

## Research Article

# Multifeatures Based Compressive Sensing Tracking

**Liang He, Yuming Bo, and Gaopeng Zhao**

*Department of Automation, Nanjing University of Science and Technology, Nanjing 210000, China*

Correspondence should be addressed to Liang He; [brown19860526@gmail.com](mailto:brown19860526@gmail.com)

Received 19 March 2014; Revised 30 April 2014; Accepted 8 May 2014; Published 21 May 2014

Academic Editor: Qingsong Xu

Copyright © 2014 Liang He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To benefit from the development of compressive sensing, we cast tracking as a sparse approximation problem in a particle filter framework based on multifeatures. In this framework, the target template is composed of multiple features extracted from visible and infrared frames; in addition, occlusion, interruption, and noises are addressed through a set of trivial templates. With this model, the sparsity is achieved via a compressive sensing approach without nonnegative constraints; then the residual between sparsity representation and the compressed sensing observation is used to measure the likelihood which weights particles. After that, the target template is adaptively updated according to the Bhattacharyya coefficients. Some experimental results demonstrate that the proposed tracker appears to have better robustness compared with four different algorithms.

## 1. Introduction

Visual tracking is an essential task in computer vision. It is applicable in many fields such as vehicle tracking, medical imaging, robotics, and surveillance. Many efforts have been paid in developing a robust visual tracking algorithm to overcome the challenges when occlusion, illumination changes, viewpoints variation, or noise interruption occurs [1–3].

With the help of sparse representation technique, [4] proposed the L1 tracker to perform a robust visual tracking, casting tracking as a sparse approximation problem under a particle filter framework. This approach shows better performance in dealing with occlusions, pose changes, and illumination changes, when compared with mean shift tracker, covariance tracker, and appearance adaptive particle filter. But it is computationally demanding because it needs to solve an  $l_1$  norm related minimization problem for many times. Hereafter many efforts are paid to accelerate tracking progress and improve robustness based on it: [5] developed a real-time compressive sensing tracking (RTCST) algorithm to reduce the computational complexity; meanwhile [6] promoted an accelerated proximal gradient approach instead of the interior point method for acceleration; the result shows that both algorithms perform with higher accuracy and robustness than standard L1 tracker in many complex scenarios but are still insufficient in accommodating extreme illumination

variations. A main reason causing this insufficiency may be that both algorithms directly use the tailored target image to generate the target templates, which is not sensitive to environmental changes; besides, it will be unstable when tracking target is similar to background. To improve robustness and accuracy, a multifeature based method is employed in our approach, which leads us to treat the tracking target as a sparse representation in the linear span of multifeature space.

Actually, the multifeature methods are widely used in image fusion and face recognition; many attempts are taken to extend them to deal with tracking problems [7, 8]. A mixture of infrared and visible features is a typical method for modeling the tracking object. Reference [9] uses the mixed visible and infrared features and combines a mean shift method with the level set evolution algorithm to track visual objects; [10] employs the intensity cue and edge cue of infrared target as the feature template and applies a particle filter framework to track with pedestrian. All these algorithms could successfully cope with the complex environmental cases such as illumination change, shadow, and occlusion but should be improved to handle challenges under more severe conditions. To benefit from these previous works, we use some of the features mentioned above, together with a modified compressed sensing tracking method to receive a more robust tracking result.

The remainder of the paper is arranged as follows. We briefly review some related works in Section 2. Section 3 details the proposed tracker, including a sparse target template representation model, a compressive sensing tracking strategy, and an adaptive template update scheme. Experimental results and conclusions are given in Sections 4 and 5.

## 2. Related Work

**2.1. Particle Filtering.** The particle filter is a sequential Monte Carlo sampling method for Bayesian filtering; it provides a convenient framework for estimating the posterior distribution of state variables characterizing a dynamic system which is nonlinear and non-Gaussian. Assume that  $s_t$  is the state variable at time  $t$  which can be defined as affine motion or any other parameters reflecting the properties of the system. A predicting distribution of  $s_t$  given all available observations  $y_{1:t-1}$  up to time  $t - 1$  can be computed as

$$p(s_t | y_{1:t-1}) = \int p(s_t | s_{t-1}) p(s_{t-1} | y_{1:t-1}) ds_{t-1}. \quad (1)$$

When observation  $y_t$  is available at time  $t$ , the state vector is updated using the Bayes rule

$$p(s_t | y_{1:t}) = \frac{p(y_t | s_t) p(s_t | y_{1:t-1})}{p(y_t | y_{1:t-1})} \quad (2)$$

in particle filter; the posterior  $p(s_t | y_{1:t})$  is approximated by a weighted sample set  $S = \{(s^{(n)}, \omega^{(n)}) | n = 1, \dots, N\}$ . Because it is generally impossible to sample from the state posterior  $p(s_t | y_{1:t})$  directly, an importance distribution  $q(s_t | s_{1:t-1}, y_{1:t})$  is used for updating the weights of the samples

$$\omega_t^i = \omega_{t-1}^i \frac{p(y_t | s_t^i) p(s_t^i | s_{t-1}^i)}{q(s_t | s_{1:t-1}, y_{1:t})} \quad (3)$$

and a typical value of importance function is  $q(s_t | s_{1:t-1}, y_{1:t}) = p(s_t | s_{t-1})$ ; in this situation, the weights become the observation likelihood  $p(y_t | s_t)$ . Then the samples are resampled to generate a set of equally weighted particles according to their importance weights to avoid degeneracy.

**2.2. Compressive Sensing.** Assuming that a signal  $\mathbf{x} \in R^N$  can be sparsely expressed in a basis of  $N \times 1$  vectors

$$\mathbf{x} = \sum_{i=1}^N s_i \psi_i \quad \text{or} \quad \mathbf{x} = \Psi \mathbf{s} \quad (4)$$

the signal  $\mathbf{x}$  is  $K$ -sparse if  $\|\mathbf{s}\|_0 = K$ , in which  $\|\mathbf{s}\|_0$  counts the nonzero elements in vector  $\mathbf{s}$ . The compressive sensing theory demonstrates that  $\mathbf{x}$  can be recovered with an overwhelming probability by using few measurements:

$$\mathbf{y} = \Phi \mathbf{x} = \Phi \Psi \mathbf{s} = \Theta \mathbf{s}, \quad (5)$$

where  $\Phi$  is incoherent with  $\Psi$  and  $\Theta$  satisfies the RIP (restricted isometry property) [11].

Many studies indicate that the Gaussian measurement matrix  $\Phi$  is stable to protect the salient information of any  $K$ -sparse or compressible signal in dimensionality reduction. With a small constant  $c$ , an  $M \times N$  iid Gaussian matrix  $\Phi$  can be shown to have the RIP with high probability if  $M \geq cK \log(N/K)$ ; besides, for  $\Phi$  being universal,  $\Theta = \Phi\Psi$  will be iid Gaussian, thus having the RIP regardless of the choice of  $\Psi$  [12].

Since directly obtain  $\mathbf{s}$  (which equals to  $\mathbf{x}$  in the  $\Psi$ -domain) via

$$\hat{\mathbf{s}} = \arg \min \|\mathbf{s}\|_0 \quad \text{s.t. } \Theta \mathbf{s} = \mathbf{y} \quad (6)$$

is an NP-complete problem, it is commonly relaxed to solve the optimization

$$\hat{\mathbf{s}} = \arg \min \|\mathbf{s}\|_1 \quad \text{s.t. } \Theta \mathbf{s} = \mathbf{y}. \quad (7)$$

When considering noises, the modified optimization

$$\hat{\mathbf{s}} = \arg \min \|\mathbf{s}\|_1 \quad \text{s.t. } \|\Theta \mathbf{s} - \mathbf{y}\|_2 \leq \varepsilon, \quad (8)$$

where  $\varepsilon$  is a prespecified threshold.

**2.3.  $l_1$  Tracker and Real-Time Compressive Sensing Tracking.** The  $l_1$  tracker considers the tracking target as a sparse representation in the linear span of target template set, along with noises. Given target template set  $T = [\mathbf{t}_1 \cdots \mathbf{t}_n] \in R^{d \times n}$  ( $d \gg n$ ) and trivial templates  $I = [\mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_d] \in R^{d \times d}$ , a tracking result  $\mathbf{y} \in R^d$  can be modeled as

$$\mathbf{y} = [T, I, -I] \begin{bmatrix} \mathbf{a} \\ \mathbf{e}^+ \\ \mathbf{e}^- \end{bmatrix} \triangleq B\mathbf{c}, \quad \text{s.t. } \mathbf{c} \geq 0, \quad (9)$$

where  $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n)^T \in R^n$  is a target coefficient vector and  $\mathbf{e}^+ \in R^d$ ,  $\mathbf{e}^- \in R^d$  are called positive and negative trivial coefficient vectors; with this definition, at time  $t$ , the coefficient vector  $\mathbf{c}_t^i$  of each candidate target  $\mathbf{y}_t^i$  is gained by solving the  $l_1$  regularized least squares problem

$$\min_{\mathbf{c}} \frac{1}{2} \|\mathbf{y}_t^i - B\mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_1, \quad \mathbf{c} \geq 0, \quad (10)$$

where  $i = 1, \dots, N_s$  is the number of particles; then the residuals of particles  $\mathbf{r}_t^i = \|\mathbf{y}_t^i - T_t \mathbf{a}_i\|_2^2$  are imported into particle filter to estimate the tracking result [4].

Based on the model above, to reduce computational load, [6] modifies (10) to the following minimization model:

$$\min_{\mathbf{c}} \frac{1}{2} \|\mathbf{y} - B'\mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_1 + \frac{\mu_t}{2} \|\mathbf{e}\|_2^2, \quad \mathbf{a} \geq 0, \quad (11)$$

where  $B' = [T, I]$ ,  $\mathbf{c} = [\mathbf{a}, \mathbf{e}]^T$ ,  $\mathbf{e} = [\mathbf{e}^+, \mathbf{e}^-]^T$ , and  $\mu_t$  is a parameter controlling energy in trivial templates; after that, an accelerated proximal gradient approach is proposed for solving problem (11). At the same time, RTCST uses compressive sensing to deal with the extremely high dimension of feature space in  $l_1$  tracker. Given the measurement matrix  $\Phi$ , we get the measurement vector  $\mathbf{z} = \Phi \mathbf{y} = \Phi B\mathbf{c}$ ; then a sparse coefficient vector  $\mathbf{c}$  can be recovered from

$$\min \|\mathbf{c}\|_1 \quad \text{s.t. } \|\Phi B\mathbf{c} - \Phi \mathbf{y}\| \leq \varepsilon. \quad (12)$$

### 3. Multifeature Based Compressive Sensing Tracking

**3.1. Target Template Initialization.** Previous studies indicate that multifeature methods perform better robustness than methods employing single feature [13, 14]; moreover, good feature descriptors are helpful for stable tracking. This motivates us to employ a multifeature method to form the target template set. Suppose  $p^k(y)$ ,  $k = 1, \dots, N_f$  to be the  $k$ th feature extracted from target image whose central coordinate is  $y$ ; then an observation of the target can be characterized by a vector  $\mathbf{p}(y) = [p^1(y), p^2(y), \dots, p^k(y)]^T$ . In the first frame, after manually selecting the target, the features of target image  $q^k(y_0) = p^k(y_0)$ ,  $k = 1, \dots, N_f$ , are calculated to form a template  $\mathbf{t}_1 = [q^1(y_0), q^2(y_0), \dots, q^k(y_0)]^T$ ; for initialization, we first perturb the center of target image  $y_0$  to generate candidate image center  $y_i$ ,  $i = 1, \dots, N_t$ , crop candidates with the same size as target image according to  $y_i$ , extract features of each candidate to form the rest templates  $\mathbf{t}_i = [q^1(y_i), q^2(y_i), \dots, q^k(y_i)]^T$ ,  $k = 1, \dots, N_f$ , and finally initialize the template set as  $T = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_i]$ ,  $i = 1, \dots, N_t$ . Here we use a combination of visible and infrared features to improve robustness of the tracking algorithm; the mainly used visible feature is the RGB histogram and local binary patterns (LBP); in addition, the main infrared feature used is the intensity histogram.

**3.1.1. RGB and Intensity Histogram.** Histogram is an estimate of the probability distribution of a continuous variable; here we employ an alternative histogram extracting approach where pixels further away from the region center are assigned smaller weights. The weighting function (also can be called kernel function) is

$$k(r) = \begin{cases} 1 - r^2, & r < 1 \\ 0, & \text{otherwise,} \end{cases} \quad (13)$$

where  $r$  is the distance from region center; then the histogram at location  $y_k$  can be described as

$$h_{i,y}(u) = C \sum_{i=1}^m k\left(\left\|\frac{y_i - y_0}{h}\right\|^2\right) \delta(b(y_i) - u); \quad (14)$$

$C = [\sum_{i=1}^m k(\|(y_i - y_0)/h\|^2)]^{-1}$  here is a normalization factor,  $\{y_i\}$ ,  $i = 1, \dots, m$  is the current coordinates of pixels in candidate target, and  $h = \sqrt{h_x^2 + h_y^2}$  is the bandwidth of kernel function.  $\delta$  is a Kronecker delta function, where  $u = 1, \dots, B_I$  is the histogram bin and  $b(y_i)$  associates pixel  $y_i$  with the histogram bin according to the corresponding RGB or gray intensity.

**3.1.2. LBP.** LBP (local binary pattern) is a type of feature characterizing the texture spectrum of an image; it is proved to be a powerful feature for texture classification. The original LBP feature first divides the examined window into cells and then, for each pixel  $x_0$  in a cell, compares it to each of its 8

neighbors  $\{x_i\}$ ,  $i = 1, \dots, 7$ ; the LBP value can be calculated as

$$\text{LBP}(x_0) = \sum_{i=1}^7 s(x_i - x_0) 2^i, \quad \text{s.t. } s(x) = \begin{cases} 1, & x > 0 \\ 0, & x < 0. \end{cases} \quad (15)$$

After that, a histogram is computed over the cell to generate the feature vector for the window.

**3.2. Compressive Sensing Tracking Based on Multifeatures.** As done in [5, 6], we apply affine image warping to model the object motion of two consecutive frames; the state variable  $s_t$  is defined to be  $s_t = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, t_x, t_y)$  which combine deformation parameters  $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$  with 2D translation parameters  $(t_x, t_y)$ ; the state transition distribution  $p(s_t | s_{t-1})$  is modeled as a Gaussian distribution.

In  $l_1$  tracker and RTCST, the dictionary  $B$  is composed of target template  $T$ , positive trivial template  $I$ , and negative trivial template  $-I$ ; meanwhile the coefficient vector is divided into target coefficient vector  $\mathbf{a}$ , positive trivial coefficient vector  $\mathbf{e}^+$ , and negative trivial coefficient  $\mathbf{e}^-$  to enforce nonnegativity constraints. The main reason of exploiting nonnegativity constraints into account is that they help filter out clutter which is similar to target templates but in reversed intensity patterns; furthermore, it is normal that the target image in current frame appears to be more like that in previous frame. Since multifeatures have the potential of filtering clutter and occlusion, the nonnegativity constraints can be relaxed; besides, an extra occlusion handling method becomes less necessary; for example, when tracking with pedestrian, the occlusion may occur in visible image but not in related infrared image; in other words, infrared features help filter out the occlusions. While the target template set  $T = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_{N_t}]$  is available after initialization or update, we model the observation as

$$\mathbf{z} = [T, I] \begin{bmatrix} \mathbf{x}_t \\ \mathbf{x}_e \end{bmatrix} \triangleq \mathbf{A}\mathbf{x}, \quad (16)$$

where  $I$  is the trivial template representing noises along with the trivial coefficient vector  $\mathbf{x}_e$  and  $\mathbf{x}_t$  is the target coefficient vector.

Under a certain threshold more weak correlated features will increase robustness and stability but will take more computational cost. To deal with this problem, we adopt the compressed sensing theory. Choosing random Gaussian matrix  $\Phi \in R^{m \times n}$ ,  $\phi_{i,j} \sim \mathcal{N}(0, 1)$  as the measurement matrix, we take the measurements  $\mathbf{z}' = \Phi\mathbf{z} = \Phi\mathbf{A}\mathbf{x}$ ; then by using orthogonal matching pursuit (OMP), we recover the coefficient vector  $\mathbf{x}$  from

$$\min \|\mathbf{x}\|_1, \quad \text{s.t. } \|\Phi\mathbf{A}\mathbf{x} - \Phi\mathbf{z}\|_2 \leq \varepsilon. \quad (17)$$

The main flow of multifeature based compressive sensing tracking (MFCST) algorithm is presented in Algorithm 1.

**3.3. Template Update.** Intuitively, as time goes by, a static template will not be able to exactly capture the appearance

**Input:**

- (i) Current visible frame  $I_v^k \in R^{h \times w}$  and infrared frame  $I_{ir}^k \in R^{h \times w}$
  - (ii) Particles  $s_{k-1}^i$ ,  $i = 1, 2, \dots, N_s$
  - (iii) Particles Measurement matrix  $\Phi \in R^{m \times n}$
  - (iv) Template set  $A = [T, I] \in R^{n \times (n+N_t)}$
  - (v) Preset parameter  $\lambda > 0$
- (1) Normalize each column of  $\Phi A$ ;
  - (2) Generate particles  $s_k^i$ ,  $i = 1, 2, \dots, N_s$  according to state transition distribution;
  - (3) **for**  $i = 1 \rightarrow N_s$  **do**
  - (4) Obtain target candidates  $\mathbf{g}_i$  and relevant candidate center  $y_i$  according to  $s_k^i$ ;
  - (5) Calculate RGB histogram  $\mathbf{p}_{\text{RGB}}$ , LBP  $\mathbf{p}_{\text{LBP}}$  of candidate in  $I_v^k$  and intensity histogram  $\mathbf{p}_{\text{gray}}$  of candidate in  $I_{ir}^k$ ;
  - (6) Stack  $\mathbf{p}_{\text{LBP}}$ ,  $\mathbf{p}_{\text{gray}}$  and  $\mathbf{p}_{\text{RGB}}$  into an observation  $\mathbf{z}_i = [\mathbf{p}_{\text{LBP}}, \mathbf{p}_{\text{gray}}, \mathbf{p}_{\text{RGB}}]^T$ ;
  - (7) Get measurements  $\mathbf{z}' = \Phi \mathbf{z}_i$ ;
  - (8) Solving (17) to get coefficient vector  $\mathbf{x}_i$ ;
  - (9) Calculate residual  $\mathbf{r}_i = \|\Phi \mathbf{z}_i - \Phi A \mathbf{x}_i\|_2$ ;
  - (10) Obtain observation likelihood  $l_i = \exp(-\lambda \mathbf{r}_i)$ ;
  - (11) **end**
  - (12) Resample the particles  $s_k^i$  according to  $l_i$ ;
  - (13) Using Maximum A Posterior (MAP) or Mean Square Error (MSE) to estimate the state  $\mathbf{s}_k$  of current frame, which characterizes the current tracking target  $\mathbf{g}$ ;
  - (14) Get target observation  $\hat{\mathbf{z}}$  corresponding to current tracking result  $\mathbf{g}$ ;
  - (15) Recalculate  $\hat{\mathbf{x}}$  for  $\hat{\mathbf{z}}$  by solving (17);
  - (16) Update target template  $T$ ;
- Output:**
- (i) Tracked target  $\mathbf{g}$
  - (ii) Updated target dynamic state  $\mathbf{s}_k$
  - (iii) Updated target template set  $T$

ALGORITHM 1: Multifeature based compressive sensing tracking.

variations of target due to illumination or pose changes; otherwise, frequently updating the template will accumulate errors and drift the tracker from target. We tackle this problem by adaptively updating the target template. Notice that  $\mathbf{x}_i$  can be seen as a sparse representation of  $\mathbf{z}$  in the linear span of  $T$ , which means that the bigger  $x_i$  is, the more important  $\mathbf{t}_i$  will be. The Bhattacharyya coefficient is applied to measure the similarity between target observation  $\hat{\mathbf{z}}$  and the template  $\mathbf{t}_m$  which has maximum coefficient  $x_m$ , which can be defined as

$$\rho(\hat{\mathbf{z}}, \mathbf{t}_m) = \sum_{i=1}^n \sqrt{\hat{z}_i t_{m_i}}; \quad (18)$$

a typical value of  $\rho(\cdot)$  is between (0, 1); in addition, bigger  $\rho(\hat{\mathbf{z}}, \mathbf{t}_m)$  means that  $\hat{\mathbf{z}}$  is more similar to  $\mathbf{t}_m$ . We adopt two thresholds  $\tau_2 > \tau_1$  to guide the updating progress, where  $\rho(\hat{\mathbf{z}}, \mathbf{t}_m) < \tau_2$  indicates that  $\hat{\mathbf{z}}$  is not so similar to  $\mathbf{t}_m$ , which reflects variations in tracking; moreover, when  $\rho(\hat{\mathbf{z}}, \mathbf{t}_m)$  decreases under a certain threshold that is  $\rho(\hat{\mathbf{z}}, \mathbf{t}_m) < \tau_1$ , we consider it as a strong interference occurring in the tracking progress. The template update scheme is summarized in Algorithm 2.

#### 4. Experiments

We test our algorithm on many real-world visible and infrared video sequences obtained from the data set 03

OSU Color-Thermal Database of the OTCBVS Benchmark Dataset [15]; three scenarios of two videos are involved in our experiments. To evaluate the performance of the proposed tracking framework, we compare the tracking results of our proposed multifeature based compressive sensing tracking (MFCST) with the adaptive multicue particle filter (AMC-PF) [10], the tracking method based on infrared and visible dual-channel video (IVDT) [9], the  $l_1$  tracker [4], and the real-time compressive sensing tracking (RTCST) [5]. All targets are manually marked in the first tracking frame without careful selection.

The test sequences used in our first and second experiments S1 and S2 are based on location 2 in the data set. In S1, two pedestrians intersect with each other, which causes a partial occlusion. From the visible frames, we can see that the whole scene is under well illumination; two pedestrians are distinct from both each other and the background but in corresponding infrared frames; these two pedestrians are similar to each other. Some tracking results are given in Figure 1, where the frame indexes are labeled in the upper left corner of the images. It can be observed that although all trackers can well follow the target pedestrian, AMC-PF, RTCST, and MFCST do more precise work than those of  $l_1$  tracker and IVDT.

The second experiment S2 is based on the scene in which target pedestrian passes by a static car. The target pedestrian has similar appearance as the front part and tire of the car in both visible and corresponding infrared frames. Results

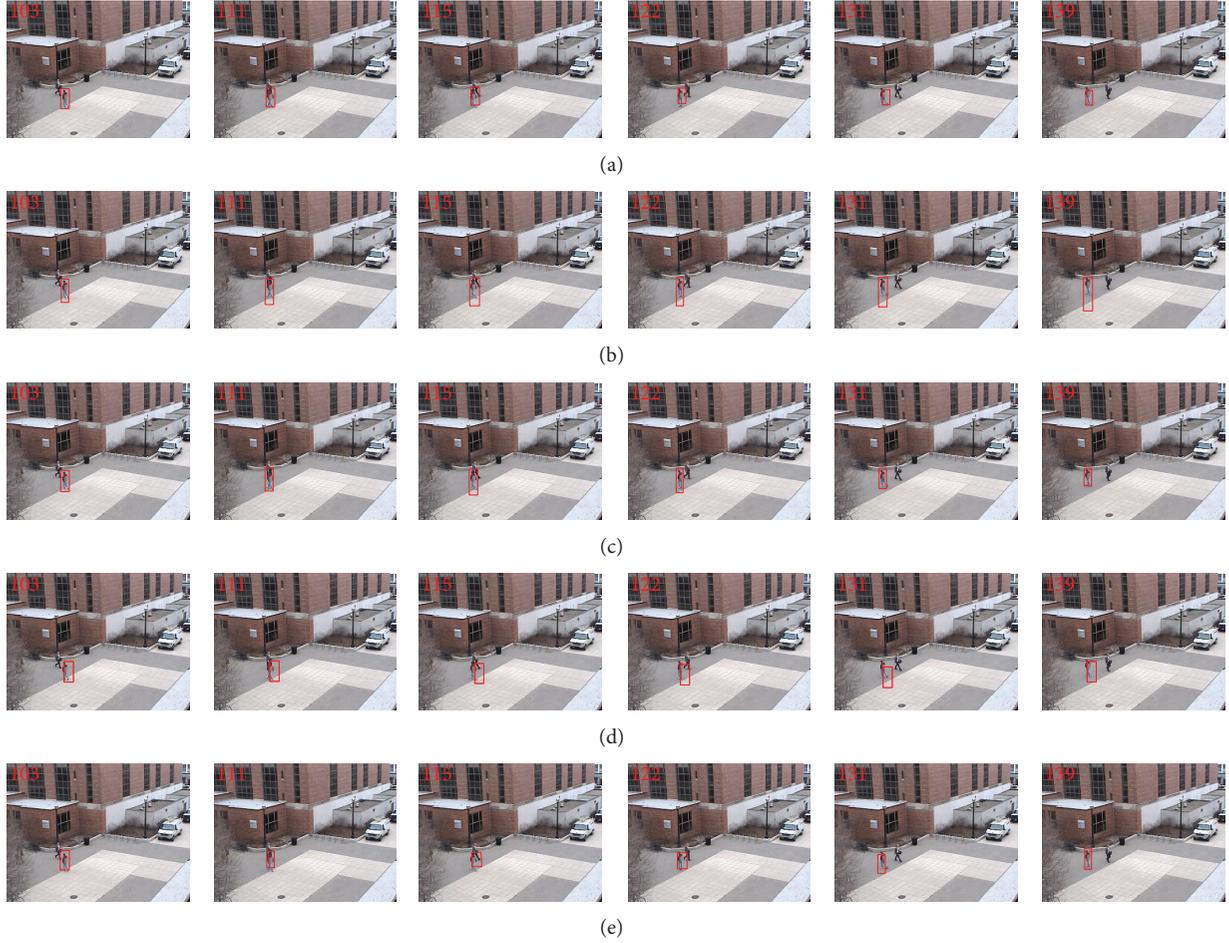


FIGURE 1: Tracking results of SI: AMC-PF (a), the  $l_1$  tracker (b), RTCST (c), IVDT (d), and our proposed MFCST (e).

**Input:**

- (i) Newly tracking result  $\mathbf{g}$  and the corresponding observation  $\tilde{\mathbf{z}}$
- (ii) Target coefficient vector  $\tilde{\mathbf{x}}_t$
- (iii) Template set  $A = [T, I] \in R^{n \times (n+N_t)}$ , where  $T = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_{N_t}] \in R^{n \times N_t}$
- (iv) Bhattacharyya thresholds  $\tau_1, \tau_2, 1 > \tau_2 > \tau_1 > 0$

- (1)  $m \leftarrow \arg \max_{1 \leq i \leq N_t} x_i$ ;
- (2) **if**  $\tau_2 > \rho(\tilde{\mathbf{z}}, \mathbf{t}_m) > \tau_1$  **then**
- (3)  $n \leftarrow \arg \min_{1 \leq i \leq N_t} x_i$ ;
- (4)  $\mathbf{t}_n \leftarrow \tilde{\mathbf{z}}$ ;
- (5) **end**

(6) Normalize every columns of  $T$ ;

**Output:** Updated template set  $A = [T, I]$

ALGORITHM 2: Template update.

in Figure 2 indicate that AMC-PF drifts to the car when target occludes the tire. This is because AMC-PF uses the intensity cue and edge cue of the infrared image; however, the appearance of pedestrian and the front part of the car in infrared frame is easily confused.

The third experiment S3 considers a more complex scene; we use a video sequence from location 1 in the OSU Color-Thermal Database and then test 5 trackers mentioned above from frame 220 to 429. Since it is hard to distinct target pedestrian from background, we manually labeled the

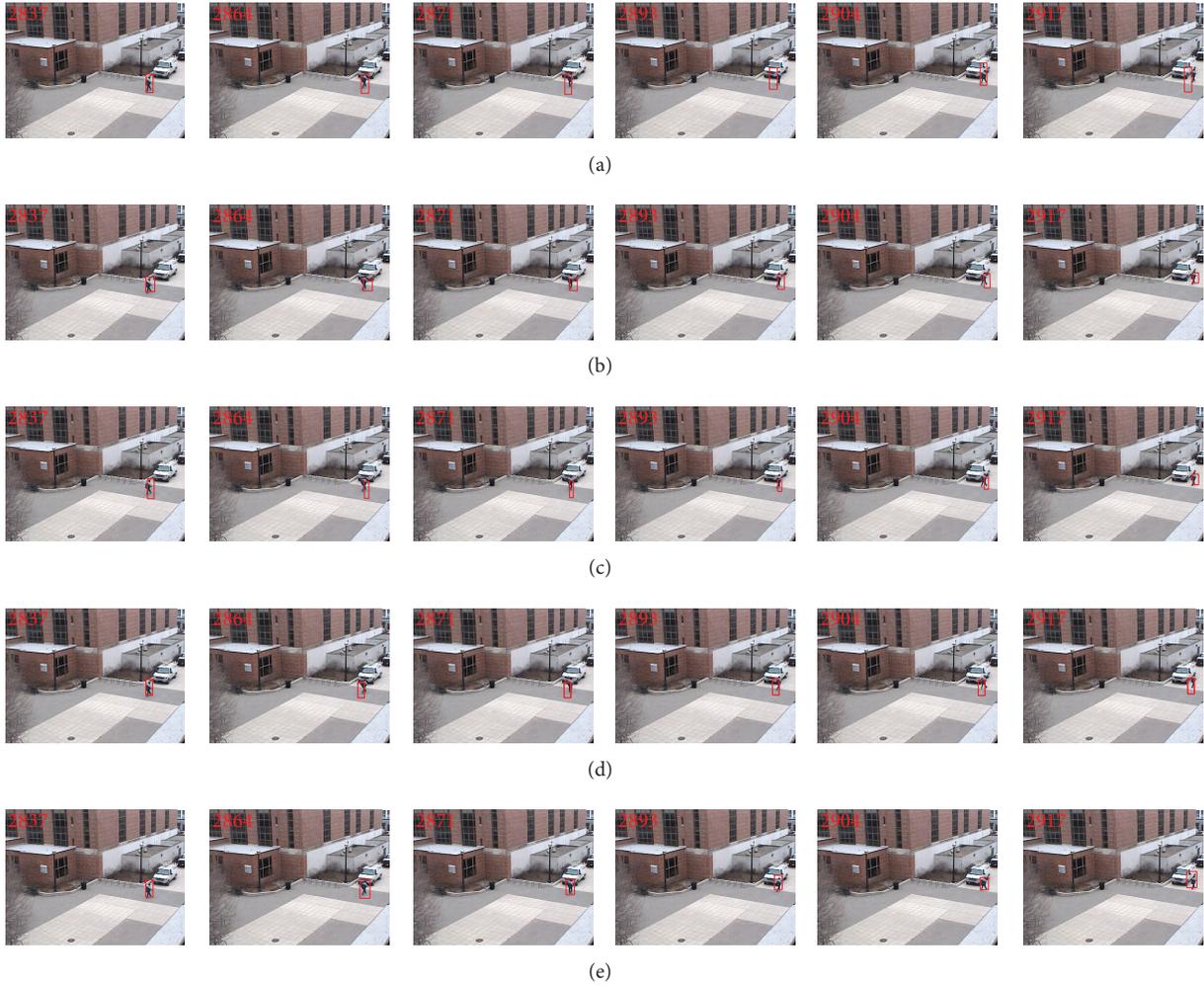


FIGURE 2: Tracking results of S2: AMC-PF (a), the  $l_1$  tracker (b), RTCST (c), IVDT (d), and our proposed MFCST (e).

tracking target in the first infrared frame (frame 220) at the beginning of our experiment S3, without careful selection. The whole video sequence can be described in 5 clips. In the first clip, which covers from starting frame 220 to frame 250, the target walking in the shadow is confused with background in visible frame but is distinct in the infrared frame. As target walks straightforward, in clip 2, from frame 250 to 322, the tracked pedestrian particle occludes the first street lamp, which we sign as lamp 1; then, as target walks out of shadow, the illumination conditions are drastically changed. A heavy occlusion appears in clip 3, from frame 322 to 374; when target passes by the second street lamp, street lamp 2 completely occludes the pedestrian both in visible and infrared image. After that, target is particle occluded by pedestrian 2, wearing a red coat, which is obviously different from target pedestrian in visible frame; after several frames, the target walks by pedestrian 2 and encounters the third street lamp; these sequences are described in clip 4, from frame 374 to 390. At last, in clip 5, from frame 390 to 427, the target walks into a square and finishes the whole tracking process. The performance of 5 methods is shown in Figure 3, from which we can see that the  $l_1$  tracker and RTCST failed

to track with target in clip 1; the main reason may be that these two algorithms only use pixel information of visible images; unfortunately, it is severely a hard work to distinguish target from background in these visible frames. AMC-PF drifts to street lamp 1 from clip 2, where the illumination is severely changed and lamp 1 occludes the target both in visible and infrared frames. The dual-channel method IVDT succeeded in clips 1 and 2 but began to lose its target from clip 3. Meanwhile our proposed MFCST accurately tracks the target pedestrian throughout the whole sequence; this indicates that our approach appears to have better robustness in dealing with heavy occlusions and drastic illumination changes. In addition, when comparing with multifeature tracking methods AMC-PF and IVDT, our approach takes less measurements; furthermore, comparing with  $l_1$  and RTCST, our algorithm receives better accuracy.

## 5. Conclusion

We model the tracking target as a sparse representation in the linear span of multifeature space. To generate the dictionary, we use RGB histogram and local binary patterns (LBP) of

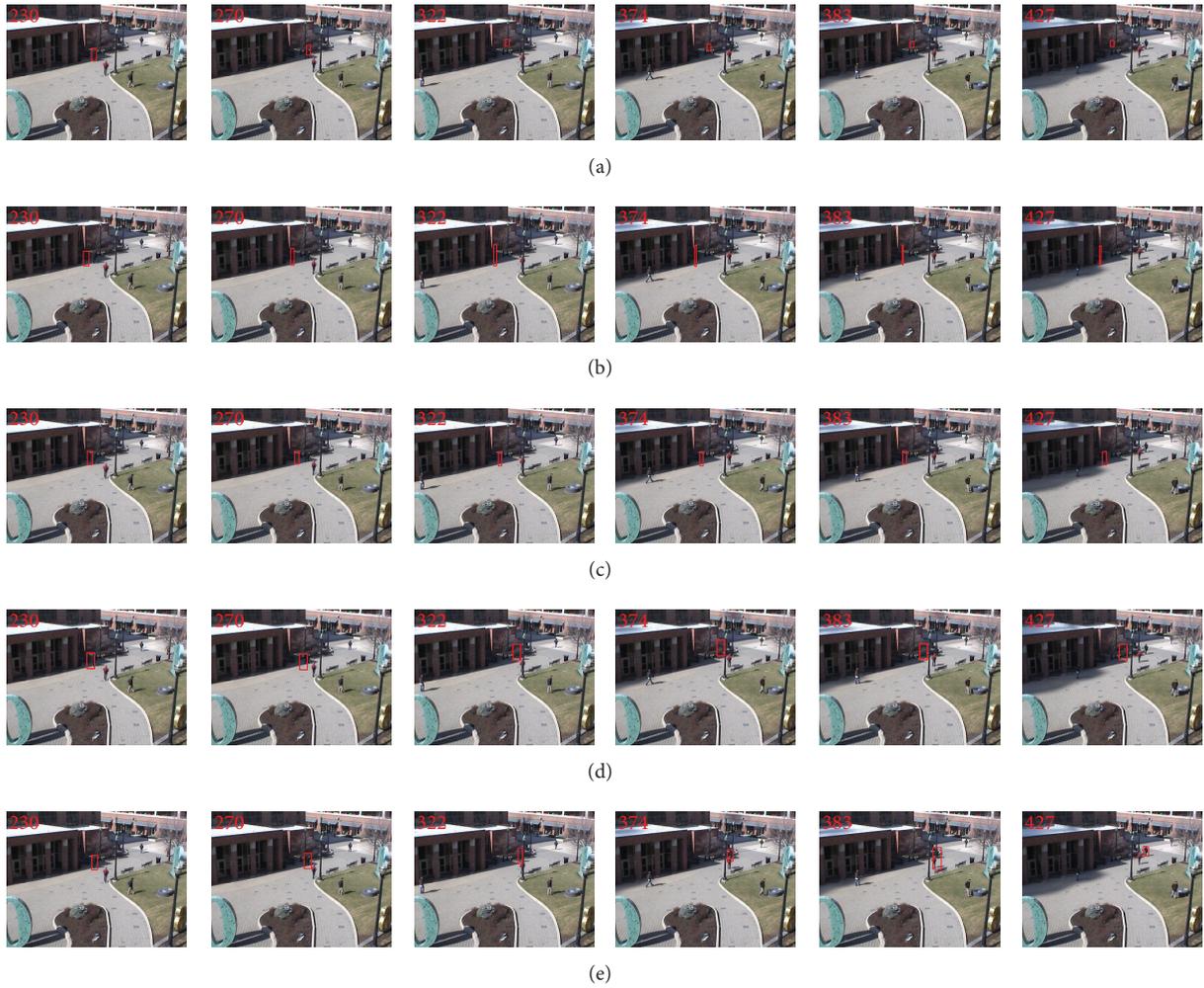


FIGURE 3: Tracking results of S3: AMC-PF (a), the  $l_1$  tracker (b), RTCST (c), IVDT (d), and our proposed MFCST (e).

the visible frames with intensity histogram of the infrared frames to create the target template  $T$  and then combine  $T$  with the trivial template  $I$ . A compressive sensing method is employed under particle filter framework to reduce the high dimension of the feature space and solve the coefficients for sparse representation. For further robustness, we introduce an adaptive template update scheme for this system. The proposed tracker receives robust and stable performance in dealing with occlusions and illumination variations.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### Acknowledgments

This work is supported by National Natural Science Foundation (NNSF) of China under Grant 61203266, Doctoral Foundation of the Ministry of Education of China under

Grant 20113219110027, and National Defense Pre-Research Foundation of China under Grant 40405020201.

### References

- [1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Computing Surveys*, vol. 38, no. 4, article 13, 2006.
- [2] C. Shen, J. Kim, and H. Wang, "Generalized kernel-based visual tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 1, pp. 119–130, 2010.
- [3] G. Zhao, Y. Bo, and Y. Chen, "A method for tracking object in infrared and visible image based on multiple features," *Acta Armamentarii*, vol. 32, no. 4, pp. 445–451, 2011.
- [4] X. Mei and H. Ling, "Robust visual tracking using  $L_1$  minimization," in *Proceedings of the 12th IEEE International Conference on Computer Vision*, pp. 1436–1443, IEEE, 2009.
- [5] H. Li, C. Shen, and Q. Shi, "Real-time visual tracking using compressive sensing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 1305–1312, IEEE, June 2011.
- [6] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust  $L_1$  tracker using accelerated proximal gradient approach," in *Proceedings*

of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12), pp. 1830–1837, IEEE, 2012.

- [7] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, “Recent advances and trends in visual tracking: a review,” *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, 2011.
- [8] J. H. Yoon, D. Y. Kim, and K.-J. Yoon, “Visual tracking via adaptive tracker selection with multiple features,” in *Computer Vision—ECCV 2012*, vol. 7575 of *Lecture Notes in Computer Science*, pp. 28–41, Springer, Berlin, Germany, 2012.
- [9] G. Zhao, Y. Bo, and M. Ying, “An object tracking method based on infrared and visible dual-channel video,” *Journal of Electronics & Information Technology*, vol. 34, no. 3, pp. 529–534, 2012.
- [10] X. Wang and Z. Tang, “Modified particle filter-based infrared pedestrian tracking,” *Infrared Physics and Technology*, vol. 53, no. 4, pp. 280–287, 2010.
- [11] E. J. Candès, “Compressive sampling,” in *International Congress of Mathematicians. Vol. III*, pp. 1433–1452, The European Mathematical Society, Zürich, Switzerland, 2006.
- [12] R. G. Baraniuk, “Compressive sensing [lecture notes],” *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–124, 2007.
- [13] P. Pérez, J. Vermaak, and A. Blake, “Data fusion for visual tracking with particles,” *Proceedings of the IEEE*, vol. 92, no. 3, pp. 495–513, 2004.
- [14] P. Brasnett, L. Mihaylova, D. Bull, and N. Canagarajah, “Sequential Monte Carlo tracking by fusing multiple cues in video sequences,” *Image and Vision Computing*, vol. 25, no. 8, pp. 1217–1227, 2007.
- [15] J. Davis and V. Sharma, Otcvbs benchmark dataset collection, 2007.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

