Target Tracking in Infrared Image Sequences Using Diverse AdaBoostSVM

Zhenyu Wang[1,2], Yi Wu[1,2], Jinqiao Wang[1,2], and Hanqing Lu[1]

[1] *National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, 100080, China*
[2] *Graduate School of the Chinese Academy of Sciences, Beijing, 100039, China*
{zywang, ywu, jqwang, luhq}@nlpr.ia.ac.cn

Abstract

This paper presents a novel algorithm named Diverse AdaBoostSVM Tracking(DABSVT) for target tracking in infrared imagery. The tracker trains a Support Vector Machine(SVM) classifier per frame. All of the classifiers are combined into an ensemble classifier using AdaBoost. By proper parameter adjusting strategies, a set of effective SVM classifiers with moderate accuracy are obtained, and the dilemma problem between accuracy and diversity of AdaBoost is dealt with too. To cope with the changes in features of both foreground and background, the component classifier can be discarded or added at any time. The experiments performed on several sequences show the robustness of the proposed method.

1. Introduction

Detection and tracking of targets in infrared image sequences are challenging problems in pattern recognition and computer vision. More information which is not available in visual images can be provided from infrared images in night or bad weather condition, because the infrared images are obtained by sensing the radiation in the infrared spectrum. However, in contrast to visual imagery, there are lots of shortcomings in infrared imagery such as extremely low signal to noise ratio, high ego motion of the camera, variations of the target features and target partial occlusion. All of them make target tracking in infrared imagery even harder.

To overcome the aforementioned shortcomings, there is a very limited amount of work which has been reported in the literature. In a recent study, Dawoud and Alam[1] proposed the decision fusion algorithm using the weighted composite reference function. Bal and M. S. Alam[2] utilized intensity variation function and template modeling. Yilmaz et al.[3][4] performed a multi-resolution global motion compensation using the Gabor responses of the consecutive frames. Jagdish Chandra Patra et al.[5] combined Principal Component Analysis and Artificial Neural Networks to reduce training time.

In this paper, we propose a novel algorithm named Diverse AdaBoostSVM Tracking(DABSVT) for automatic target tracking in infrared imagery involving targets affected by high global motion and changes in target features. We train component classifiers on-line to separate the infrared target from the background. All of the component classifiers are combined into an ensemble classifier by AdaBoost[10]. We use the ensemble classifier to distinguish the target from the background in the next frame and form a confidence map. Then mean shift[6] is used to find the peak of the map, which is thought as the target position.

Our approach is similar to that of [7] that use ensemble tracking methods in visual imagery. We extend their work and apply it into target tracking in infrared image sequences. In our algorithm, Support Vector Machine(SVM)[8] is adopted as a component classifier in every frame. The parameters of all SVM classifiers are different and unknown beforehand, so it is a problem how to adjust the parameters adaptively. We utilize proper parameter adjusting strategies[9] to get a set of effective SVM classifiers with moderate accuracy and combine them into an ensemble classifier. This method not only deals with the well known problem of accuracy/diversity dilemma of the original AdaBoost, but also gives a chance to settle the problem resulting from unbalanced data sets in classification.

2. Diverse AdaBoostSVM Tracking

The proposed DABSVT algorithm treats target tracking as a binary classification problem and utilizes SVM as classifier to test the pixels in the next frame. In order to improve the robustness of the approach, we train classifier not only using the information of the current frame, but also considering information of

the previous frames. SVM is trained as a component classifier by the positive and negative samples of every frame. All of the SVM component classifiers are integrated together into an ensemble classifier by AdaBoost[10]. But there are three questions. First, the parameters of all SVM component classifiers are unknown beforehand, then how should they be adjusted adaptively? Second, with the foreground and background changing, some information of previous frames may become unhelpful to classification in the newest frame, then how could the algorithm evaluate and discard these samples? Third, there is a dilemma problem between accuracy and diversity in AdaBoost[13]. Not only the SVM component classifiers should have moderate accuracy, but also their errors should not be highly correlated. We utilize the Diverse AdaBoostSVM algorithm[9] to cope with these problems.

2.1. Classifier and feature selection

SVM[8] is used as a component classifier for AdaBoost in every frame. In a binary classification, the decision function of SVM is:

$$f(X) = \langle w, \phi(X) \rangle + b \tag{1}$$

The optimal values of w and b can be obtained by solving the following optimization problem,

$$\min imize: \quad g(w, \xi) = \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} \xi_i \tag{2}$$

$$subject\ to: \quad y_i(\langle w, \phi(X) \rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \tag{3}$$

We choose the radial basis function(RBF) as the kernel function. The RBF kernel function is:

$$k \langle X_i, X_j \rangle = \exp(-\frac{\| X_i - X_j \|^2}{2\sigma^2}) \tag{4}$$

The performance of RBFSVM is mainly affected by the parameters, for example, the regularization parameter, C, and the Gaussian width of the RBF kernel. But it is not known beforehand which parameters are the best. We utilize proper parameter adjusting strategies[9] to cope with the problem.

Each pixel $\{x_i\}$ is represented as a feature vector. The feature vector we use includes the local orientation histogram[11], the Gabor responses[12] and the gray scale intensity value. The pixels which belong to the target are considered as positive samples and the pixels which belong to the background are considered as negative samples. So the label $\{y_i\}$ of samples $\{x_i\}$ is +1 or -1.

2.2. Classifier update

There is a dilemma problem between accuracy and diversity in AdaBoost method[13]. If two component classifiers are very accurate, they may become highly

correlated. We use proper parameter adjusting strategies[9] to balance the accuracy and diversity. The diversity of the t-th component classifier is defined:

$$d_t(X_i) = \begin{cases} 0, & if \quad h_t(X_i) = f(X_i) \\ 1, & if \quad h_t(X_i) \neq f(X_i) \end{cases} \tag{5}$$

$h_t(x_i)$ is the prediction label of the t-th component classifier on sample x_i, and $f(x_i)$ is the combined prediction label of all the other component classifiers.

Table 1. Input and output of DABSVT algorithm

Input: n video frames $I_1, …, I_n$;
     Rectangle r_1 of target in the first frame;
     Initial σ, σ_{ini}; Minimal σ, σ_{min}; Step of σ, σ_{step};
     Threshold of diversity, D_{min}
Output: Rectangle r_j of target in the frame I_j, j=2, …, n

Table 2. The initialization of DABSVT algorithm

Initialization(for the first frame):
1. Extract examples with labels $\{(x_1,y_1), …, (x_N,y_N)\}$
2. Set the weight of samples: $w_i = 1/N, \quad i = 1,…,N$
3. For t=1, …, T(while σ>σ_{min})
  (1) Make $\{w_i\}_{i=1}^{N}$ a distribution
  (2) Train the RBFSVM classifier h_t
  (3) Calculate error: $err = \sum_{i=1}^{N} w_i, \quad y_i \neq h_t(X_i)$
  (4) Calculate diversity: $D_t = \frac{1}{N} \sum_{i=1}^{N} d_t(X_i)$
  (5) If err>0.5 or D_t<D_{min}, then σ=σ-σ_{step}, go to (1)
  (6) Set component classifier weight: $\alpha_t = \frac{1}{2} \ln \frac{1-err}{err}$
  (7) Update example weights: $w_i = w_i \exp\{-\alpha_t y_i h_t(X_i)\}$
4. The ensemble classifier: $H(X) = sign(\sum_{t=1}^{T} \alpha_t h_t(X))$

When the new RBFSVM classifier is trained, the diversity value is calculated. If it is smaller than the predefined threshold, the new RBFSVM classifier will be discarded. Otherwise, the new RBFSVM classifier will be adopted.

In the update stage, the oldest RBFSVM classifier is removed while the newest one is added. Before adding a new RBFSVM classifier, the weights of the remaining old RBFSVM classifiers should be updated. During weight recalculation or during RBFSVM classifier update, the RBFSVM classifier will be thrown away when err is larger than 0.5 or D_t is smaller than D_{min}, until σ is smaller than σ_{min}.

3. Experiments

We have implemented the proposed algorithm and tested its performance on several infrared image sequences. The feature vector of each pixel we use is

25 dimensions, which consists of one gray scale intensity value, 16-bin local orientation histogram and the response of Gabor filter in 8 directions. In the first frame, we train 5 RBFSVM classifiers and combine them into an ensemble classifier by AdaBoost. In all cases, the ego-motion of camera is allowed.

Table 3. The update of DABSVT algorithm

For every new frame I_j do (j=2, …, n):
1. Extract examples with respect to the rectangle r_{j-1}
2. Test the examples using the ensemble classifier H(x) and create confidence map L_j
3. Run mean shift on L_j with r_{j-1} as the initial guess. Let r_j be the result of mean shift
4. Extract new examples with new labels $\{(x_1,y_1), …, (x_N,y_N)\}$, with respect to the new rectangle r_j
5. Remove the oldest component classifier
6. Sort the other T-1 component classifiers by their errors from minimum to maximum
7. Set the weight of samples: $w_i = 1/N$, $i = 1,…,N$, and set $\sigma=\sigma_{ini}$
8. For t=2, …, T (Update weights, while $\sigma>\sigma_{min}$)
   (a) Make $\{w_i\}_{i=1}^N$ a distribution
   (b) Calculate training error of h_t: err
   (c) Calculate diversity of h_t: D_t
   (d) If err>0.5 or $D_t<D_{min}$, then $\sigma=\sigma-\sigma_{step}$, go to (a)
   (e) Update component classifier weights and example weights
9. Add a new component classifier, while $\sigma>\sigma_{min}$
   (a) Make $\{w_i\}_{i=1}^N$ a distribution, and set $\sigma=\sigma_{ini}$
   (b) Train the RBFSVM classifier h_1
   (c) Calculate training error of h_1: err
   (d) Calculate diversity of h_1: D_1
   (e) If err>0.5 or $D_1<D_{min}$, then $\sigma=\sigma-\sigma_{step}$, go to (a)
   (f) Calculate the new component classifier weight
10. The updated classifier: $H(X) = sign(\sum_{t=1}^T \alpha_t h_t(X))$

In the first experiment, we track a plane flying in sky. Fig. 1 shows the tracking result of several frames from the infrared image sequence. The small rectangle marks the target position and the big rectangle marks the background. As can be seen, the target object is very dim and the signal to noise ratio(SNR) is extremely low. Both foreground and background features change as the target moves from one place to another. When the target is buried under clutter and noise, the algorithm can still track the target, because it can separate the target from the background successfully. The first row shows the actual images with tracking rectangles marked. The second row shows the confidence map corresponding to the big rectangle, which is produced by the ensemble classifier. The third row shows the result after the

confidence map is back projected into the original image space.

(a)

(b)

(c)
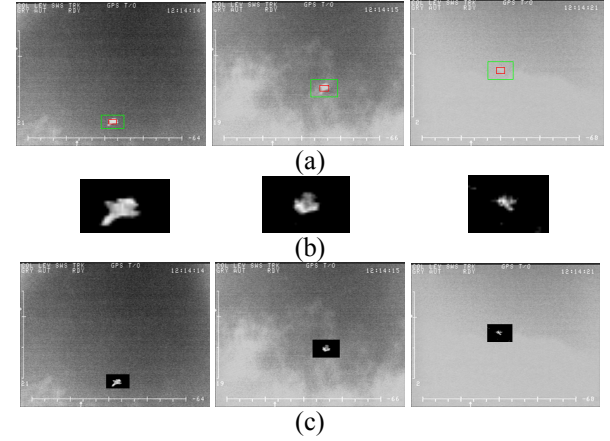
Fig. 1. (a) target tracking result with the proposed DABSVT algorithm in frame 68, 99 and 274; (b) the confidence map corresponding to the big rectangle for each frame; (c) the result with the confidence map back projected into the original image space

The second experiment is carried out on an infrared image sequence of two planes flying in sky, which are very similar. We manage to track one of the planes and avoid the disturbing effect of the other. Fig. 2 shows the tracking result of several frames from the infrared image sequence.

In the third experiment, we try to track a truck running on the ground. Fig. 3 shows the tracking result of several frames. In order to demonstrate the effectiveness of our approach, we also present the tracking results on the same sequence using two other methods. In the first case, we use the DABSVT algorithm proposed in this paper. In the second case, we use AdaBoost with the least-squares based classifiers mentioned in [7]. In the third case, we use a tracker which trains 5 RBFSVM classifiers in the first frame and does not update the ensemble classifier in each new frame. The tracking result of the first case is better than that of the second, because AdaBoost with RBFSVM performs better than AdaBoost with the least-squares based classifier in classification. There are feature variations of both target and background, so it is easy to understand why the result of the first case is better than that of the third. The comparison of the results demonstrates the robustness of our proposed approach for tracking infrared targets.

4. Conclusions

In this paper, we propose a robust algorithm named DABSVT for tracking the target in infrared image sequence. Target tracking is considered as a binary

classification problem, where we adopt RBFSVM as component classifiers and integrate them into an ensemble classifier using AdaBoost. By adaptively adjusting the parameters, a set of effective RBFSVM classifiers are obtained. The ensemble classifier is used to test the pixels in the next frame and produce a confidence map. The results demonstrated on several sequences, which have high ego motion and low SNR, show the robustness of the proposed method for tracking the target in infrared imagery.

(a)

(b)

(c)
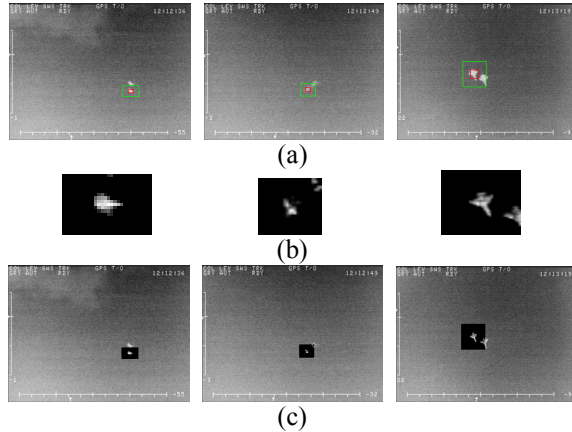
Fig. 2. Target tracking result with the proposed DABSVT algorithm in frame 168, 488 and 1256.

(a)

(b)

(c)
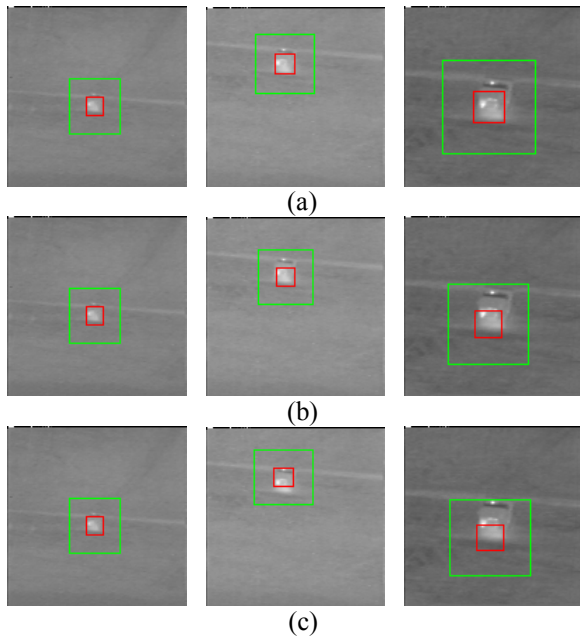
Fig. 3. (a) target tracking result with the proposed DABSVT algorithm in frame 450, 516 and 588; (b) target tracking result using AdaBoost with the least-squares based classifiers; (c) target tracking result using a tracker which trains 5 RBFSVM classifiers in the first frame and does not update the ensemble classifier in each new frame.

6. References

[1] Amer Dawoud and M. S. Alam, "Target Tracking in Infrared Imagery Using Weighted Composite Reference Function-Based Decision Fusion," IEEE Trans. on Image Processing, vol. 15, no. 2, pp.404-410, Feb, 2006

[2] A. Bal and M. S. Alam, "Automatic Target Tracking in FLIR Image Sequences Using Intensity Variation Function and Template Modeling," IEEE Trans. on Instrum. and Meas., vol. 54, no. 5, pp.1846-1852, Oct. 2005

[3] A. Yilmaz, K. Shafique, and M. Shah, "Target tracking in airborne forward looking infrared imagery," Image Vision Comput. J., vol. 21, pp. 623–635, 2003

[4] A. Yilmaz et al., "Target-tracking in FLIR imagery using mean-shift and global motion compensation," in Proc. IEEE Workshop Computer Vision Beyond Visible Spectrum, Kauai, HI, 2001

[5] Jagdish Chandra Patra et al, "A Fast Neural Network-Based Detection and Tracking of Dim Moving targets in FLIR Imagery," Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, pp. 3144-3148, July 31-August 4, 2005

[6] Comanciu, D., Visvanathan R., Meer, P, "Kernel-Based Object Tracking," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.25, no.5, pp. 564-575, May 2003

[7] Shai Avidan, "Ensemble Tracking," Proceedings of International Conference on Computer Vision and Pattern Recognition(CVPR'05), San Diego, June 2005

[8] Vladimir N. Vapnik, "The Nature of Statistical Learning Theory," 2nd ed., Springer-Verlag, Berlin Heidelberg New York, 2000

[9] Xuchun Li, Lei Wang, Eric Sung, "A Study of AdaBoost with SVM Based Weak Learners," Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, pp. 196-201, July 31-August 4, 2005

[10] Robert E. Schapire and Yoram Singer, "Improved boosting algorithms using confidence-rated predictions," Machine Learning, vol. 37, no. 3, pp. 297-336, Dec 1999

[11] Kobi Levi and Yair Weiss, "Learning Object Detection from a Small Number of Examples: The Importance of Good Features," In IEEE Conf. on Computer Vision and Pattern Recognition, 2004

[12] R.N. Braithwaite, B. Bhanu, "Hierarchical gabor filters for object detection in infrared images," In IEEE Conference on Computer Vision and Pattern Recognition, pp. 628–631, 1994

[13] Thomas G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," Machine Learning, vol. 40, no. 2, pp. 139-157, Aug 2000