including RGB, Thermal, and RGB-T. The RGB datasets [22, 23, 27, 32, 43, 44] only provide RGB image sequences. Since RGB spectrum is sensitive to lighting conditions and could be easily affected by bad weathers, such as rain, smog and fog, these datasets are limited to "good" lighting and environmental conditions. Since the thermal datasets [9, 14, 38, 45] only include thermal videos, its weakness is revealed when thermal crossover occurs. Therefore, the RGB-T datasets [10, 28, 40] are necessary for addressing above issues. And fusion of these two data enables long-term object tracking in day and night.

Our dataset is classified as a RGB-T one as it provides aligned RGB and thermal video pairs. Compared to other existing datasets, our dataset has the following advantages: 1) it includes a large amount of annotated highly-accurate frames (total frames: 210K, maximum frames per sequence pair: 8K), which allows trackers to perform the large-scale performance evaluation. 2) Due to the superior imaging mechanism, its alignment across modalities is more accurate, and does not require any pre- and post-processing (e.g., stereo matching [24, 28] and color correction [20]). 3) Its occlusion levels, including no, partial and heavy occlusions, are annotated for occlusion-sensitive evaluation of different algorithms. 4) Since the imaging parameters of RGB and thermal cameras in our platform are the same and their optical axises are parallel, its videos can be captured by both static and moving cameras while keeping the alignment accurate.

Table 2: Comparison of RGBT210 against other tracking datasets.

Type		Properties					
		# total frames	max frames per video	occlusion annotation	attribute annotation	moving camera	publish year
RGB	OTB50 [43]	29.4K	3.8K	✓	✓	✓	2013
	OTB100 [44]	59K	3.8K	✓	✓	✓	2015
	VOT2014 [23]	10.3K	1.2K	✓	✓	✓	2014
	VOT2015 [22]	21.8K	1.5K	✓	✓	✓	2015
	Temple-Color [32]	55.3K	3.8K	✓	✓	✓	2015
	NUS-PRO [27]	135.8K	5K	✓	✓	✓	2016
Thermal	OSU-T [9]	0.2K	-	✓	✓	-	2005
	ASL-TID [38]	4.3k	-	-	-	-	2014
	TIV [45]	63K	5.9K	-	-	-	2014
	LTIR [14]	11.2K	1.4K	✓	✓	✓	2015
RGB-T	OSU-CT [10]	17K	2K	✓	✓	-	2007
	LITIV [40]	6.3K	1.2K	-	-	-	2012
	GTOT [28]	15.8K	0.7K	✓	✓	✓	2016
	Ours	210K	8K	✓	✓	✓	2017

6 PERFORMANCE EVALUATION

The experiments are executed on a PC with an Intel Core i7 @3.4 GHz CPU and 16GB RAM. All the codes are implemented in C++ without any code optimization. The proposed tracker performs at about 5 FPS (frames per second).

6.1 Settings

We present implementation details of the experiments, including parameter settings of the proposed approach, two datasets, baseline trackers, and evaluation metrics. The source code, the new dataset, the baseline trackers and the evaluation results will be available online for free academic usage and accessible reproducible research.

Parameters. For fair comparisons, we fix all parameters and other settings in all experiments. We partition the bounding box into 64 non-overlapping patches to balance accuracy and efficiency, and extract color and gradient histograms for each patch. The dimension of gradient and each color channel is set to be 8. To improve efficiency, each frame is scaled so that the minimum side length of bounding box is 32 pixels, and the side length of a searching window is fixed to be $2\sqrt{\bar{W}\bar{H}}$, where $\bar{W}$ and $\bar{H}$ are the width and height of the scaled bounding box, respectively.

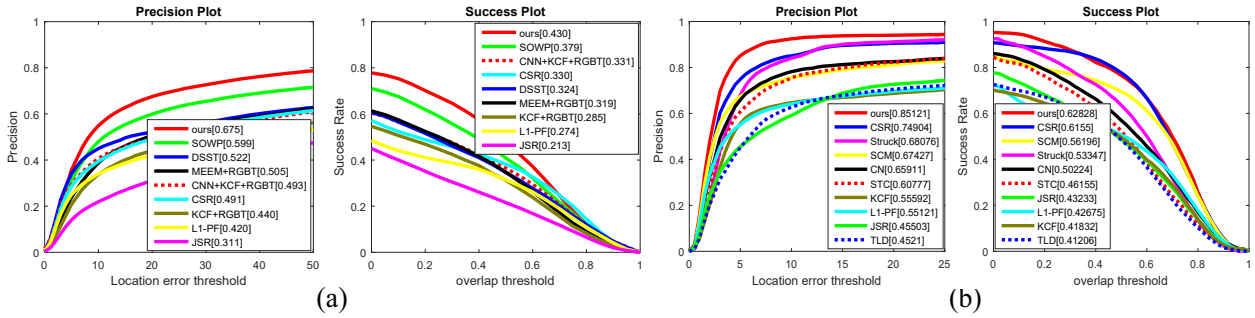


Figure 3: (a) and (b) denote the evaluation results on our RGBT210 dataset and the public GTOT benchmark, respectively. The representative score of PR/SR is presented in the legend.

Although (9) seems complex, its parameters are easy to tune, and the tracking performance is insensitive to parameter variations. On one hand, to simplify our parameters, we let $\lambda = \gamma$, and empirically set them to 0.1. The smoothness parameter α and the fitting parameter β of $\mathbf{s}$ are correlated. And we set $\alpha = 100\beta = 15$. The parameters of $\mathbf{A}$ are set to be $\{\delta, \omega\} = \{11, 1\}$. All parameters are optimal by varying them on a certain scope. On the other hand, when we slightly adjust the parameters, tracking performance only changes a little, we find the results are slightly different from current settings (show in Table 5). In Struck, we empirically set $\{\nu, \zeta\} = \{0.67, 0.35\}$.

Datasets. *RGBT210* is a new dataset created by us, which includes 210 highly-aligned RGB-T video pairs with about 210K frames in total. More details are presented in Section 5. This dataset provides a comprehensive platform for large-scale evaluation.

To further demonstrate the effectiveness of the proposed method, we also conduct the comparison on the public dataset GTOT [28]. GTOT includes 50 aligned RGB-T video pairs with about 15K frames in total. And each frame pair is annotated with ground truth bounding box. It also provides the fine-grained annotations (i.e., 7 attributes) to analyze the attribute-sensitive performance of other trackers.

Baselines. On GTOT, we use the RGB-T baseline trackers implemented in [28] to evaluate the proposed approach. In particular, some of these RGB-T trackers first concatenate RGB and thermal features into a single vector, and then employ existing RGB trackers to perform object tracking.

Following this strategy, we implement several new RGB-T baselines for comprehensive evaluation, including CNN+KCF+RGBT , KCF [19]+RGBT , MEEM [49]+RGBT. Here, CNN+KCF+RGBT is implemented as follows: we first utilize two-stream CNN to extract the specific features for different modalities, where one CNN is applied to process the RGB stream and the another is used to handle the thermal stream. The CNN of both two streams has similar structure as a general deep CNN for image classification (VGG-16 [39], which directly takes individual object patches as network inputs followed by 13 convolutional layers and 4 pooling layers). We remove the fully connected layers for tracking purpose. Then, we directly concatenate the features of different modalities into a vector, followed by KCF to perform tracking.

Additionally, three RGB-T trackers are added in our platform, including L1-PF [42], JSR [34], and CSR [28]. We also evaluate two state-of-the-art RGB trackers: DSST [8] (correlation filter-based tracker) and SOWP [21] (SVM-based tracker).

Metrics. We utilize two widely used metrics, precision rate (PR) and success rate (SR), to evaluate RGB-T tracking performance. PR is the percentage of frames whose output location is within the given threshold distance of ground truth. As in other benchmarks [43, 44], we set the threshold to 20 pixels to obtain the representative PR. Note that GTOT sets the threshold to 5 pixels as most of its tracked targets are small. Similarly, SR is the ratio of the number of successful frames whose overlap is larger than a threshold. By varying the threshold, the SR plot can be obtained, and we employ the area under curve of SR plot to define the representative SR.

6.2 Evaluation on RGBT210

We first present the evaluation results on RGBT210 against 8 popular trackers, as shown in Figure 3 (a). The comparison curves show that our tracker outperforms others with a clear margin. Table 3 presents the attribute-based comparison results of our tracker with recent 8 trackers on RGBT210. The superior results over other methods demonstrate the effectiveness of our method in handling sequences with various challenging factors. We analyze the details as follows.

Comparison with RGB trackers. We evaluate our method with several state-of-the-art RGB trackers, including SOWP [21] (most related to our method) and DSST [8] (the winner of VOT2014 competition). Table 3 shows that our method outperform these trackers, demonstrating the effectiveness of introducing thermal information in visual tracking. In particular, our tracker outperforms SOWP and DSST with 7.6%/5.1%, 15.3%/10.6% in PR/SR, respectively.

Comparison with RGB-T trackers. We further compare our tracker with several state-of-the-art RGB-T trackers, including CSR [28], JSR [34], L1-PF [42], KCF [19]+RGBT, and MEEM [49]+RGBT. Figure 3 (a) shows that our tracker significantly outperforms them, demonstrating the effectiveness of employing RGB and thermal information adaptively to construct robust object representations in our approach. In particular, our tracker achieves 18.4%/10.0%,

Table 3: Precision Rate and Success Rate (PR/SR %) based on attribute of RGBT210 dataset with 8 conventional trackers, including CSR, JSR, L1-PF, DSST, MEEM, CNN, SOWP and KCF. The best and second results are in red and green colors, respectively.

	CSR [28]	JSR [34]	L1-PF [42]	KCF+RGBT [19]	MEEM+RGBT [49]	CNN+KCF [19]+RGBT	SOWP [21]	DSST [8]	Ours
NO	68.1/45.2	39.4/27.3	56.3/37.9	56.6/36.3	64.7/41.2	63.7/42.9	75.0/46.1	70.2/41.4	82.4/50.7
PO	52.7/36.6	34.8/23.7	47.7/31.0	49.6/31.6	57.4/35.5	56.0/36.4	61.3/39.5	57.0/35.1	75.4/48.3
HO	37.1/24.3	23.8/16.5	30.4/19.5	33.0/22.2	37.2/24.2	36.6/25.9	52.0/32.8	39.4/25.7	53.1/34.1
LI	47.3/31.1	35.1/24.1	41.2/26.3	48.3/30.4	39.2/25.6	52.8/34.5	48.3/30.7	47.8/29.0	71.6/44.7
LR	46.0/23.1	27.9/15.4	45.5/22.0	42.6/26.2	44.9/23.4	54.6/32.5	51.0/29.1	52.8/29.1	65.8/37.5
TC	43.2/29.3	26.3/16.6	34.8/21.8	39.0/24.1	58.2/35.6	49.6/33.2	70.0/44.9	50.9/32.2	64.9/40.7
DEF	44.7/33.0	26.2/20.8	33.3/23.6	40.6/29.5	48.7/33.5	44.8/34.4	61.4/41.7	46.5/33.0	65.3/45.9
FM	42.6/25.0	19.3/11.9	26.8/16.3	33.3/19.1	43.5/26.8	37.1/24.1	56.0/32.3	34.4/21.2	58.0/33.1
SV	53.3/37.5	33.5/22.8	45.8/30.7	42.4/27.5	52.8/33.0	50.3/32.6	62.8/37.7	58.7/33.5	67.4/41.7
MB	34.7/23.8	19.9/14.9	26.7/19.0	29.1/20.7	46.2/31.4	30.4/22.0	55.2/38.3	32.3/23.2	58.6/39.6
CM	38.9/27.4	26.3/19.8	32.6/22.2	37.5/26.0	48.7/31.9	35.8/27.0	55.8/36.9	38.7/26.9	59.0/40.7
BC	38.4/23.7	27.0/16.9	31.4/18.1	41.0/25.6	40.5/23.4	42.3/28.4	47.2/28.6	43.8/26.3	58.6/35.5
ALL	49.1/33.0	31.1/21.3	42.0/27.4	44.0/28.5	50.5/31.9	49.3/33.1	59.9/37.9	52.2/32.4	67.5/43.0

Table 4: PR/SR (%) of the proposed method with the different versions on GTOT.

	Ours	Ours-RGB	Ours-T	Ours-noS	Ours-noR	Ours-noA
PR	85.121	69.350	73.796	76.960	84.115	82.548
SR	62.828	53.537	56.836	58.650	62.527	61.860

Table 5: Evaluation of parameter sensitivity on GTOT.

Param	Setting	PR/SR	Param	Setting	PR/SR
δ	5	83.0/61.3	α	10	83.3/62.1
	11	85.1/62.8		15	85.1/62.8
	15	82.6/61.4		20	82.1/61.2
β	0.05	84.5/62.6			
	0.15	85.1/62.8			
	0.5	84.6/62.4			

36.4%/21.7% and 25.5%/15.6% performance gains in PR/SR over CSR, JSR and L1-PF, respectively.

Comparison with deep learning tracker. We also evaluate with popular deep learning based method, including CNN [39]+KCF +RGBT presented in the Figure 3 (a). Overall, our tracker achieves a superior performance than CNN+KCF+RGBT in all challenges from the Table 3. In particular, our method also has the following advantages over the deep learning based method. 1) To obtain better tracking performance, CNN-based method usually need new labeled training data to adapt to the new task. We train our model using the ground truth in the first frame, and update it in subsequent frames. 2) It is easy to implement as each subproblem of proposed model has a closed-form solution.

6.3 Evaluation on GTOT

For further demonstrate the effectiveness of the proposed approach, we also perform evaluation on the public benchmark dataset GTOT. The comparison curves in Figure 3 (b) show that our tracker significantly outperforms other baseline trackers in both PR and SR. In particular, our tracker significantly outperforms the top second tracker CSR [28], achieving 10.2% gain in PR over it. Note that our SR is only achieves 1.3% gain over CSR, which suggests the particle filter based sampling strategy is more effective than ours in scale estimation. In future work, we will combine this strategy in our framework for better handling scale variations.

6.4 Component Analysis

To justify the significance of the main components using GTOT, we implement 5 special versions of our approach for comparative analysis including: 1) Ours-RGB, that denote only using the single RGB information in our tracking algorithm. 2) Ours-T, that just consider the thermal information as the input. 3) Ours-noS, that remove the patch weights in our algorithm. 4) Ours-noR, that give each modality a fixed weight, here, we set each modal weight to 1/2. 5) Ours-noA, that removes affinity learning, and directly utilize the representation coefficients to diffuse patch weights. Table 4 presents the evaluation results of our algorithm. We can draw the following observations and conclusions. 1) Ours outperform Ours-RGB and Ours-T, which suggests that the effectiveness of fusing RGB and thermal information. 2) Introducing patch weights into the object representations benefits to suppress the effects of background in tracking by observing the weaker performance of Ours-noS than others. 3) Ours outperform Ours-noR and Ours-noA which justify the effectiveness of the modal weights and the effectiveness of learning graph affinity matrix A, respectively.

7 CONCLUSION

In this paper, we presented a graph-based approach for RGB-T tracking, which jointly learnt the modality weights, the sparse representations, and the graph. The optimized modality weights and patch node weights were incorporated into the Struck algorithm to perform object tracking while suppressing the effects of background information. In addition, we also contributed a comprehensive dataset for RGB-T tracking purpose. Extensive experiments demonstrated the effectiveness of the proposed approach and the big challenges of the created dataset. Our future work will concentrate on developing a more powerful graph model, e.g., integrating both local and global cues, for RGB-T tracking, and expanding our evaluation platform with more challenging video sequences and baseline approaches for facilitating the related research on RGB-T tracking.

ACKNOWLEDGMENTS

This work is supported in part by the National Natural Science Foundation of China under Grant 61472002, in part by NSF CNS-1305302 and Texas State Research Enhancement Program grant to Dr. Lu, in part by the Natural Science Foundation of Anhui Higher Education Institution of China under Grant KJ2017A017, in part by the Co-Innovation Center for Information Supply & Assurance Technology, Anhui University under Grant Y01002449, and in part by the Key Scientific and Technological Project of State Grid Anhui Electric Power Company under Grant ERP No.521200130M0U.

REFERENCES

[1] G.-A. Bilodeau, A. Torabi, and P.-L. St-Charles et al. 2014. Thermal-visible registration of human silhouettes: A similarity measure performance evaluation. *Infrared Physics & Technology* 64 (2014), 79–86.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. 2011. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning* 3, 1 (2011), 1–122.

[3] F. Bunyak, K. Palaniappan, S. K. Nath, and G. Seetharaman. 2007. Geodesic active contour based fusion of visible and infrared video for persistent object tracking. In *Proceedings of IEEE Workshop on Applications of Computer Vision*.

[4] D. Comaniciu, V. Ramesh, and P. Meer. 2003. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2003).

[5] C. O. Conaire, N. Connor, and A. Smeaton. 2007. Thermo-visual feature fusion for object tracking using multiple spatiogram trackers. *Machine Vision and Applications* 7 (2007), 1–12.

[6] C. O Conaire, N. E. Connor, E. Cooke, and A. F. Smeaton. 2006. Comparison of fusion methods for thermo-visual surveillance tracking. In *Proceedings of International Conference on Information Fusion*.

[7] N. Cvejic, S. G. Nikolov, H. D. Knowles, A. Loza, A. Achim, D. R. Bull, and C. N. Canagarajah. 2007. The effect of pixel-level fusion on object tracking in multi-sensor surveillance video. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

[8] M. Danelljan, G. Hager, F. Khan, and M. Felsberg. 2014. Accurate Scale Estimation for Robust Visual Tracking. In *Proceedings of British Machine Vision Conference*.

[9] J. W. Davis and M. A. Keck. 2005. A Two-Stage Template Approach to Person Detection in Thermal Imagery. In *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*.

[10] J. W. Davis and V. Sharma. 2007. Background-subtraction using contour-based fusion of thermal and visible imagery. *Computer Vision and Image Understanding* 106, 2 (2007), 162–182.

[11] D. L. Donoho. 2006. Compressed sensing. *IEEE Transactions on Information Theory* 52, 4 (2006), 1289–1306.

[12] S. Duffner and C. Garcia. 2013. Pixeltrack: A fast adaptive algorithm for tracking non-rigid objects. In *Proceedings of IEEE International Conference on Computer Vision*.

[13] E. Elhamifar and R. Vidal. 2009. Sparse subspace clustering. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

[14] M. Felsberg, A. Berg, and G. et al. Hager. 2015. The Thermal Infrared Visual Object Tracking VOT-TIR2015 Challenge Results. In *Proceedings of IEEE International Conference on Computer Vision*.

[15] R. Gade and T. B. Moeslund. 2014. Thermal cameras and applications: a survey. *Machine Vision and Applications* 25 (2014), 245–262.

[16] X. Guo. 2015. Robust Subspace Segmentation by Simultaneously Learning Data Representations and Their Affinity Matrix. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[17] S. Hare, A. Saffari, and P. H. S. Torr. 2011. Struck: Structured output tracking with kernels. In *Proceedings of IEEE International Conference on Computer Vision*.

[18] S. He, Q. Yang, R. Lau, J. Wang, and M.-H. Yang. 2013. Visual tracking via locality sensitive histograms. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

[19] Joao F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. 2015. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2015).

[20] S. Hwang, J. Park, and N. et al. Kim. 2015. Multispectral Pedestrian Detection: Benchmark Dataset and Baseline. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

[21] H.-U. Kim, D.-Y. Lee, J.-Y. Sim, and C.-S. Kim. 2015. SOWP: Spatially Ordered and Weighted Patch Descriptor for Visual Tracking. In *Proceedings of IEEE International Conference on Computer Vision*.

[22] M. Kristan, J. Matas, A. Leonardis, and M. Felsberg et al. 2015. The Visual Object Tracking VOT2015 challenge results. In *Proceedings of IEEE International Conference on Computer Vision*.

[23] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, and L. Cehovin et al. 2014. The Visual Object Tracking VOT2014 challenge results. In *Proceedings of European Conference on Computer Vision*.

[24] S. J. Krotosky and M. M. Trivedi. 2007. On Color-, Infrared-, and Multimodal-Stereo Approaches to Pedestrian Detection. *IEEE Trans. Intelligent Transportation Systems* 8, 4 (2007), 619–629.

[25] X. Lan, A. J. Ma, and P. C. Yuen. 2014. Multi-cue visual tracking using robust feature-level fusion based on joint sparse representation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

[26] A. Leykin and R. Hammoud. 2010. Pedestrian tracking by fusion of thermal-visible surveillance videos. *Machine Vision and Applications* 21, 4 (2010), 587–595.

[27] A Li, M Lin, Y Wu, MH Yang, and S Yan. 2016. NUS-PRO: A New Visual Tracking Challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 2 (2016), 335–349.

[28] C. Li, H. Cheng, S. Hu, X. Liu, J. Tang, and L. Lin. 2016. Learning Collaborative Sparse Representation for Grayscale-thermal Tracking. *IEEE Transactions on Image Processing* 25, 12 (2016), 5743–5756.

[29] C. Li, L. Lin, W. Zuo, and J. Tang. 2017. Learning Patch-Based Dynamic Graph for Visual Tracking. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. 4126–4132.

[30] C. Li, X. Sun, X. Wang, L. Zhang, and J. Tang. 2017. Grayscale-thermal Object Tracking via Multi-task Laplacian Sparse Representation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47, 4 (2017), 673–681.

[31] C. Li, X. Wang, L. Zhang, J. Tang, H. Wu, and L. Lin. 2016. WELD: Weighted Low-rank Decomposition for Robust Grayscale-Thermal Foreground Detection. *IEEE Transactions on Circuits and Systems for Video Technology* (2016).

[32] P. Liang, E. Blasch, and H. Ling. 2015. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Transactions on Image Processing* 24, 12 (2015), 5630–5644.

[33] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. 2013. Robust Recovery of Subspace Structures by Low-Rank Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 1 (2013), 171–184.

[34] H. Liu and F. Sun. 2012. Fusion tracking in color and infrared images using joint sparse representation. *Information Sciences* 55, 3 (2012), 590–599.

[35] C. Ma, X. Yang, C. Zhang, and M.-H. Yang. 2015. Long-term Correlation Tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

[36] I. A. Matthews, T. Ishikawa, and S. Baker. 2004. The Template Update Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 6 (2004), 810–815.

[37] A. Y. Ng, Michael I. Jordan, and Yair Weiss. 2001. On Spectral Clustering: Analysis and an algorithm. In *Proceedings of Neural Information Processing Systems*.

[38] J. Portmann, S. Lynen, M. Chli, and R. Siegwart. 2014. People detection and tracking from aerial thermal views. In *Proceedings of IEEE International Conference on Robotics and Automation*.

[39] K. Simonyan and A. Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

[40] A. Torabi, G. Masse, and G.-A. Bilodeau. 2012. An iterative integrated framework for thermal-visible image registration, sensor fusion, and people tracking for video surveillance applications. *Computer Vision and Image Understanding* 116, 2 (2012), 210–221.

[41] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. 2005. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research* 6 (2005), 1453–1484.

[42] Y. Wu, E. Blasch, G. Chen, L. Bai, and H. Ling. 2011. Multiple source data fusion via sparse representation for robust visual tracking. In *Proceedings of International Conference on Information Fusion*.

[43] Y. Wu, J. Lim, and M.-H. Yang. 2013. Online object tracking: A benchmark. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

[44] Y. Wu, J. Lim, and M.-H. Yang. 2015. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2015).

[45] Z. Wu, N. Fuller, D. Theriault, and M. Betke. 2014. A Thermal Infrared Video Benchmark for Visual Analysis. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

[46] Y. Xu and W. Yin. 2013. A Block Coordinate Descent Method for Regularized Multiconvex Optimization with Applications to Nonnegative Tensor Factorization and Completion. *SIAM Journal on Imaging Sciences* 6, 3 (2013), 1758–1789.

[47] S. Yan and H. Wang. 2009. Semi-supervised Learning by Sparse Representation. In *Proceedings of the SIAM International Conference on Data Mining*.

[48] F. Yang, H. Lu, and M.-H. Yang. 2014. Robust Superpixel Tracking. *IEEE Transactions on Image Processing* 23, 4 (2014), 1639–1651.

[49] J. Zhang, S. Ma, and S. Sclaroff. 2014. MEEM: robust tracking via multiple experts using entropy minimization. In *Proceedings of European Conference on Computer Vision*.

[50] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Scholkopf. 2004. Ranking on data manifolds. In *Proceedings of Neural Information Processing Systems*.