

Article

# Robust Object Tracking in Infrared Video via Adaptive Weighted Patches

Jiangtao Wang and Jingai Zhang \*

Department of Physical and Electronic Information, Huaibei Normal University, No. 100 Dongshan Street, Huaibei 235000, China; jangtaowang@chnu.edu.cn

\* Correspondence: ellazhangja@126.com; Tel.: +86-561-380-2229

Academic Editor: Fazal M. Mahomed

Received: 12 October 2016; Accepted: 23 December 2016; Published: 30 December 2016

**Abstract:** With the quick development of computer and electronic techniques, infrared sensor-based object tracking has become a hot research topic in recent years. However, infrared object tracking is still a challenging task due to low resolution, lack of representing information, and occlusion. In this work, we present an adaptive weighted patch-based infrared object tracking scheme. First, the candidate local region is divided into non-overlapping sub regions, and a set of belief weights is set on these patches. After this, a particle filtering-based infrared object tracking system is realized. In the last, the belief weight of each patch is evaluated based on the linear discriminative analysis (LDA) and particle sampling scheme. Experimental results on challenging infrared sequences show that the proposed algorithm can effectively locate the tracking object.

**Keywords:** infrared object tracking; linear discriminative analysis; particle filter; patch-based tracker

---

## 1. Introduction

Tracking an object in video sequences is a key function of the intelligent surveillance system [1–7], and it also plays important role in many other computer vision applications, including video compression, intelligent human computer interaction (HCI), and so on. During the last two decades, object tracking with visible camera has been well addressed [8–10]. However, this is not available for the nighttime case, for that the visible camera relies heavily on the light conditions. Compared with visible cameras, the infrared imaging system is more robust to the illumination change and can work with almost no difference between daytime and nighttime. Recently, due to the quick development of computer and electronic techniques, infrared sensors have been widely utilized from military to civil fields, and object tracking in infrared sequences has become a hot research topic.

Despite that the infrared sensing system can work all the time, the main disadvantage lies in that the information it acquires is not as rich as the one from a visible camera. A visible camera can acquire visible images with ample color information, which can describe the object precisely. In contrast, the infrared camera can only record the intensity information for the working scene. Hence, in the infrared object tracking system, building an efficient model to represent the object is a critical task.

In the recent years, considerable efforts have been devoted to the infrared-based detection or tracking fields in computer vision. Similar to Comaniciu et al. [11], Wang et al. [12] presented a target-local background likelihood ratio based feature model to weight the histogram distribution of the target region, and this feature model was then inserted into the mean shift framework to complete the tracking process. The sparse representation technique was utilized to build the feature model of the target [13]. In this work, they first obtained the compressed feature vector by the sparse representing method, then designed a naïve Bayes nearest neighbor classifier to track the target. Another sparse representation-based infrared target tracking algorithm was proposed in [14], the original Haar-like features were first projected into low-dimensional features. In the next step, the L1 tracker was adopted

to track the object based on compressed features. Different from the above-mentioned methods, both a saliency model and eigenspace model were employed in [15] to serve as the observed models. This fused model was embedded in the particle filter framework to complete the infrared object tracking task. Besides single feature based infrared object tracking scheme, Wang et al. [16] developed a multi-cue based infrared object tracking system, in this system, the intensity cue and edge cue were fused through estimating the discriminant ability for it. In [17], an iterative particle filter was designed to implement the object tracking task, compared with the traditional particle filter, the iterative particle filter can converge much closer to the true target state with high computing efficiency.

When the tracking object is non-rigid, such as a walking person, the torso of the person is relatively stable, but the arms and legs are moving in cycles. In this case, treating the whole person as one target may lead to tracking diffusion. To enhance the performance of tracking a non-rigid object, a parts-based object representation model is employed. Adam et al. [18] proposed a fragment-based object model by splitting the object window into multiple fragments, and each fragment was featured by a corresponding histogram. A voting scheme was then performed on each patch to determine the position and scales of the object. To reduce the computing complexity, the well-known integral histogram was utilized to extract histograms for object patches in this method. It was noticed that articulated objects, such as a human, could be approximated by several overlapped blocks, and Nejhun et al. [19] presented an adaptive parts-based model to depict the articulated object. In this model, both the block configuration and their corresponding weights could be tuned adaptively. In the tracking process, each block associated with a histogram was extracted through the integral histogram data structure, and the object was located in a whole-image scanning manner.

In this paper, we extend the patch-based or fragment-based tracking strategy to the infrared object tracking system, and introduce an adaptively weighted patch-based infrared object tracking algorithm under the particle filtering framework. The proposed algorithm first divides the object windows into a series of non-overlapping sub-regions. After this, a particle filtering based object tracking system is realized. Meanwhile, the discriminative ability of every patch is evaluated based on the linear discriminative analysis (LDA) and particle sampling scheme.

The rest of the paper is organized as follows: in Section 2 we make a brief introduction of the particle filter framework; then, in Section 3, after the feature representation model is introduced, we present our new adaptive patch-based infrared object tracking scheme; Section 4 gives the experimental results; and, finally, conclusions are made in Section 5.

## 2. Particle Filter Review

In contrast to the Kalman filter, which restricts the filtering system with linear modeling and Gaussian assumptions, particle filters are often used to solve the non-linear and non-Gaussian problems. It has been proved to be an effective algorithm for object tracking [20–22].

Particle filter is a model estimation technique based on the Monte Carlo methodologies within a Bayesian inference framework. Let  $\mathbf{X}_t = \{x_0, x_1, \dots, x_t\}$  and  $\mathbf{Y}_t = \{y_0, y_1, \dots, y_t\}$  denote the state vector and the measurement up to time  $t$ , respectively. Based on the Bayesian estimation theory, the optimal estimate of  $x_t$  can be deduced by the posterior mean  $E[x_t | \mathbf{Y}_t]$ . Assuming that, at time  $t - 1$ , the posterior probability density function (pdf)  $p(x_{t-1} | \mathbf{Y}_{t-1})$  is known, then the posterior pdf at time  $t$  can be achieved by the following two stages:

Prediction:

$$p(\mathbf{x}_t | \mathbf{Y}_{t-1}) = \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{Y}_{t-1}) d\mathbf{x}_{t-1} \quad (1)$$

Update:

$$p(\mathbf{x}_t | \mathbf{Y}_t) = \frac{p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{Y}_{t-1})}{p(\mathbf{y}_t | \mathbf{Y}_{t-1})} \quad (2)$$

During prediction stage, the prior  $p(x_{t-1} | Y_{t-1})$  is propagated to  $p(x_t | Y_{t-1})$  through a system dynamical model  $p(x_t | x_{t-1})$ .  $p(x_t | Y_{t-1})$  is then be modified by the coming observation  $Y_t$  with the likelihood function  $L(Y_t | x_t)$  in the update stage. However, the integral in Equation (1) is difficult to obtain because the pdf of  $x$  is usually unknown. To avoid this problem, a particle filter employs a weighted sample set  $\mathbf{S} = \{(s^{(n)}, w^{(n)}) | n = 1, \dots, N\}$  to approximate the posterior pdf. In  $\mathbf{S}$ , each sample provides one candidate state of the object, with a corresponding weight  $w$ , where  $\sum_{n=1}^N w^{(n)} = 1$  and  $N$  is the number of particles, and the value of  $w$  is proportional to the observation likelihood function. The expected state of the object can be estimated by:

$$E(\mathbf{S}) = \sum_{n=1}^N w^{(n)} s^{(n)} \quad (3)$$

In order to model the motion of object, a first order auto-regression (AR) dynamics model is adopted to depict the object's translational motion. In addition to this, we also employ a random walk model to express the scale change. Let  $sc_t$  be the scale of the object window at time  $t$ . The random walk of  $sc$  is modeled as:  $sc_t = sc_{t-1} + Crand()$ , where  $C$  is a constant to control of the radius of the walk.

Suppose that a rectangular box is used to characterize the object, and then the state vector can be defined as:

$$\mathbf{x} = \{x, y, x', y', s_x, s_y\} \quad (4)$$

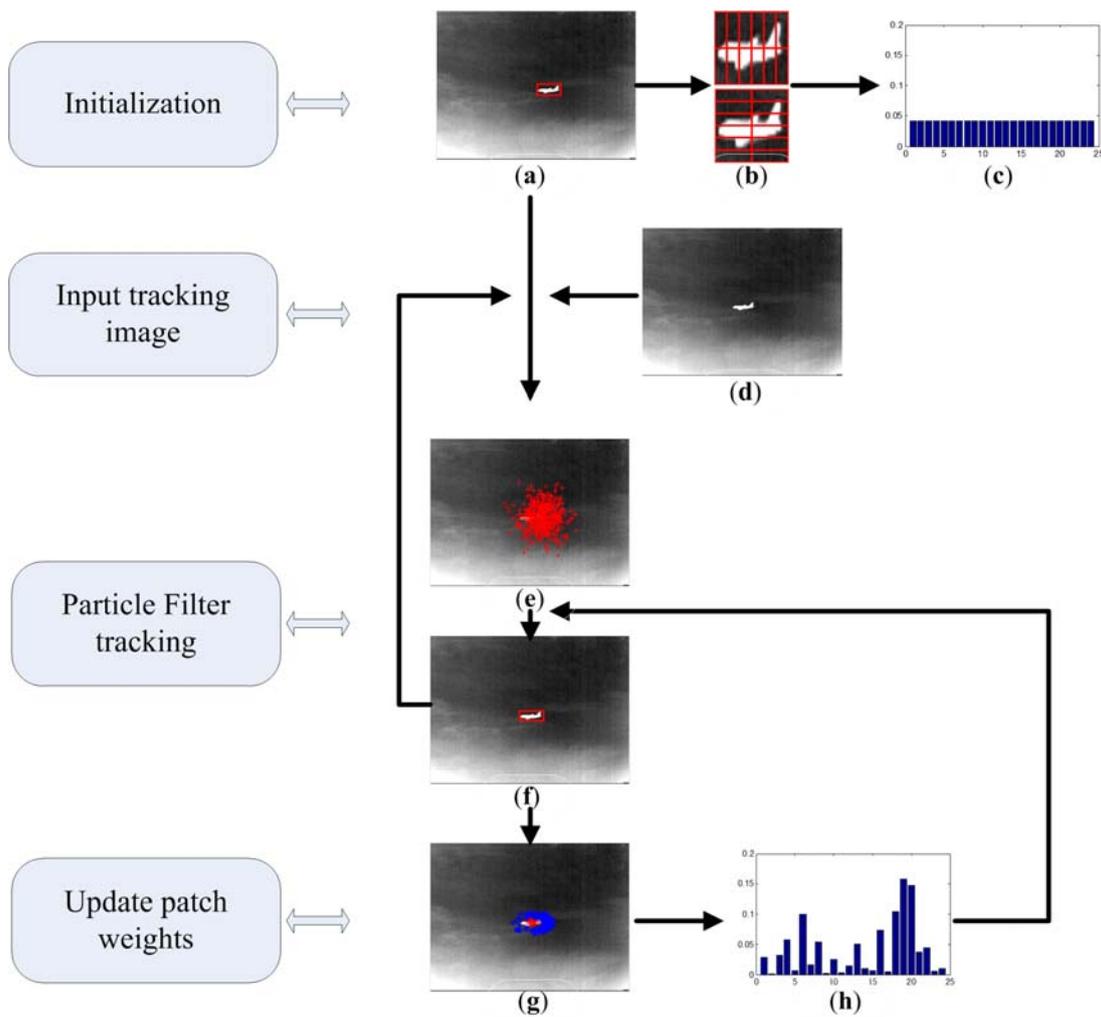
where  $x$  and  $y$  denote the center coordinate of the rectangle,  $x'$  and  $y'$  give the velocities of the centric along  $x$  and  $y$  direction,  $s_x$  and  $s_y$  are the width and height of the rectangle. Based on the above analysis, the object state translation equation has the form of:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{v}_t \quad (5)$$

where  $\mathbf{v}_t$  is stochastic noise which obey multivariate normal distribution. Matrices  $\mathbf{A}$  and  $\mathbf{B}$  are identity matrices. It is straightforward to extend this model to a higher order if a more complex dynamical model is needed.

### 3. The Proposed Algorithm

A novel patch-based infrared object tracking scheme is proposed in this section to realize the object tracking task. As can be seen in Figure 1, in the initialization module, the target position is initialized manually on the first frame with a red bounding box (Figure 1a). In the next step, the local region of target window is normalized to a predefined scale, and then be divided into a series of sub-regions which are shown in Figure 1b. For each patch, a corresponding weight is assigned on it to depict its tracking ability. As described in Figure 1c, all the weights are initialized as having equal value. When a new frame is coming, the particle filter tracking module is implemented to locate the object. This is illustrated in Figure 1e,f. Firstly, particles are drawn according to the state transition equation. Then the current position of object is estimated by the states of particles. In the last module, the tracking performance of each sub regions is evaluated and updated. The whole tracking scheme comprises two modules: object tracking and updating weights of patches. In the object tracking module, the color particle filter framework is adopted to perform this task. When the current object state has been estimated by the particle filter, we execute the weight updating step. In this step, particles are selected to form the positive sample set and negative sample set, and these two sample sets are used to update weight of each sub-region.



**Figure 1.** Procedure of the proposed algorithm. (a) Object position is initialized manually with a red bounding box in the first frame. (b) Object region is divided into no-overlapping patches, and normalized to a fixed scale. (c) Each patch is assigned with a weight and the initial value of each weight is the same. (d) Capture a new incoming frame. (e) Drawing particle samples based on the state transition equation. (f) The particle filter is conducted to estimate the current position of object. (g) Positive and negative samples are chosen form the particles. (h) Patch weights are updated by the linear discriminative analysis (LDA) technique.

### 3.1. Sub-Region Tracking Ability Evaluation

The key unit of our algorithm is the adaptive fusion of all the patches. Suppose that the current state of the object has been known, then we can measure the tracking performance of each patch based on the tracking results. In general, sub-regions which are enabled to clearly separate the object from the background have more reliable discriminative ability, thus, it should be assigned with larger weight values, and vice versa. Here, the linear discriminative analysis (LDA) technique is introduced to describe the discriminative ability of sub-regions. In the LDA, two categories of sample sets which indicate it is positive or negative should be established. In our case, we treat each particle as one sample, and the overlapping rate between particle and object is applied to decide which category the particle should belong to. As shown in Figure 2, given two rectangle regions  $R_1$  and  $R_2$ , the overlapping rate between them can be defined as the ratio of  $R_1 \cap R_2$  and  $R_1 \cup R_2$ . According to the object regions, since the positive samples describe characteristics of object, it should have large overlapping value, so as to approximate the object closely. As for the negative samples, they give characteristics of the

background around the target; thus, it should have little overlapping value. However, this does not mean the negative sample should keep far away from the object, because the object only moves in a certain local region during a period of time. Based on the above analysis, the rule of selecting positive and negative samples is formulated as Equation (6), in which  $R_i$  is the region of particle  $i$ .  $R_o$  is the region of object.  $th_p$  is the threshold to decide positive sample, and the negative samples are identified by two thresholds of  $th_{n1}$  and  $th_{n2}$ .

$$\begin{aligned} \frac{R_i \cap R_o}{R_i \cup R_o} &> th_p, \text{ positive sample} \\ \frac{R_i \cap R_o}{R_i \cup R_o} < th_{n1} \text{ and } \frac{R_i \cap R_o}{R_i \cup R_o} > th_{n2}, &\text{ negative sample} \end{aligned} \tag{6}$$

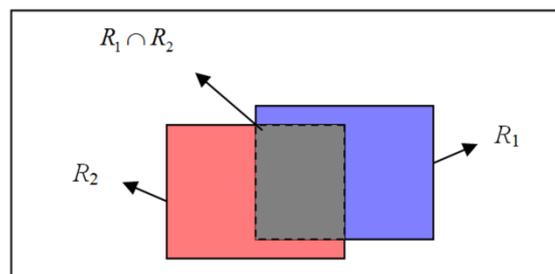


Figure 2. The overlapping rate between two rectangle regions.

To measure the performance of each sub region, the LDA technique is introduced. Suppose we have built two classes of samples,  $\chi_1$  and  $\chi_2$ , based on the above presented sample selecting criterion. The means of these two sets is defined as:

$$\begin{cases} \tilde{\mathbf{m}}_1 = \frac{1}{N_1} \sum_{\mathbf{x} \in \chi_1} \mathbf{x} \\ \tilde{\mathbf{m}}_2 = \frac{1}{N_2} \sum_{\mathbf{x} \in \chi_2} \mathbf{x} \end{cases} \tag{7}$$

The within-class scatter matrixes of them are given by:

$$\begin{cases} \tilde{\mathbf{S}}_1 = \sum_{\mathbf{x} \in \chi_1} (\mathbf{x} - \tilde{\mathbf{m}}_1)(\mathbf{x} - \tilde{\mathbf{m}}_1)^T \\ \tilde{\mathbf{S}}_2 = \sum_{\mathbf{x} \in \chi_2} (\mathbf{x} - \tilde{\mathbf{m}}_2)(\mathbf{x} - \tilde{\mathbf{m}}_2)^T \end{cases} \tag{8}$$

The LDA evaluates the separability between these two classes through:

$$J = \frac{(\tilde{\mathbf{m}}_1 - \tilde{\mathbf{m}}_2)^2}{\tilde{\mathbf{S}}_1^2 + \tilde{\mathbf{S}}_2^2} \tag{9}$$

If the two groups of sample separate from each other more distinctively, then  $J$  has a larger value. Therefore, the value of  $J$  can be employed to indicate the discrimination between object and background. As above mentioned, the tracking window is split into several patches. Let the number of sub-regions be  $M$ , and we choose  $N_p$  positive samples and  $N_n$  negative samples from all of the particles to build the two sample sets. For each patch  $P_i, i = 1, 2, 3, \dots, M$ , the corresponding  $J_i$  is given by:

$$J_i = \frac{(\tilde{\mathbf{m}}_i^P - \tilde{\mathbf{m}}_i^N)^2}{(\tilde{\mathbf{S}}_i^P)^2 + (\tilde{\mathbf{S}}_i^N)^2} \tag{10}$$

where:

$$\begin{cases} \tilde{\mathbf{m}}_i^P = \frac{1}{N_P} \sum_{j=1}^{N_P} \mathbf{x}_i^j \\ \tilde{\mathbf{m}}_i^N = \frac{1}{N_N} \sum_{j=1}^{N_N} \mathbf{x}_i^j \end{cases} \tag{11}$$

and

$$\begin{cases} \tilde{\mathbf{S}}_i^P = \sum_{j=1}^{N_P} (\mathbf{x}_i^j - \tilde{\mathbf{m}}_i^P)(\mathbf{x}_i^j - \tilde{\mathbf{m}}_i^P)^T \\ \tilde{\mathbf{S}}_i^N = \sum_{j=1}^{N_N} (\mathbf{x}_i^j - \tilde{\mathbf{m}}_i^N)(\mathbf{x}_i^j - \tilde{\mathbf{m}}_i^N)^T \end{cases} \tag{12}$$

Here,  $\mathbf{x}_i^j$  is the matrix corresponding to the  $i^{\text{th}}$  patch in the  $j^{\text{th}}$  candidate region. We define the relative discriminative factors among all the patches as:

$$R_i = \frac{J_i}{\sum_{i=1}^M J_i} \tag{13}$$

$R_i$  is employed to depict the discriminative property of the  $i^{\text{th}}$  patch.

### 3.2. The Proposed Tracking Scheme

For the lack of visible information in infrared images, such as color, in this work the intensity feature is used to represent the tracking target. To measure the difference between candidate particles and the reference model, for each particle, the corresponding image region is normalized to have the same size as the reference region. Hence, the likelihood for patch  $i$  can be defined as:

$$likelihood_i = \frac{1}{\sqrt{2\pi}\sigma_i} \exp - \frac{\|C_i - T_i\|^2}{2\sigma_i^2} \tag{14}$$

where  $\sigma_i$  is the standard deviation which specifies the Gaussian noise in the measurements,  $C_i$  and  $T_i$  are the intensity matrix for  $i^{\text{th}}$  patch in the candidate image and reference image respectively, and the size of them have been normalized as the same. To enhance the adaptability of the tracking system, we incorporate above analyzed  $R_i$  into our proposed object tracking architecture. The corresponding particle weight is calculated as:

$$w \propto likelihood = \frac{1}{M} \sum_{i=1}^M (likelihood_i)^{R_i} \tag{15}$$

Obviously, particles with larger  $w$  approach the target state more closely. Based on the particle filter tracking frame, the proposed tracking scheme can be summarized in Algorithm 1.

**Algorithm 1. The Proposed Tracking Algorithm**

Set the initial values: the initialization state of target  $x_0$ , the number of patches  $M$ , the weight corresponding to the  $i^{\text{th}}$  patch  $R_i = 1/M$

**for**  $t = 1, 2, \dots$

1 **for**  $i = 1, 2, \dots, N$ , resample the particles:  $\tilde{x}_{t-1}^i = \text{resample}(x_{t-1})$ ; **end for**

2 Predict the state of particle at time  $t$ :

**for**  $i = 1, 2, \dots, N$

$$\mathbf{x}_t^i = \mathbf{A}\mathbf{x}_{t-1}^i + \mathbf{B}\mathbf{v}_{t-1}$$

$$w_t^i = \frac{1}{M} \sum_{j=1}^M \left( \frac{1}{\sqrt{2\pi}\sigma_j} \exp -\frac{\|C_j - T_j\|^2}{2\sigma_j^2} \right)^{R_j}$$

**end for**

**for**  $i = 1, 2, \dots, N$ , normalize the weight:  $\hat{w}_t^i = \frac{w_t^i}{\sum_{j=1}^N w_t^j}$ ; **end for**

3 State estimation:  $\hat{\mathbf{X}}_t = \sum_{i=1}^N \hat{w}_t^i x_t^i$

4 Update  $R_i$

**for**  $i = 1, 2, \dots, M$ , calculate the discriminant parameter for each patch:

$$J_i = \frac{(\tilde{\mathbf{m}}_i^P - \tilde{\mathbf{m}}_i^N)^2}{(\tilde{\mathbf{S}}_i^P)^2 + (\tilde{\mathbf{S}}_i^N)^2};$$

**end for**

**for**  $i = 1, 2, \dots, N$ , update  $R_i$ :  $R_i = \frac{J_i}{\sum_{i=1}^M J_i}$ ; **end for**

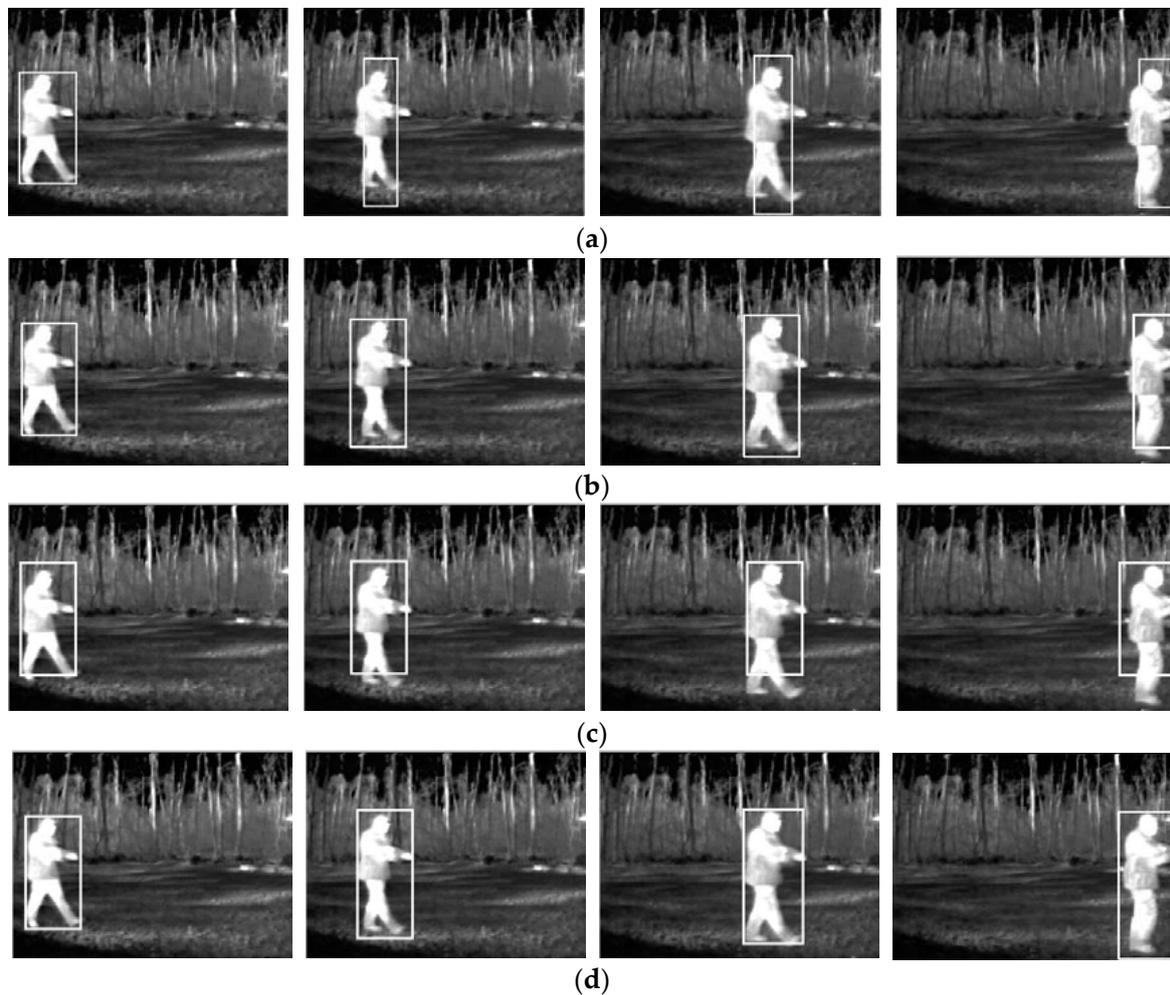
**end for**

**4. Experiment Results and Analysis**

In this section, we evaluate the performance of the presented algorithm through experiments on challenging infrared sequences. In the course of the experiment, four image sequences are used which are extracted from both the OTCBVS databases [23] and datasets constructed by ourselves, a Raytheon L-3 Thermal-Eye 2000AS infrared sensors (Raytheon Company, Waltham, MA, United States) is utilized to grab 8-bit grayscale images with a resolution of  $320 \times 240$  pixels. All of the tests are carried out on a desktop PC equipped with an Intel Core i7-4790 CPU and 8G RAM. At the first frame, the object window was drawn manually for each sequence. Both quality and quantity analysis was conducted during the experiments. We also compared the presented algorithm with three other state-of-art methods, Incremental learning for robust Visual Tracking (IVT) [24], L1 [25], and Compressive Tracking (CT) [26] tracker, to depict the tracking performance more straightforwardly. Taking both the effectiveness and the computation efficiency into account, the particle number is fixed as 800 for all the tests. In our proposed tracking scheme, the value of  $th_p$ ,  $th_{n1}$  and  $th_{n2}$  are experimentally set as 0.75, 0.4, and 0.1, respectively, and the target region is divided into 24 patches as illustrated in Figure 1b. We should note that, as described in [19], this choice of patches is arbitrary. Under this parameter setting our algorithm can run at speed of 2.5~10 fps. All of the four methods are running on MATLAB 2012a (MathWorks, Natick, MA, United States) platform.

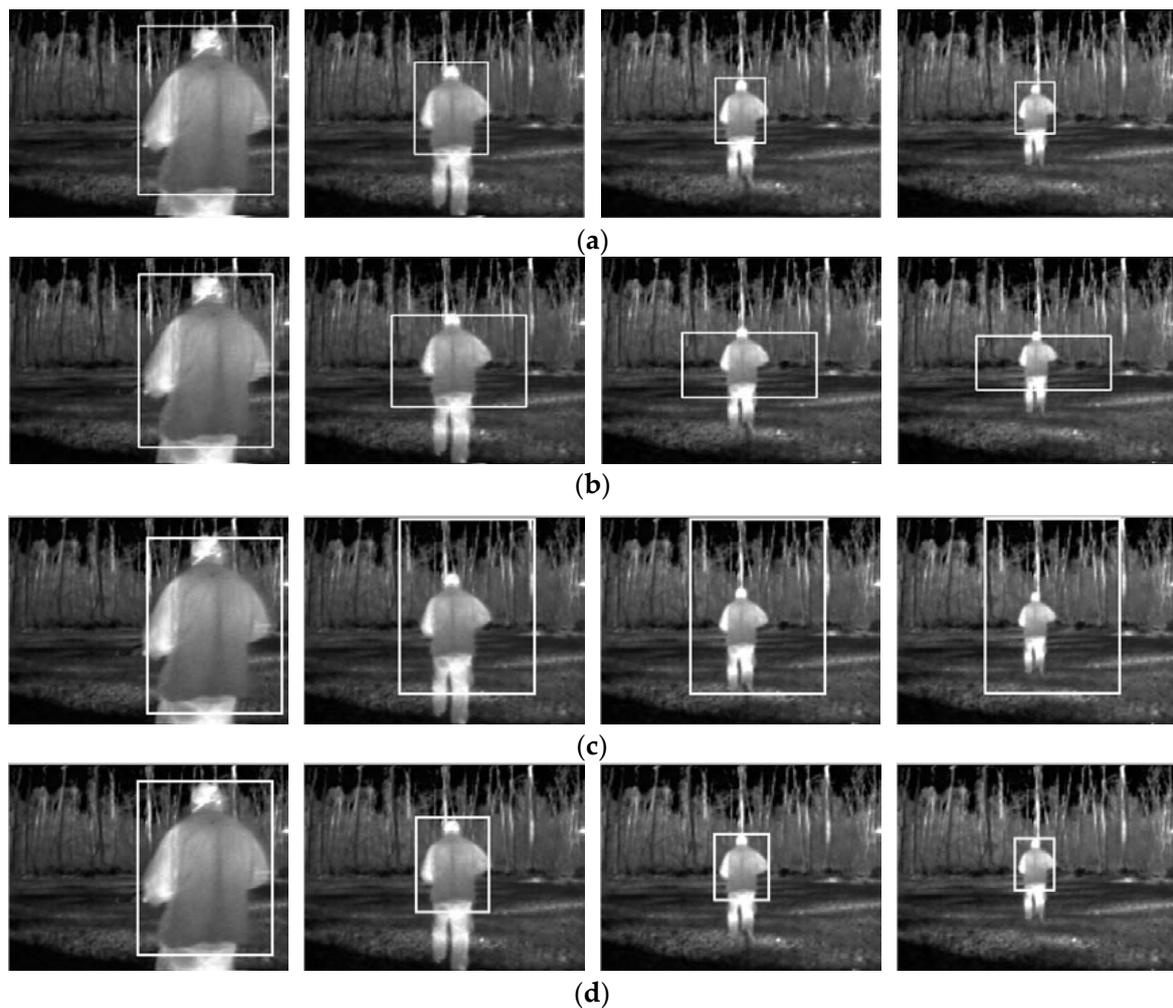
The first sequence is extracted from the familiar OTCBVS databases with a resolution of  $320 \times 240$ . In this sequence a man walks from the left side to the right side with deformation. Four sample images are given in Figure 3, and their indices are 218, 238, 284, and 331. From top to bottom, each column in

this figure is corresponding to the tracking result using IVT, L1, CT, and our algorithm, respectively. The difficulty of this sequence lies in that the shape of the target is changing with the non-rigid motion of limbs. As shown in Figure 3, we can see that both the L1 and our proposed method enable coping with this difficulty and track the target precisely. However, the IVT and CT tracker yields considerable tracking error and that they lost some scale information during the tracking process.



**Figure 3.** Tracking results for sequence 1 by: (a) L1 tracker; (b) Incremental learning for robust Visual Tracking (IVT) tracker; (c) Compressive Tracking (CT) tracker; and (d) the proposed algorithm.

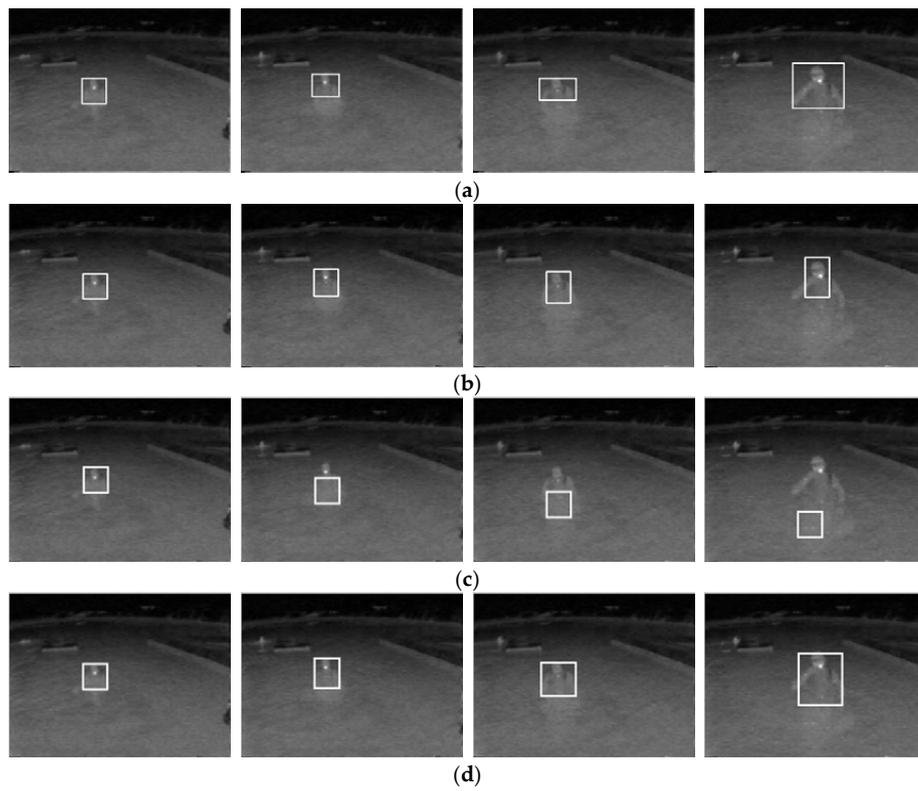
The second sequence is drawn from the Terravic Motion IR Database [27] which is included in the OTCBVS databases with the same resolutions as sequence 1. The experimental goal of this test is to investigate the ability of handling large scale changing of the three trackers. For that, in this sequence a man walks from the near to the far with sharp variation of scale. In Figure 4, some samples are provided; the frame indices of them from left to right are 129, 179, 222, and 261. In this test, the IVT tracker (see Figure 4b) and our presented method (see Figure 4d) can locate the target precisely in both position and scale. In contrast, the L1 tracker (see Figure 4a) and the CT tracker (see Figure 4c) failed to locate the target tightly.



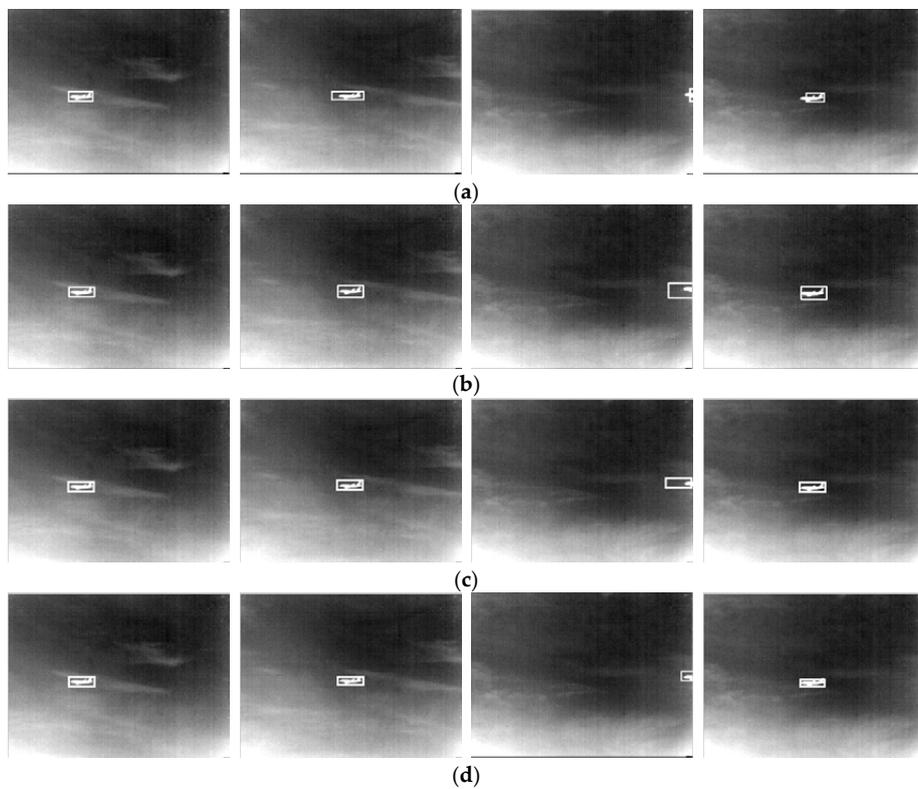
**Figure 4.** Tracking results for sequence 2 by: (a) L1 tracker; (b) IVT tracker; (c) CT tracker; and (d) the proposed algorithm.

An underwater sequence with low signal to noise ratio (SNR) is chosen as the third test sequence from the Terravic Motion IR Database [27]. In this sequence, a man moves in water and the background is very cluttered, and this was monitored by an infrared sensor on the water. Four selected frames with indices of 1643, 1698, 1756, and 1853 are supplied in Figure 5 to illustrate the tracking results. It can be observed from Figure 5 that, besides our proposed method (see Figure 5d), large error aroused in frame 1756 for L1 (see Figure 5a) and frame 1853 for IVT (see Figure 5b). As for the CT tracker, it lost the target from frame 1698. The proposed method achieved good tracking performance due to the ability of adaptive estimating the discrimination of various patches.

The last sequence is built by ourselves. The tracking challenge for this test lies in that the airplane had been partially flown out the camera field. In Figure 6, four representative frames with indices of frame 1001, 1010, 1084, and 1121 are provided to display the tracking results. Each column of Figure 6 is corresponding to one type of tracker; four columns are arranged in the same order as the past three tests. It can be observed that the L1 tracker enables coarse locating of the target, however the bounding box drifted from the target center with large errors. As for the IVT and CT tracker, despite that these could precisely locate the target in frame 1001, 1010, and 1121, but heavy bias occurred in frame 1084 due to the target partially flying out of the image. Comparing with above three trackers, our presented method was able to locate the target tightly throughout the whole sequence.



**Figure 5.** Tracking results for sequence 3 by: (a) L1 tracker; (b) IVT tracker r; (c) CT tracker; and (d) the proposed algorithm.



**Figure 6.** Tracking results for sequence 4 by: (a) L1 tracker; (b) IVT tracker r; (c) CT tracker; and (d) the proposed algorithm.

We also conduct quantitative comparison on performance of the tracking methods. In Table 1, the average error between center of tracking results and the ground truth are given. As can be seen, our designed algorithm achieved more accurate tracking results than the L1, IVT, and CT tracker for all of the four test sequences.

**Table 1.** Average center error to ground truth (in pixels).

Image Sequence	L1 Tracker	IVT Tracker	CT Tracker	Proposed
Sequence 1	4.65	7.52	9.52	3.42
Sequence 2	6.63	5.34	23.24	4.15
Sequence 3	7.45	9.26	25.36	3.53
Sequence 4	5.84	4.23	4.45	2.68

## 5. Conclusions

Tracking objects in infrared videos is a task full of potential in the computer vision field. In this paper, an adaptive weighted patch-based infrared object tracking algorithm is presented. To enhance the describing ability of the object, the candidate local region is divided into non-overlapping sub-regions, and a set of belief weight is set on these patches. Then the particle filter scheme and the linear discriminative analysis technique are utilized to evaluate each patch's belief, as well as locating the object. Experimental results on various infrared sequences demonstrated that the presented algorithm was able to obtain promising performance.

**Acknowledgments:** This work was partially supported by the Natural Science Foundation of China (61203272, 61572224), Natural Science Foundation of Anhui Province (1508085MF116, 1308085MF105), the seventh Batch of '115' Industrial Innovation Team of Anhui Province, Key Project of University Natural Science Research of Anhui Province (KJ2013A237) and the International Science & Technology Cooperation Plan of Anhui Province (10080703003). The Key Program in the Youth Elite Support Plan in Universities of Anhui Province (gxyqZD2016113).

**Author Contributions:** Jingai Zhang designed the overall system and developed the pedestrian detection algorithms. Jiangtao Wang wrote the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Yilmaz, A.; Javed, O.; Shah, M. Object tracking: A survey. *ACM Comput. Surv.* **2006**, *38*, 13. [[CrossRef](#)]
2. Mei, X.; Ling, H. Robust visual tracking and vehicle classification via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2259–2272. [[PubMed](#)]
3. Chan, A.L.; Schnelle, S.R. Fusing concurrent visible and infrared videos for improved tracking performance. *Opt. Eng.* **2013**, *52*, 177–182. [[CrossRef](#)]
4. Liu, H.P.; Sun, F.C. Fusion Tracking in Color and Infrared Images Using Joint Sparse Representation. *Sci. China Inf. Sci.* **2012**, *55*, 590–599. [[CrossRef](#)]
5. Li, X.; Guo, R.; Chen, C. Robust Pedestrian Tracking and Recognition from FLIR Video: A Unified Approach via Sparse Coding. *Sensors* **2014**, *14*, 11245–11259. [[CrossRef](#)] [[PubMed](#)]
6. Richard, S. *Computer Vision: Algorithms and Applications*; Springer-Verlag: London, UK, 2011; pp. 87–104.
7. Boguslaw, C. *Object Detection and Recognition in Digital Images: Theory and Practice*; John Wiley & Sons: West Sussex, UK, 2013; pp. 389–394.
8. Zhang, S.; Yao, H.; Sun, X.; Liu, S. Robust visual tracking using an effective appearance model based on sparse coding. *ACM Trans. Intell. Syst. Technol.* **2012**, *3*, 43. [[CrossRef](#)]
9. Han, J.; Pauwels, E.J.; de Zeeuw, P.M.; de With, P.H.N. Employing a RGB-D Sensor for Real-Time Tracking of Humans across Multiple Re-Entries in a Smart Environment. *IEEE Trans. Consum. Electron.* **2012**, *58*, 255–263.
10. Zhang, S.; Yao, H.; Sun, X.; Lu, X. Sparse coding based visual tracking: Review and experimental comparison. *Pattern Recogn.* **2013**, *46*, 1772–1788. [[CrossRef](#)]
11. Comaniciu, D.; Ramesh, V.; Meer, P. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 564–577. [[CrossRef](#)]

12. Wang, Z.; Hou, Q.; Hao, L. Improved infrared target-tracking algorithm based on mean shift. *Appl. Opt.* **2012**, *51*, 5051–5059. [[CrossRef](#)] [[PubMed](#)]
13. Gao, S.; Kim, I.; Jhang, S.T. Infrared Target Tracking Using Naïve-Bayes-Nearest-Neighbor. *IEICE Trans. Inf. Syst.* **2015**, *98*, 471–474. [[CrossRef](#)]
14. Li, Y.; Li, P.; Shen, Q. Real-time infrared target tracking based on  $\ell_1$  minimization and compressive features. *Appl. Opt.* **2014**, *53*, 6518–6526. [[CrossRef](#)] [[PubMed](#)]
15. Wang, F.; Zhen, Y.; Zhong, B.; Ji, R. Robust infrared target tracking based on particle filter with embedded saliency detection. *Inf. Sci.* **2015**, *301*, 215–226. [[CrossRef](#)]
16. Wang, J.; Chen, D.; Chen, H.; Yang, J. On pedestrian detection and tracking in infrared videos. *Pattern Recogn. Lett.* **2012**, *33*, 775–785. [[CrossRef](#)]
17. Fan, Z.; Ji, H.; Zhang, Y. Iterative particle filter for visual tracking. *Signal Process. Image Commun.* **2015**, *36*, 140–153. [[CrossRef](#)]
18. Adam, A.; Rivlin, E.; Shimshoni, I. Robust Fragments-based Tracking using the Integral Histogram. In Proceedings of the 2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New York, NY, USA, 17–22 June 2006; pp. 798–805.
19. Nejhum, S.; Ho, J.; Yang, M.H. Online visual tracking with histograms and articulating blocks. *Comput. Vis. Image. Underst.* **2010**, *114*, 901–914. [[CrossRef](#)]
20. Morshidi, M.; Tjahjadi, T. Gravity optimised particle filter for hand tracking. *Pattern Recogn.* **2014**, *47*, 194–207. [[CrossRef](#)]
21. Dou, J.; Li, J. Robust visual tracking based on interactive multiple model particle filter by integrating multiple cues. *Neurocomputing* **2014**, *135*, 118–129. [[CrossRef](#)]
22. Liu, J.; Meng, J.; Wang, J.; Han, M. The Application of Particle Filter Algorithm in Multi-target Tracking. *Opt. Eng.* **2016**, *129*, 419–424.
23. OTCBVS Benchmark Dataset Collection. Available online: <http://vcipl-okstate.org/pbvs/bench/> (accessed on 7 September 2016).
24. Ross, D.A.; Lim, J.; Lin, R.S.; Yang, M.H. Incremental learning for robust visual tracking. *Int. J. Comput. Vis.* **2008**, *77*, 125–141. [[CrossRef](#)]
25. Mei, X.; Ling, H. Robust visual tracking using  $\ell_1$  minimization. In Proceedings of the IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 27 September–4 October 2009; pp. 1436–1443.
26. Zhang, K.; Zhang, L.; Yang, M.H. Real-time compressive tracking. In Proceedings of the 2012 European Conference on Computer Vision (ECCV), Florence, Italy, 8–11 October 2012; pp. 864–877.
27. Mieziako, R. *Terravic Research Infrared Database*. IEEE OTCBVS WS Series Bench. Available online: <http://vcipl-okstate.org/pbvs/bench/> (accessed on 10 October 2016).



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).