PRIFYSGOL ABERYSTWYTH UNIVERSITY

## Aberystwyth University

# Real-time infrared target tracking based on ℓ1 minimization and compressive features

Li, Ying; Li, Pengcheng; Shen, Qiang

*Published in:*
Applied Optics

*DOI:*
10.1364/AO.53.006518

*Publication date:*
2014

*Citation for published version (APA):*
Li, Y., Li, P., & Shen, Q. (2014). Real-time infrared target tracking based on ℓ1 minimization and compressive features. *Applied Optics*, *53*(28), 6518-6526. https://doi.org/10.1364/AO.53.006518

**General rights**
Copyright and moral rights for the publications made accessible in the Aberystwyth Research Portal (the Institutional Repository) are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the Aberystwyth Research Portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the Aberystwyth Research Portal

**Take down policy**
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

tel: +44 1970 62 2400
email: is@aber.ac.uk

Download date: 22. Mar. 2026

# Real-time infrared target tracking based on $l_1$ minimization and compressive features

Ying Li[1] , Pengcheng Li[1], Qiang Shen[2]
1. School of Computer Science, Northwestern Polytechnical University, Shaanxi, Xi'an 710129, China
2. Institute of Mathematics, Physics and Computer Science, Aberystwyth University, Aberystwyth SY23 3DB, UK

**Abstract** Tracking a target in infrared (IR) sequences is a challenging task because of low resolution, low signal-to-noise ratio, occlusion, and poor target visibility. For many civil and military applications, real-time performance is also a key requirement for tracking algorithms. This inevitably makes the tracking in IR sequences more difficult. This paper presents an approach for real-time IR target tracking under complex conditions, based on $l_1$ minimization and compressive features. We adopt a sparse measurement matrix technique to project the original high dimensional Harr-like features to low dimensional ones in appearance modeling. Such a model allows significant reduction in computation complexity and hence the cost of target tracking. In particular, the appearance model is utilized within the framework of the popular $l_1$ tracker. Each candidate target is depicted by an appearance template that reflects the underlying structure of sparse representation. The candidate that has the minimum reconstruction error is selected as the tracking result. This proposed approach combines the real-time advantage of compressive tracking and the robustness of the $l_1$ tracker. Systematic experimental results on challenging IR

image sequences, including both aerial targets and ground targets, demonstrate that the proposed method indeed outperforms two latest state-of-the-art tracking algorithms in terms of robustness and real-time performance.

**Keywords** infrared target tracking, appearance model, compressive features, sparse representation.

# 1. Introduction

Tracking a target in infrared (IR) image sequences is required in many civil and military applications such as precision guidance, early warning, and video surveillance. However, tracking objects in IR sequences is often a complex and difficult process. This is because of a number of entangled factors, including low contrast between targets and background, noise and background clutter, sophisticated object motion, relevant ego motion, partial and full occlusion, complex object shapes, and illumination and scale changes [1].

Different approaches have been proposed to address this difficult task. For instance, Bal and Alam proposed a novel tracking method using intensity variation function and template modeling [2]. Paravati et al. provided an alternative algorithm through the use of GAs to reduce the cost of computing intensity variation function [1]. In [3], Ling et al. presented a tracking technique via the application of kernel-based performance metric and eigenvalue-based similarity measures. Also, an improved IR target-tracking mechanism is given in [4] which works based on mean shift. There have been many other techniques proposed too, e.g., those as reported in [5-10].

Despite such developments, infrared target tracking is still a tough and challenging task, especially when real-time requirements have to be taken into account.

Recently, sparse representation has been successfully applied to visual tracking [11-14]. In this case, the tracker models the target appearance using a sparse approximation over a template set. This leads to the so-called $l_1$ tracker [11] as it works by resolving a $l_1 - norm$ related minimization problem. The $l_1$ tracker can effectively handle partial occlusion with a set of trivial templates and show favorable tracking accuracy. However, it has to carry out computationally expensive $l_1 - norm$ related minimization over each image frame. Furthermore, in a particle filter frame, computation cost grows linearly with the number of sampled particles. Obviously, this computational bottleneck precludes the use of the $l_1$ tracker in real-time scenarios.

In [15] and [16], helpful strategies are proposed to create an efficient solver for the $l_1 - norm$ related minimization problems, thereby improving the tracker speed considerably. Nevertheless, the nature of expensive computation involved in $l_1 - norm$ related minimization problems is caused by the underlying high dimensionality of target features. Thus, using low-dimensional target features will reduce the computation of the $l_1$ tracker, but low-dimensional features may not contain sufficient information to distinguish the IR target from other candidates.

Fortunately, the technique of random projection can help project the features from a high dimensional space to a low dimensional one whilst preserving almost all the information embedded in the original high-dimensional features [17]. Such low-dimensional features may offer sufficient information content [18] to allow for

the separation of the real target from the rest. For example, Zhang et al. proposed a compressive tracking algorithm [19] which employs non-adaptive random projections to compress the features to support running the tracking process in real-time. Yet the tracking process in this algorithm is formulated within a naïve Bayes classifier framework, which has limited robustness for occlusions and noise.

Inspired by the aforementioned observation, this paper presents a real-time IR target tracking approach based on the utilization of both $l_1$ minimization and compressive features. The work combines the real-time ability of compressive tracking and the robust performance of the $l_1$ tracker. Note that a typical tracking system consists of three key components: an appearance model, a motion model and a search strategy for identifying the most likely location in the current frame. In our work, to represent the target, we extract the high-dimensional, multi-scale Harr-like features [20, 21] from the object region, and select a small subset of them for tracking in the compressed domain. As for the motion model, the particle filter framework is herein applied to predict the possible targets in the next frame. Each candidate target is depicted by sparse representation and the candidate target that has the least reconstruction error is chosen as the tracking result in the current frame. Experimental results demonstrate that the proposed method can significantly improve the accuracy and robustness of target tracking in IR image sequences, in comparison with the $l_1$ tracker and the compressive tracking algorithm.

The remainder of this paper is organized as follows. In Section 2, we review the relevant work including both particle filter and sparse representation based tracking.

We also reinforce our motivation for the present work. In Section 3, we describe the proposed tracking algorithm. In Section 4, we show both qualitative and quantitative experimental results, systematically evaluated over infrared video clips, in comparison with the $l_1$ tracker and the compressive tracking algorithm. Finally, we conclude the paper in Section 5, including a brief discussion of interesting further research.

## 2. Background

As indicated previously, much work has been done in infrared target tracking. In this section, we focus on the discussion of the most relevant algorithms, covering the tracking techniques based on particle filters and sparse representation. We also emphasize our motivation for the subsequent development.

### A. Particle Filters for Tracking

A particle filter [22] implements the Bayesian sequential importance sampling technique for approximately computing the posterior distribution of state variables that characterize a dynamic system. In visual tracking, such a filter offers an effective means for estimating the target of next frame without presuming concrete observation probabilities.

The particle filter method essentially consists of two steps: prediction and update. Formally, let $X_t$ describe the location and shape of the target in frame $t$, and $y_{1:t-1} = \{y_1, y_2, \mathrm{L}\ , y_{t-1}\}$ denote the observation of the target from the first frame to

frame $t-1$. A particle filter operates the above two steps with the following two probabilities:

$$p(X_t \mid y_{1:t-1}) = \int p(X_t \mid X_{t-1}) p(X_{t-1} \mid y_{1:t-1}) dX_{t-1} \tag{1}$$

$$p(X_t \mid y_{1:t}) = \frac{p(y_t \mid X_t) p(X_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})} \tag{2}$$

The optimal state for frame $t$ is obtained according to the maximal approximate posterior probability:

$$X_t^* = \arg\min_X p(X \mid y_{1:t}) \tag{3}$$

The posterior probability (2) is approximated by using finite samples $S_t = \{X_t^1, X_t^2, \mathrm{L}\ , X_t^N\}$ with different weights $W = \{w_t^1, w_t^2, \mathrm{L}\ , w_t^N\}$, where $N$ is the number of samples. These samples are generated from the sequential importance distribution $\prod(X_t \mid y_{1:t}, X_{1:t-1})$ and the weights are updated by

$$w_t^i \propto w_{t-1}^i \frac{p(y_t \mid X_t^i) p(X_t^i \mid X_{t-1}^i)}{\prod(X_t \mid y_{1:t}, X_{1:t-1})} \tag{4}$$

For tracking systems, an affine image warping is usually used to capture the target motion between two consecutive frames. We apply a Gaussian model to describe the state transition distribution P(xt | xt-1), and the observation model P(yt|xt) to represent the similarity between a target candidate and a certain target template. The latter is often formulated using the reconstruction error approximated by the given target templates.

## B. Sparse Representation for Tracking

Sparse representation is a task of reconstructing a given signal by selecting a small

subset of bases from a large basis pool, while keeping the reconstruction error to the minimal possible. It has been used in many computer vision applications, such as visual tracking, image restoration, and image classification [11-16, 23]. In particular, in the field of target tracking, the sparse representation model aims at reconstructing the target candidate by selecting a relatively small subset of appearance templates for use.

Formally, at frame $t$, given the appearance template set $T_t = \{t_t^1, t_t^2, \mathrm{L}\ , t_t^n\}$, let $O_t = \{y_t^1, y_t^2, \mathrm{L}\ , y_t^N\}$ denote the corresponding candidate targets in the current IR image. The sparse representation model takes the form

$$\begin{aligned} &\min_c \frac{1}{2}\|y - Bc\|_2^2 \\ &s.t. \quad \|c\|_0 \le u \end{aligned} \tag{5}$$

where $y$ denotes the target candidate, $B = [T, I]$ denotes the appearance templates which include target templates $T$ and also trivial templates $I$, and $c = \begin{bmatrix} a & e \end{bmatrix}^T$ denotes the corresponding coefficients where $a$ indicates the weights of target templates, and $e$ is an error term (which can be viewed as the weight of trivial templates). This model can be resolved via optimizing the following $\mathrm{l}_1 - norm$ related minimization

$$\min_c \frac{1}{2}\|y - Bc\|_2^2 + \lambda\|c\|_1 \tag{6}$$

where $\|\cdot\|_2$ and $\|\cdot\|_1$ denote the $\mathrm{l}_2$ and $\mathrm{l}_1$ norm respectively, and $\lambda$ is a regularization constant.

Given the coefficients $c$, the reconstruction error of each candidate $y_i$ can be evaluated by the following equation:

$$E(y_i) = \|y_i - Ta\|_2^2 \quad (7)$$

From this, the candidate with the smallest reconstruction error is selected as the tracking result for the current frame.

## C. Motivation for this Research

Sparse representation has been applied to visual tracking by modeling the target appearance using an approximation over a template set. This leads to the so-called $l_1$ tracker [11] as the underlying work translates to resolving $l_1 - norm$ related minimization problems. This type of tracker can effectively handle partial occlusion with a set of trivial templates and show favorable tracking accuracy. However, typically, such a tracker has to solve hundreds of the relevant minimization problems for each individual frame during a tracking process, thereby involving expensive computation. Clearly, a direct implementation of this technique is not practically feasible for real-time applications.

Interesting strategies have been proposed to create an efficient solver for the $l_1 - norm$ related minimization problems, in an effort to improve the speed of the resulting tracker [15, 16]. Unfortunately, the underlying limitation due to the need of repeatedly performing $l_1 - norm$ related minimization is not reduced, because of the high dimensionality of target features. However, by the use of target features of a reduced dimensionality the computation of an $l_1$ tracker will be reduced accordingly. The question is whether the low-dimensional features may still contain sufficient information that is required to distinguish the IR target from other candidates.

Also, there have been a number of discriminative tracking algorithms proposed in the literature, by viewing the task of tracking as a binary classification problem between the target and the background. For example, Avidan has extended the optical flow approach with a support vector machine classifier for object tracking [24]. In [19], Zhang has employed a naïve Bayes classifier to run a real time tracker. These algorithms focus on how to train a robust classifier to distinguish the target from the background. Generally, they can run with fast speed and may possess good robustness. However, when the target suffers from partial occlusion or is blurred with noise, the potential of these algorithms is significantly restricted due to the lack of the noise-handling ability that an $l_1$ tracker has.

As indicated earlier, the technique of random projection may help project the features from a high dimensional space onto a low dimensional one, while preserving almost all the information contained within the original high-dimensional features [19, 25]. The resulting low-dimensional features can therefore offer a similar amount of information to support distinguishing the real target from the background. The idea of utilizing feature selection to aid in the reduction of computation in order to improve the efficacy of target tracking is known. For example, Grabner et al. have provided an online boosting method to select features for tracking [26]. Yet, little has been done to use the same idea to support the development of effective and efficient $l_1$ trackers. It is inspired by this observation that we apply random projection to compress the image features and thus, to reduce the computation complexity to be incurred in performing target tracking. In so doing, this work exploits the strengths of both techniques: sparse

representation for handling partial occlusion and noise and random projection for reducing the dimensionality of features.

## 3. Proposed Tracking Algorithm

In this section, we present the real-time infrared target tracking algorithm based on the use of $l_1$ minimization and feature compression. Fig.1 shows the flow chart of the proposed approach. In the following subsections, we will detail the three main components of this tracking algorithm, including: the appearance model, the motion model, and the search strategy. The others are self-explanatory.

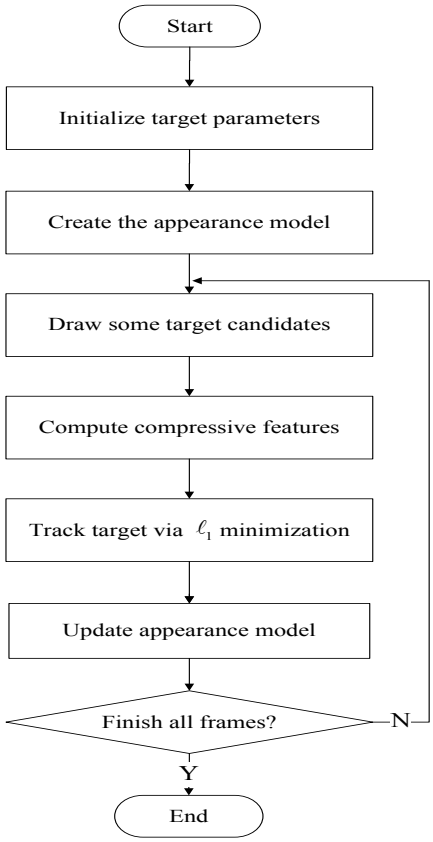



**Fig. 1** Flow chart of the proposed algorithm

*A. Appearance Model*

In this work, Harr-like features are used to represent given infrared targets, which are introduced first below. Then, we describe the method for obtaining compressive features based on random projection, in an effort to efficiently reduce the dimensionality of the original Harr-like features. Finally, we present the complete appearance model at the end of this subsection.

*1. Harr-like Features*

Harr-like features have been widely used for object detection and tracking with demonstrated success. The basic types of Harr-like feature are normally designed for use in performing tasks different from the present one [20, 21]. In our algorithm, each IR target sample $z \in \circ^{w \times h}$ is firstly transformed by convoluting it with a series of different scale rectangle filters $\{h_{1,1}, h_{1,2}, \mathrm{L}\ , h_{w,h}\}$, for copying with the scale problem. These filters are defined by

$$h_{i,j}(x,y) = \begin{cases} 1, & 1 \le x \le i, 1 \le y \le j \\ 0, & otherwise \end{cases} \tag{8}$$

where $i, j$ are the width and height of a rectangle filter, respectively. Then, each filtered image is represented as a column vector in $\circ^{w \times h}$ and these vectors are concatenated as a very high-dimensional multi-scale image feature vector $x = (x_1, x_2, \mathrm{L}\ , x_m)^T \in \circ^{m}$ where $m = (w \times h)^2$. The dimensionality $m$ is usually in the order of $10^6$ to $10^8$. Thus, these feature vectors that would otherwise be directly applied to represent the IR target inevitably lead to heavy computational overheads.

*2. Compressive Features*

In order to track infrared target in real time, we have to efficiently reduce the dimensionality of Harr-like features. Inspired by the work of [19], a large set of given Harr-like features is to be compressively sensed with non-adaptive random projections. The compressive sensing theories [25, 27] ensure that the returned features in the compressed domain preserve almost all the information contained within the original image.

For completeness, a brief introduction to random projection [19] is presented here. A random matrix $R \in °^{n \times m}$ whose rows have a unit length projects data from a given high-dimensional image space $x \in °^{m}$ to a lower-dimensional space $v \in °^{n}$

$$v = Rx \tag{9}$$

where $n = m$. Ideally, $R$ is expected to provide a stable embedding that approximately preserves the distance between all pairs of the original signals. It I known that, if the random matrix $R$ in Eq. (9) satisfies the Johnson-Lindenstrauss lemma, we can reconstruct audio or image signal $x$ with a minimum error from $v$ with high probability. Moreover, $v$ preserves almost all the information contained within $x$.

In compressive sensing, a typical measurement matrix satisfying the restricted isometry property is the random Gaussian matrix $R \in °^{n \times m}$, where $r_{ij} : N(0,1)$. This measurement matrix can be utilized to project features from a high-dimensional space to a low-dimensional one, thereby reducing computational complexity. However, if the matrix is dense, the memory and computational loads required to

implement this projection process are usually still too large when $m$ is large.

To address this issue, we adopt a sparse random measurement matrix $R \in °^{n \times m}$ to accomplish the projection task. The entries of such a sparse matrix $R$ are defined as

$$r_{ij} = \sqrt{s} \times \begin{cases} 1 & with \quad probability \quad \frac{1}{2s} \\ 0 & with \quad probability \quad 1 - \frac{1}{s} \\ -1 & with \quad probability \quad \frac{1}{2s} \end{cases} \tag{10}$$

Note that as indicated in [18], when $s = m / \log(m)$, the random projections achieved are almost as accurate as the conventional random projections where $r_{ij} \sim N(0,1)$. In particular, when $s = m / 4$, the computational complexity is low and the required memory to store the nonzero entries of $R$ is also rather light.

### *3. Appearance Model*

A good appearance model is expected to be able to handle the variance of the target such as illumination changes, background clutter, occlusion, etc. The appearance model adopted in this work comprises both target templates and trivial templates. In particular, the target templates are obtained initially, by randomly selecting a number of samples (patches) around the true target center in the first frame. Then, the aforementioned feature extraction method is used to create the features of these samples, forming the set $T = [t_1, t_2, \mathrm{L}\ , t_n]$. The set of trivial templates is introduced to deal with any partial occlusions and noise in the IR sequences. Here, within the set $I = [i_1, i_2, \mathrm{L}\ , i_d] \in R^{d \times d}$ each trivial template $i_i \in R^d$ is implemented with a vector of only one nonzero entry such that $I$ is an identity matrix. Fig. 2 illustrates the basic

construction of the appearance model.

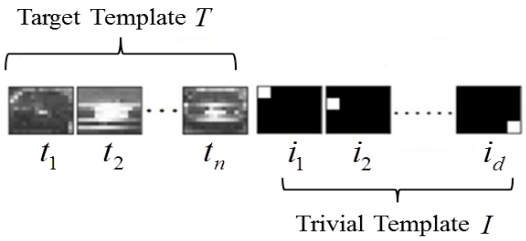



Fig.2 Appearance model in the proposed algorithm

*B. Motion Model*

As indicated previously, particle filters provide a significant tool for estimating the target of the next frame without knowing any concrete observation probability. Such filters execute two steps of operation with two different probabilities as formulated in Eqs. (1) and (2), respectively.

From the viewpoint of Bayes theorem, Eq. (1) can be seen as a prior probability of the relationship between the observations gathered so far, from the first frame up to frame $t-1$, and the target at frame $t$. While obtaining the observation at time $t$, the probability is revised by Eq. (2). This probability is referred to as posterior probability due to the influence of observation in frame $t$. It is approximated by using finite samples $S_t = \{X_t^1, X_t^2, \mathrm{L}\ , X_t^N\}$ with different weights $W = \{w_t^1, w_t^2, \mathrm{L}\ , w_t^N\}$ where $N$ is the number of samples.

In the proposed tracking algorithm, we apply four parameters $[x, y, w, h]$ to model the target, where $x, y$ represent the coordinates of the target center, and $w, h$ denote the width and height of the target, respectively. Here, we simply adopt the target parameters $(x_t, y_t, w_t, h_t)$ to model the state transition $p(x_t \mid x_{t-1})$ which is

formulated by random walk, i.e., $p(x_t \mid x_{t-1}) = N(x_t; x_{t-1}, \Psi)$, where $\Psi$ is a diagonal covariance matrix.

*C. Search Strategy*

The search strategy used in this work is based on sparse representation. Given certain infrared candidate targets produced by the motion model in the current frame, we firstly adopt the sparse representation algorithm to describe each candidate in the target temple space. Then, we compute the reconstruction error of each candidate and choose the candidate that has the minimum reconstruction error as the track result in current frame.

*1. Sparse Representation of Infrared Targets*

Given the target template set $T_t = \{t_t^1, t_t^2, \mathrm{L}\ , t_t^n\}$, an IR target $y$ can be approximately represented by the linear combination of the elements of this set, that is

$$y \approx Ta = a_1 t_1 + a_2 t_2 + \mathrm{L}\ + a_n t_n \tag{11}$$

where $a = (a_1, a_2, \mathrm{L}\ , a_n)^T$ is the target coefficient vector.

The trivial template $I = [i_1, i_2, \mathrm{L}\ , i_d]$ ($I$ is an identity matrix) is introduced to take the occlusion and noise into consideration, so Eq. (11) can be rewritten as

$$y = \left[T, I\right]\begin{bmatrix} a \\ e \end{bmatrix} = Bc \tag{12}$$

where $e = (e_1, e_2, \mathrm{L}\ , e_d)^T$ is the trivial coefficient vector.

For a good IR target candidate, there should be only a limited number of nonzero coefficients in $e$ that account for the noise and partial occlusion. Therefore, we

pursue the minimum reconstruction error to meet the requirements of sparseness in the coefficient $c$ in Eq. (12). Recall that the sparse representation model takes the form as formulated in Eq. (5). For each target candidate $y_i$, the corresponding coefficient $c_i$ can thus be computed by

$$L(c_i) = \min_{c_i} \frac{1}{2} \| y_i - Bc_i \|_2^2 + \lambda \| c_i \|_1 \tag{13}$$

where $\lambda$ is a regularization constant which denotes the weight of the sparseness in coefficient. The larger $\lambda$ is, the more important the weight of sparseness signifies.

*2. Reconstruction Error*

From above it is known that for each target candidate $y_i$ in the current frame, its coefficient is $c_i = \begin{bmatrix} a_i \\ e_i \end{bmatrix}$. Then, the reconstruction error can be computed by

$$E(y_i) = \frac{1}{\Gamma} \exp\{-\alpha \| y_i - Ta_i \|_2^2\} \tag{14}$$

where $\alpha$ is a constant controlling the shape of the Gaussian kernel, $\Gamma$ is a normal factor, and $a_i$ is the corresponding target template coefficient. Thus, the optimal target candidate $y_i^*$ can be obtained by

$$y_i^* = \arg\min_{y_i \in Y} E(y_i) \tag{15}$$

This means that we choose the candidate which has the minimum reconstruction error as the target result in the current frame.

*3. Target Template Update*

In the above description, it is assumed that the target is tracked through the infrared

sequences by initially extracting a template from the first frame and then searching the target in successive frames. Due to the changes in the background of IR sequences, a fixed target template in appearance model is not sufficient to handle the variance of the target [28]. However, a rapidly changing template is susceptible to drift. That is, if we do not update the template, the template cannot capture the target appearance variations. If however, we update the template too frequently, small errors are introduced each time as the template is updated and, as such, errors may be accumulated and the tracker may drift from the target.

To balance, we adopt a dynamical template update scheme as introduced in [12], especially to overcome pose and illumination changes. The update scheme is summarized in Algorithm 1.

---

**Algorithm 1** Template Update

---

**Input:** the original template set $T$, the newly chosen tracking target $y$, the solution $a$ to (16), the current weights $w$ ($w_i \leftarrow \|t_i\|_2$), the predefined threshold $\tau$

1: Update weights according to the coefficients of the target templates by

$w_i \leftarrow w_i * \exp(a_i)$

---

2: **If** ( $sim(y,t_m) < \tau$ ), where $sim(\cdot)$ is the SSD between two vectors after normalization, and $t_m$ has the largest coefficient $a_m$, that is, $m = \arg\max_{1 \le i \le n} a_i$

**Then**

$i_0 \leftarrow \arg\min_{1 \le i \le n} w_i$

$t_{i_0} \leftarrow y$ /* replace an old template*/

$w_{i0} \leftarrow median(w)$ /* replace an old weight*/

**End If**

3: Normalize $w$ such that $sum(w) = 1$

4: Adjust $w$ such that $\max(w) = 0.3$ to prevent skewing

5: Normalize $t_i$ such that $\|t_i\|_2 = w_i$

**Output**: the updated template set $T$

## 4. Experimental Results

The performance of the proposed target tracking algorithm was evaluated by computer simulation. All experiments are implemented using MATLAB 2012a on a PC with Intel(R) Core(TM) i5, 3.20GHz CPU and 3.20GB RAM. Five challenging infrared image sequences (denoted respectively as Seq.1, Seq.2, Seq.3, Seq.4 and Seq.5 in the following) were used as test sequences. For each sequence, the location of the target is manually labeled in the first frame. As a trade-off between computation efficiency and effectiveness, the implementation of the proposed approach is empirically set to use 600 particles. The appearance model is updated by

the above template-updating algorithm. We compare our work with the $l_1$ tracker and the compressive tracking algorithm, both qualitatively and quantitatively.

*A. Qualitative Comparison*

Seq.1 is set in the background of cloud and sky with a size of 256×200 pixels for each frame and the tracked target is a small moving aircraft somewhat similar to the background [29]. The target-to-background contrast is low and the noise level is high for these IR frames. Samples of the final tracking results are shown in Fig. 3. The frame indexes are 9, 30, 51, 69, 90 and 134. Due to the low target-to-background contrast, the $l_1$ tracker loses the target easily because the gray features cannot distinguish the target from the background. The compressive tracking algorithm is superior to the $l_1$ tracker and successfully tracks the target. The proposed algorithm adopts the compressive Harr-like features to effectively represent the target, which not only reduces the dimensionality of features but also improves the tracking accuracy further.

Seq.2 is a very challenging infrared sequence, again of the size of $256 \times 200$ pixels per frame. The target-to-background contrast is very low, and the target is a dim point object embedded in heavy clutter background [29]. Six representative frames with indices 8, 12, 24, 52, 88 and 120 are shown in Fig. 4. From this, we can see that the $l_1$ and compressive trackers both lose target quickly. In contrast, the proposed algorithm is capable of successfully tracking the small dim infrared target in the entire sequence even with severe occlusions by the clouds.

Seq.3 is the *PkTest02* sequence (size 320×256) that is obtained from the VIVID benchmark dataset [30]. It is a vehicle sequence with significant pose, lighting and scale variations in a cluttered scene. Frames with indices 250, 300, 350, 400, 410 and 420 are shown in Fig. 5. It can be observed that the compressive tracking algorithm drifts apart in several frames. The $l_1$ tracker and the proposed method can track the target car well.

Seq.4 is also a vehicle sequence (but of a size $320 \times 240$) with very low target-to-background contrast and partial occlusions. We give the tracking results on six representative frames with indices 27, 59, 86, 107, 138 and 171 as shown in Fig. 6. The results demonstrate both the $l_1$ tracker and the compressive tracking method fail to track the target when the car is occluded by the trees. Although the proposed tracking algorithm also loses the target temporarily for a few frames, it is able to relocate on the target during the following tracking.

Seq.5 is an IR sequence taken from the popular OSU thermal database (size $320 \times 240$) [31]. We present the tracking results on six representative frames with indices 6, 135, 159, 274, 358 and 424 in Fig. 7. The proposed algorithm and the compressive tracking method perform better than the $l_1$ tracker. The proposed algorithm can deal with the partial occlusion, while the $l_1$ tracker fails to capture the object using appropriate features.

*B. Quantative Comparison*

To quantitatively compare the performance of the tracking methods, we use two

different metrics, namely, the mean distance to the ground truth and the percentage of correctly tracked frames. A frame is correctly tracked if the tracked target and its underlying ground truth have an overlap that is larger than half of the union of their areas. This is, if the tracked target is of an area $A$, and the ground truth is of an area $B$, then a frame is correctly tracked if $(A \cap B)/(A \cup B) > 0.5$. The second metric is much more informative than the distance, since once the track is lost the distance to the ground truth is somewhat arbitrary, and may bias the average distance.

The comparison results, obtained by the use of the above two metrics over four sequences are shown in Tables 1 and 2, respectively. It is clear that both tables show superior results of the proposed method to the others, over all four sequences.

Finally, in terms of runtime performance, the average temporal cost for each of the tracking methods to process one frame on the MATLAB platform is shown in Table 3. The proposed algorithm runs much faster than the $l_1$ tracker, and almost as fast as the compressive tracking algorithm. Importantly, of course, the efficiency of the proposed approach is achieved while it gains overall tracking accuracy (as shown above).



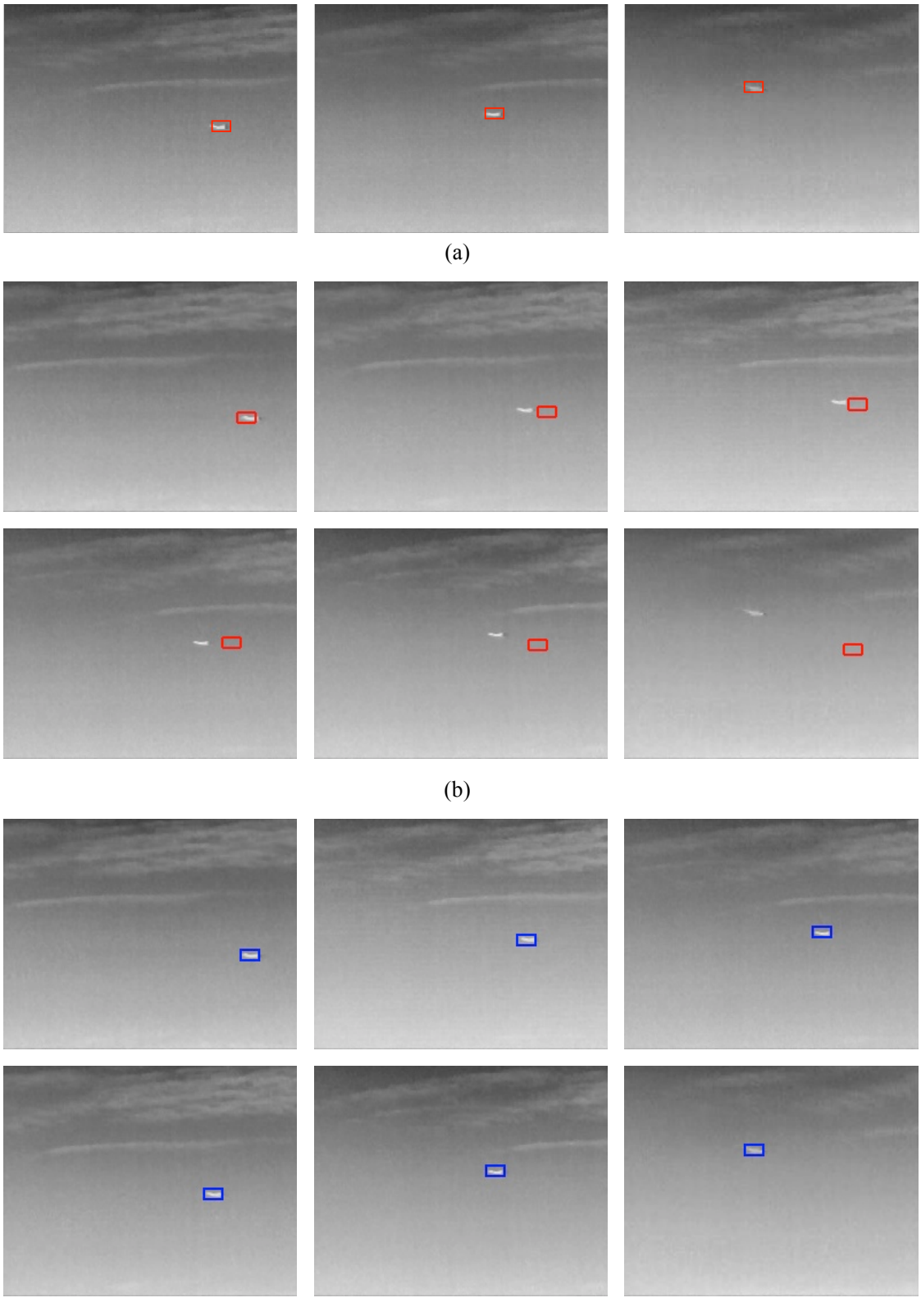
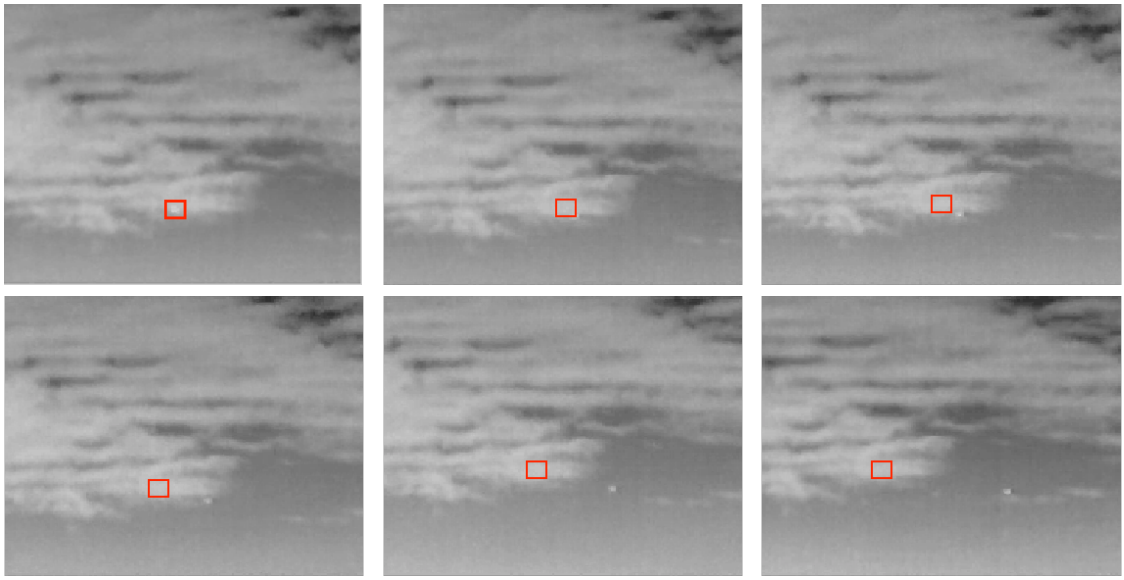
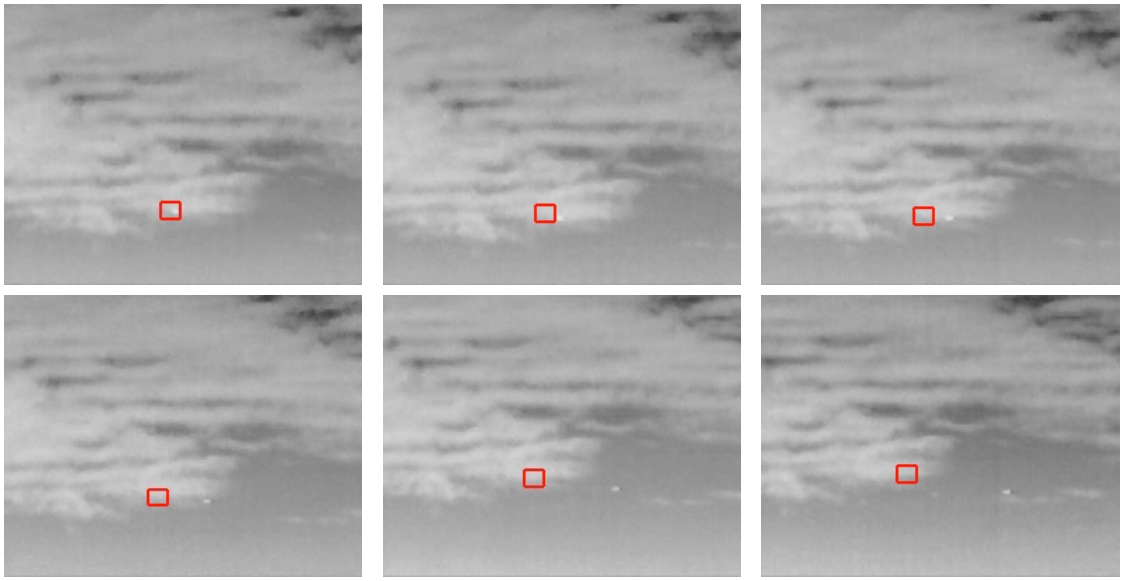


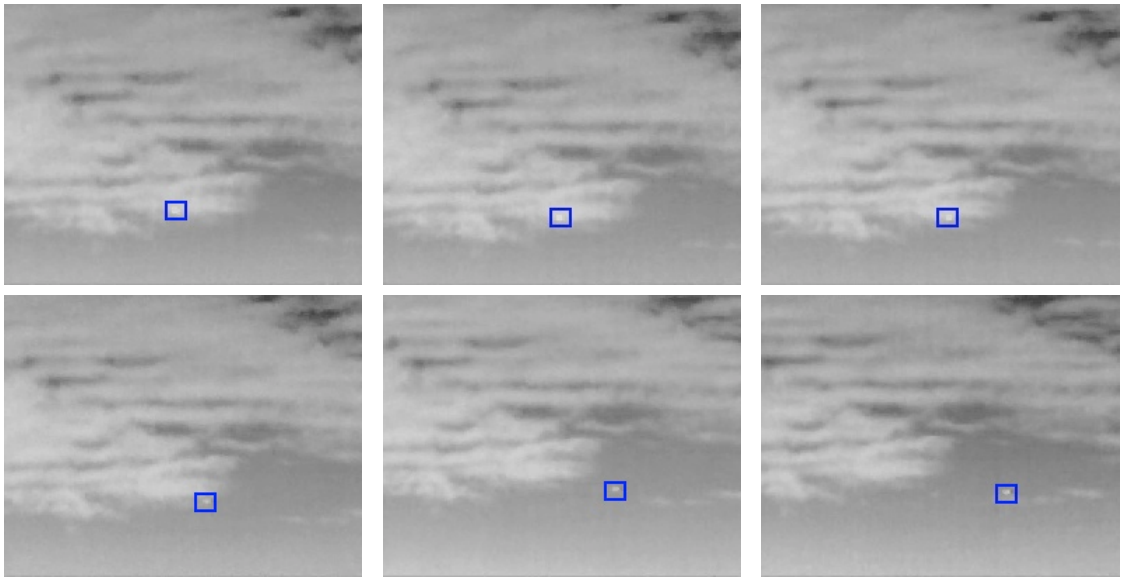
**Fig. 3** Tracking results of Seq.1 by (a) the compressive tracking algorithm, (b) the $l_1$ tracker, (c) the proposed tracking method. Frames 9, 30, 51, 69, 90 and 134 are displayed.



(a)



(b)



©

**Fig. 4** Tracking results of Seq.2 by (a) the compressive tracking algorithm, (b) the $l_1$ tracker, (c) the proposed tracking method. Frames 8, 12, 24, 52, 88 and 120 are displayed.



(a)



(b)



(c)

**Fig. 5** Tracking results of Seq.3 by (a) the compressive tracking algorithm, (b) the $l_1$ tracker, (c) the proposed tracking method. Frames 250, 300, 350, 400, 410 and 420 are displayed.



(a)

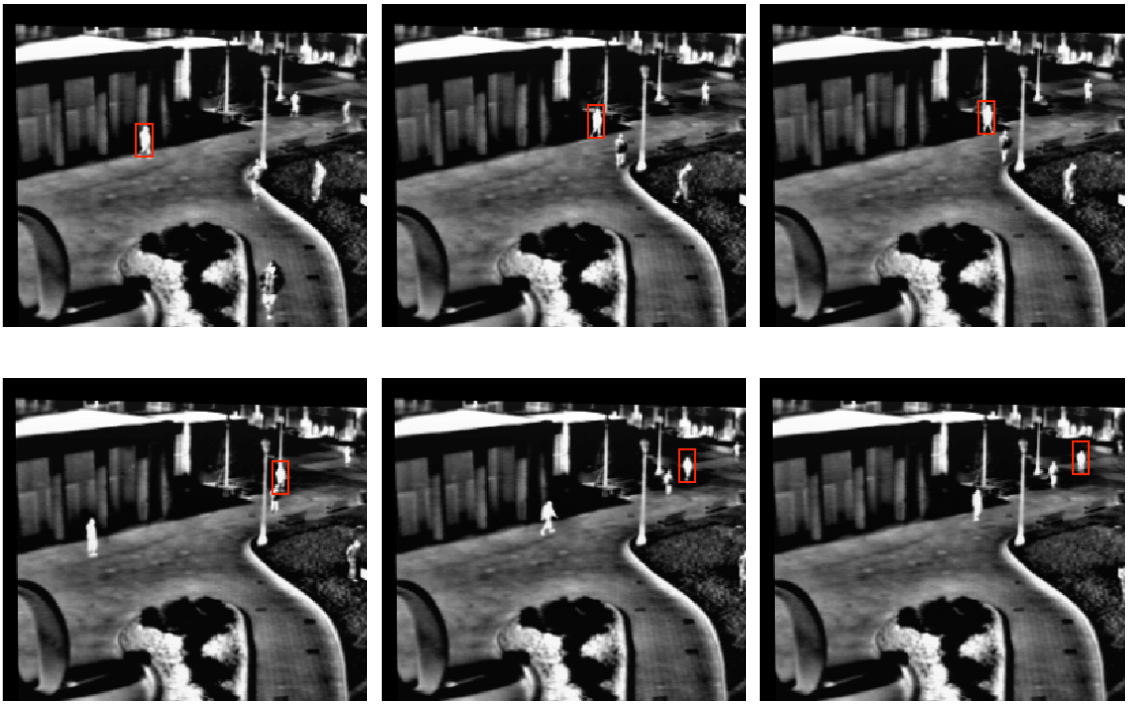


(b)

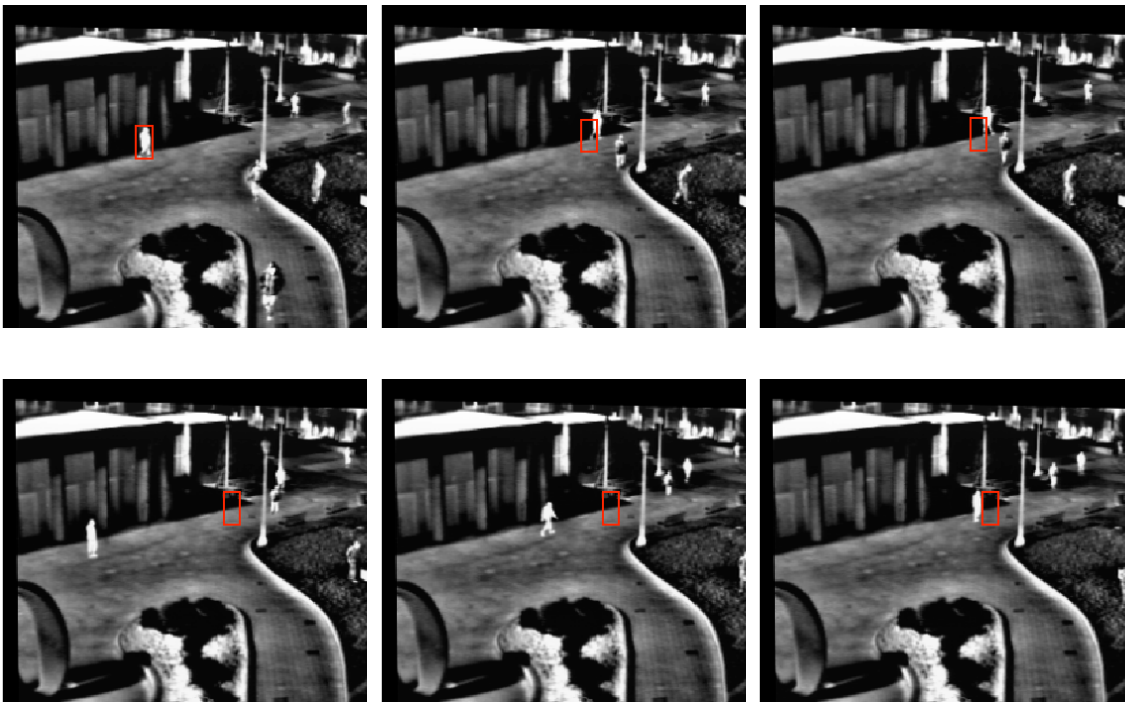


(c)

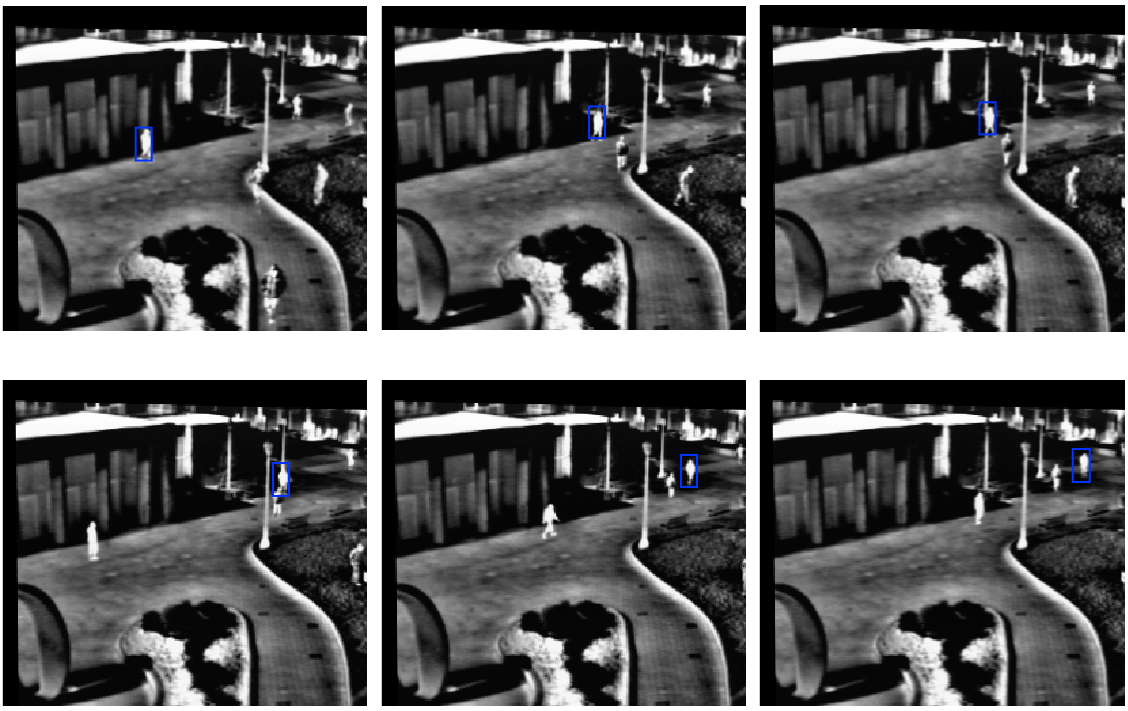
**Fig. 6** Tracking results of Seq.4 by (a) the compressive tracking algorithm, (b) the $l_1$ tracker, (c) the proposed tracking method. Frames 27, 59, 86, 107, 138 and 171 are displayed.



(a)



(b)



(c)

**Fig. 7** Tracking results of Seq.6 by (a) the compressive tracking algorithm, (b) the $l_1$ tracker, (c) the proposed tracking method. Frames 16, 135,159, 274, 358 and 424 are displayed.

Table 1 Mean distance to the ground truth

| Image sequence | $l_1$ tracker | Compressive tracking | Proposed |
|---|---|---|---|
| Seq.1 | 41.36 | 10.23 | **4.34** |
| Seq.2 | 58.54 | 50.49 | **3.72** |
| Seq.3 | 12.23 | 42.64 | **11.68** |
| Seq.4 | 47.45 | 46.34 | **4.12** |
| Seq.5 | 28.73 | 5.31 | **5.06** |

Table 2 Percentage of correctly tracked frames

| Image sequence | $l_1$ tracker | Compressive tracking | Proposed |
|---|---|---|---|
| Seq.1 | 17.51 | 92.37 | **100.00** |
| Seq.2 | 6.67 | 19.80 | **96.55** |
| Seq.3 | 94.50 | 70.62 | **96.00** |
| Seq.4 | 33.33 | 36.59 | **91.20** |
| Seq.5 | 18.70 | 93.60 | **94.68** |

Table 3 Average run time for one frame (in second)

| Image sequence | $l_1$ tracker | Compressive tracking | Proposed |
|---|---|---|---|
| Seq.1 | 1.2936 | **0.0741** | **0.0762** |
| Seq.2 | 1.2886 | **0.0632** | **0.0698** |
| Seq.3 | 1.2847 | **0.0602** | **0.0641** |
| Seq.4 | 1.2853 | **0.0617** | **0.0667** |
| Seq.5 | 1.2886 | **0.0602** | **0.0642** |

## C. Conclusion

In this paper, we have proposed a real-time infrared target tracking approach based on the use of $l_1$ minimization and compressive features. This work combines the real-time advantage of the compressive tracking algorithm and the robust performance of the $l_1$ tracker. Systematic experimental results on infrared image sequences demonstrate that the proposed approach outperforms both $l_1$ and compressive trackers.

Although generally performing well, the existing implementation occasionally encounters the target drift problem. An investigation into how this may be better addressed remains active research, together with building a more effective appearance model. Also, there have been other advanced techniques that may be adopted to perform feature selection in support of sparse representation-based target tracking. We intend to conduct such research next, especially to examine the potential benefits of utilizing those methods (e.g., [32], [33]) which have proven to be successful in complex image handling tasks [34]. In addition, we plan to exploit prior knowledge with online learning for more effective object tracking. Finally, it is important to note that this paper has focused on tracking of infrared targets in image sequences. However, there have been many different approaches that address visual target tracking in the literature. In particular, the work in [35] covers many important datasets and gives experimental comparative results of such interesting tracking algorithms. An investigation into how experience gained from these techniques may be used to strengthen our work will be undertaken in future.

## Acknowledgement

This work was supported by the Research Fund for the Doctoral Program of Higher Education of China under Grant 20126102110041, the Aeronautical Science Foundation of China under Grant 20125153025, and the Royal Academy of Engineering, UK, under Grant 1314RECI025.

## References


[1] G. Paravati, et al., "A genetic algorithm for target tracking in FLIR video sequences using intensity variation function," *IEEE Transactions on Instrumentation and Measurement*, **58**, 3457-3467(2009).

[2] A. Bal, M.S. Alam, "Automatic target tracking in FLIR image sequences using intensity variation function and template model," *IEEE Transactions on Instrumentation and Measurement*, **54**, 1846-1852 (2005).

[3] J. Ling, E. Liu, H. Liang, and J. Yang, "Infrared target tracking with kernel-based performance metric and eigenvalue-based similarity measure," *Applied Optics*, **46**, 3239-3252 (2007).

[4] Z. Wang, Q. Hou, and L. Hao, "Improved infrared target-tracking algorithm based on mean shift," *Applied Optics*, **51**, 5051-5059 (2012).

[5] F. Lamberti, A. Sanna, and G. Paravati, "Improving Robustness of Infrared Target Tracking Algorithms Based on Template Matching," *IEEE Transactions on Aerospace and Electronic Systems*, **47**, 1467-1480 (2011).

[6] H. Li, Y. Wei, L. Li, and Y. Y. Tang, "Infrared moving target detection and tracking based on tensor locality preserving projection," *Infrared Physics and Technology*, **53**, 77-83 (2010).

[7] M. Foursa, "Real-time infrared tracking system for virtual environments," In *proceedings of the 2004 ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry* (ACM, 2004), pp. 427-430.

[8] X. Wang, L. Liu, and Z. Tang, "Infrared human tracking with improved mean shift algorithm based on multicue fusion," *Applied Optics*, **48**, 4201-4212 (2009).

[9] R. Liu, E. Liu, J. Yang, Y. Zeng, F. Wang, and Y. Cao, "Automatically detect and track infrared small targets with kernel Fukunaga-Koontz transform and Kalman prediction," *Applied Optics*, **46**, 7780-7791 (2007).

[10] R. Liu, Y. Lu, "Infrared target tracking in multiple feature pseudo-color image with kernel density estimation," *Infrared Physics and Technology*, ***55***, 505-512 (2012).

[11] X. Mei, and H. Ling, "Robust visual tracking using $l_1$ minimization," In *Proceedings of IEEE International Conference on Computer Vision* (IEEE, 2009), pp. 1436-1443.

[12] D. Wang, H. Lu, and M. H. Yang, "Online Object Tracking with Sparse Prototypes," IEEE Transactions on Image Processing, **22**, 314-325 (2013).

[13] X. Mei, and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, **33**, 2259-2272 (2011).

[14] X. Jia, H. Lu, M-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," In *Proceeding of IEEE Conference on Computer Vision and pattern Recognition* (IEEE, 2012), pp. 1822-1829.

[15] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient $l_1$ tracker with occlusion detection," In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2011), pp. 1257-1264.

[16] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real-time robust $l_1$ tracker using accelerated proximal gradient approach," In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2012), pp. 1830-1837.

[17] P. Li, T. J. Hastie, and K. W. Church, “Very sparse random projections,” In *Proceedings of International Conference on Knowledge Discovery and Data Mining* (2006), pp. 287-296.

[18] R. Jensen, Q. Shen, “Are more features better?” *IEEE Transactions on Fuzzy Systems*, **17**, 1456-1458 (2009).

[19] K. Zhang, L. Zhang, and M. H. Yang, “Real-Time Compressive Tracking,” In *Proceedings of European Conference on Computer Vision* (2012), pp.864-877.

[20] P. Viola, M. Jones, “Rapid object detection using a boosted cascade of simple features,” In *Proceedings of IEEE International Conference on Computer Vision* (IEEE, 2001), pp. 511-518.

[21] S. Li, Z. Zhang, “Floatboost learning and statistical face detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**, 1-12 (2004).

[22] S. K. Zhou, R. Chellappa, and B. Moghaddam, “Visual tracking and recognition using appearance-adaptive models in particle filters,” *IEEE Transactions on Image Processing*, **13**, 1491-1506 (2004).

[23] J. Wright, Y. Ma, J. Maral, G. Sapiro et al, “Sparse representation for computer vision and pattern recognition,” Proceedings of IEEE, **98**, 1031-1044 (2010).

[24] S. Avidan, “Support vector tracking,” IEEE Transactions on Pattern Analysis and Machine Intelligence. **26**. 1064-1072 (2004).

[25] D. L. Donoho, “Compressed sensing,” IEEE Transactions on Information Theory, **52**, 1289-1306 (2006).

[26] H. Grabner, C. Leistner, H. Bischof, “Semi-supervised on-line boosting for robust tracking,” In *Proceedings of the 10th European Conference on Computer Vision*, pp. 234-247 (2008).

[27] H. Li, C. Shen, Q. Shi, "Real-time visual tracking using compressive sensing," In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2012), pp. 1305-1312.

[28] L. Matthews, T. Ishikawa, and S. Baker, "The template update problem," IEEE Transactions on Pattern Analysis and Machine Intelligence, **26**, 810-815 (2004).

[29] http://www.pudn.com/downloads108/sourcecode/graph/texture_mapping/detail447844.html

[30] VIVID database. http://vision.cse.psu.edu/data/vividEval/datasets/datasets.html

[31] OSU database. http://vcipl.okstate.edu/otcbvs/bench/Data/01/download.html

[32] R. Jensen, Q. Shen, "New approaches to fuzzy-rough feature selection," *IEEE Transactions on Fuzzy Systems*, **17**, 824-838 (2009).

[33] R. Diao, Q. Shen, "Feature selection with harmony search," *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, **42**, 1509-1523 (2012).

[34] C. Shang, D. Barnes, "Fuzzy-rough feature selection aided support vector machines for Mars image classification," *Computer Vision and Image Understanding*, **117**, 202–213 (2013).

[35] Y. Wu, J. Lim, M.H. Yang, "Online object tracking: A benchmark," In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2013), pp. 2411-2418.

# Brief Biographies of Authors

**Ying Li** received the Ph. D degree from the Xidian University, Xi'an, China, in 2002. She was a Postdoctoral Researcher with the School of Computer Science, Northwestern Polytechnical University, Xi'an, China from Feb. 2003 to Apr. 2005.

Currently, she is a professor at the School of Computer Science, Northwestern Polytechnical University. Her research interests include image processing, machine learning and signal processing. She is (co-)author of 1 research monograph and around 100 peer-reviewed papers.

**Pengcheng Li** received the B.S. degree in statistics from Northwestern Polytechnical University, Xi'an, China, in 2012. He is currently pursuing the M.S. degree in the School of Computer, Northwestern Polytechnical University, Xi'an, China.

**Qiang Shen** holds the established Chair in Computer Science and is the Director of the Institute of Mathematics, Physics and Computer Science at Aberystwyth University, UK. He has a PhD in Knowledge-Based Systems and a DSc in Computational Intelligence. His research interests include: computational intelligence, reasoning under uncertainty, pattern recognition, data mining, and their applications for intelligent decision support. Prof. Shen is a long-serving associate editor of two premier IEEE Transactions (Cybernetics and Fuzzy Systems), and an editorial board member for several other leading international journals. He has authored 2 research monographs, and around 340 peer-reviewed papers, including one receiving an IEEE Outstanding Transactions Paper award.