RANDOM PATCH BASED VIDEO TRACKING VIA BOOSTING THE RELATIVE SPACES

Duowen Chen, Jing Zhang and Ming Tang

Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
{dwchen, jzhang84, tangm}@nlpr.ia.ac.cn

ABSTRACT

In this paper, we propose a new visual tracking method based on the recently popular tracking-as-classification idea. We concentrate on exploring the intra-class variance of the foreground target to construct and update a classification based tracker. In our approach, foreground target is represented by a set of model patches. Different types of features are jointly used to represent those patches. Individual weak learners are trained based on each model patch's relative space. AdaBoost framework is applied to choose those weak classifiers to combine a strong classifier as the tracker for next frame. Moreover, with the new tracking result, the tracker is adjusted adaptively according to the change of scene to keep itself discriminative during the entire sequence. We demonstrate the effectiveness of our approach with comparison results on common video sequences.

Index Terms— Tracking, boosting, image patch, relative space

1. INTRODUCTION

Visual tracking is an important research branch in signal processing and computer vision. There have been several aspects to address video tracking problem. Some papers focus on how to construct the target model [1, 2, 3] to maximize model expressing ability by carefully selecting features (e.g. intensity, color, texture etc.), while others [4, 5] may pay attention to integrating motion cues. Because the robustness and stability of traditional tracking algorithms [6, 7] are not always satisfactory, especially if the foreground target and the background are partially similar in appearance, the target is changing in both appearance and shape, or the background is variable rapidly. As a result, tracking as classification has become one of most popular frameworks resently.

In [8], pixel-based color feature is analyzed; the discrimination of different colors are ranked; and the estimation of target location in new frame is obtained by the weighted combination of individual estimation cues provided by each top ranked colors. Later Avidan [9] proposes another tracking algorithm based on the online construction of a binary classifier to discriminate the target and the nearby background, named as Ensemble Tracking. The author more strictly treats the tracking task as the classification problem, which is achieved by training weak classifiers through least-squares methods, and combining those weak classifiers through the AdaBoost framework. Classifiers are updated adaptively according to new tracking results. It sets an exemplary framework of "tracking as classification" for future work. Later in [10], the authors develop their previous work [8] by considering space arrangement of the neighboring background. Other online adaptive tracking as classification algorithms are presented recently, for instance Grabner *et al.* [11] propose an AdaBoost feature selection algorithm for tracking. And in [12], co-training tracking approach is proposed by tracking the target cooperatively using independent features; classifier of one feature is updated based on the ground truth provided by the tracking prediction of the other feature.

The work all mentioned above treats the foreground target as a single body with parametric discriminative models. Therefore, Lu and Hager [13] propose to model foreground target and background with two sets of random image patches. In their algorithm, random patches are sampled around the target region estimated in the last frame. Each of them is then compared to patches in two model sets; the confidence that how a new patch belongs to the target is computed; and the estimated target location was obtained. In their work, patches inside the same class are treated equally, which may inevitably introduce the assumption that the target can be linearly classified. However that is not always the case in visual tracking, especially when the target class and background class have large intra-class variance.

It is common in the tracking environment that the target itself can be viewed as an integration of several relatively independent components. For example, a pedestrian wearing black trousers and blue T-shirt, carrying a white bag, cannot be easily assumed as a single class and its feature points will not distribute tightly in color feature space. It is with high possibility that from the view of feature space, the points are scattered and even those feature points from background will fall between clusters of target feature distributions. In such case, we can no longer treat the tracking as simple binary linear classification problem, but should investigate feature distribution of the target feature deeper. That's the key problem we want to address in this paper. Similar to Lu [13], we also use random patch instead of pixel to construct the feature vector, because it can contain more types of useful information, for instance, texture. However, the major difference is that, we don't treat those patches and features equally. Given those image patches, we hope to explore the intra-class variance of the target. So we turn to the concept of relative space [14]. From the view of the relative space, we can see more clearly the distribution between patches and types of features, and a simple classifier can be obtained from it. Then we apply AdaBoost framework to choose and combine those weak classifiers adaptively to construct a strong classifier whose objective is to classify the target as a whole unit in the image sequence.

The rest of paper is organized as follows: In the next two sections, we introduce the concept of relative space and how to boost them. In section 4, we describe in detail the tracking and updating

This work was supported by National Natural Science Foundation of China, Grant No. 60572057 and 60835004.

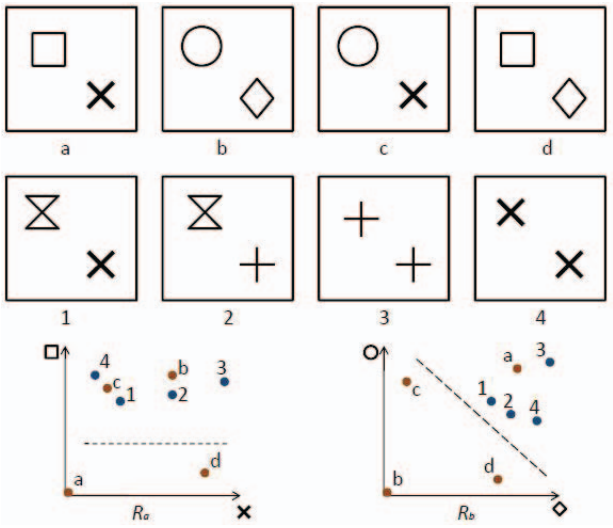

Fig. 1. Simple illustration of relative spaces

steps. Our results on various sequences are presented and discussed in section 5. Section 6 gives conclusion of our work.

2. RELATIVE SPACES

This is the key idea we applied in our tracking method. In [14], they use such idea to categorizing objects which may have large intra-class variation. Similar to object categorization, here we need to categorize the random patches with labels "foreground" and "background". Since in the tracking situation, there are only two general classes, we can construct the relative spaces for those patches simpler and faster. That happens to meet the requirement of real-time visual tracking. We first introduce the concept of relative space briefly.

Because we focus on finding the foreground patch, we construct relative space only for foreground patch set. Given two image patch sets $P_F = \{p_i\}_{1 \le i \le N}$, and $P_B = \{p_j\}_{1 \le j \le M}$, which represent the foreground random patch set with size N and background random patch set with size M respectively. The feature vector of p_i is denoted as $x_i = [x_{i1}, x_{i2}, ..., x_{iK}]^T$, where K is the number of types of features used to express each patch. For each $p_i \in P_F$, we define the raw distance vector between it and all other patches as follows

$$d_{ij} = [\| x_{i1} - x_{j1} \|, \| x_{i2} - x_{j2} \|, ..., \| x_{iK} - x_{jK} \|]^T$$
$$= [d_{ij,1}, d_{ij,2}, ..., d_{ij,K}]^T, p_j \in P_F \cup P_B \quad (1)$$

where, $\| x_{ik} - x_{jk} \|$ is the L^2 distance of kth feature between two patches. With above notation, the construction of relative space is illustrated in Figure 1. Above four represent foreground patches, and the bottom four represent background patches. Two small shapes inside each box represent two types of features and the distance between patches/feature vectors is computed by comparing the corresponding type. The relative spaces for patch a and patch b are in the third row. And the dashed lines can be explained as possible classification surface in that relative space. As we can see from each relative space, although not all in-class samples are classified correctly, it still capture the locally nearby samples. As patch "d" to patch "a", or patches "c" and "d" to patch "b".

3. LEARNING AND BOOSTING

3.1. Learning in Relative Spaces

For simplicity we omit the time stamp sometimes. Given the initial target model set P_F^0 and for each p_i in it, we build a single classifier c_i which is learnt from its own relative space R_{p_i} as depicted in Section 2. The purpose we construct the relative space is that it can linearly discriminative for partial of the positive samples against negative ones. Fisher linear discriminant [15] is designed to find an optimal direction of projection to separate the positive and negative samples. For every other patch p_j, if $p_j \in P_F^0$, we assign patch label $l_i(p_j) = 1$, and otherwise $l_i(p_j) = -1$. The learned projection vector in p_i's relative space is $\alpha_i = [\alpha_{i1}, \alpha_{i2}, ..., \alpha_{iK}]^T$. The projection function is defined as

$$g_i(d_{ij}) = \alpha_i^T d_{ij} \quad (2)$$

$$\alpha_i = (S_i^1 + S_i^2)^{-1}(\mu_i^1 - \mu_i^2) \quad (3)$$

where μ_i^1 and μ_i^2 are the means of the two classes in p_i's relative space, S_i^1 and S_i^2 are the covariance matrices [16]. Considering the sample weights w, these variables can be computed as

$$\mu_i^1 = \frac{1}{\sum_{l_i(p_j)=1} w_j} \sum_{l_i(p_j)=1} w_j d_{ij} \quad (4)$$

$$\mu_i^2 = \frac{1}{\sum_{l_i(p_j)=-1} w_j} \sum_{l_i(p_j)=-1} w_j d_{ij} \quad (5)$$

$$S_i^1 = \frac{\sum_{l_i(p_j)=1} {w_j}^2 (d_{ij} - \mu_i^1)(d_{ij} - \mu_i^1)^T}{\sum_{l_i(p_j)=1} {w_j}^2} \quad (6)$$

$$S_i^2 = \frac{\sum_{l_i(p_j)=-1} {w_j}^2 (d_{ij} - \mu_i^2)(d_{ij} - \mu_i^2)^T}{\sum_{l_i(p_j)=-1} {w_j}^2} \quad (7)$$

which are computed for two classes separately.

3.2. Boosting

Once finished the computation in each relative space, we get N projection vector α_i, and therefore N corresponding weak learners c_i, as in [17]. Then we use AdaBoost framework to choose and combine those weak learners to form a strong classifier C^t. The detail is shown in Algorithm 1.

Algorithm 1 The Boosting Algorithm

Given image patch labels and their feature vectors in the relative spaces, initialize patch weight $w_i = 1/2N$, $1/2M$ for positive foreground samples and negative background samples respectively. $minErr = 0.0$, $iter = 0$

while $minErr \le 0.5$ and $iter \le N$ do

a. Make $\{w_i\}_{i=1}^{N+M}$ a distribution, and $iter = iter + 1$

b. Train each weak classifier c_i, keep the one with minimum error $minErr$, denote as h_{iter}

c. Set weak classifier weight $\beta_{iter} = \frac{1}{2} log \frac{1 - minErr}{minErr}$

d. Update example weights $w_j = w_j e^{\beta_{iter}}$, only for p_j misclassified by h_{iter}

The strong classifier is given by $C(x) = \sum_{i=1}^{iter} \beta_i h_i(x)$

4. TRACKING AND UPDATING

The outline of the algorithm is firstly shown in Algorithm 2.

Algorithm 2 The outline of the tracking algorithm

1. Initialization. Given the first frame F^0, mark the target window W^0, sample and construct model patch set P_F^0 and background set P_B^0
2. Boosting the relative spaces. For each p_i in P_F^0, construct the relative space R_{p_i}, based on which, weak classifier c_i^0 is trained. Then, the strong classifier C^1 is obtained through AdaBoost
3. Iteration. For each frame F^t
   a. Randomly sample patches in F^t nearby W^{t-1} to form the patch set P^t
   b. Compute the confidence value $C^t(p_i)$ for each patch, and construct the confidence map M^t, see Figure 2
   c. Mean-shift from W^{t-1} to find the new location window W^t
   d. Classify each patch in P^t according to W^t and $C^t(p_i)$, form new sample sets P_F^t and P_B^t
   e. Update partial patches in P_F^0, and adjust strong classifier from C^t to C^{t+1}

4.1. Tracking and Classification

Suppose that in two consecutive frames the location of the target doesn't change significantly, which is always the case in most video. Then we can constrain the patch sampling around the target center of the previous frame. Same as [9], we use confidence value to describe the classification result. Each patch will receive a confidence value $C^t(p_i) \in [0, 1]$. And together with the position information of the patch, we can construct a confidence map as in Figure 2. With such figure, we use mean-shift [7] to explore the mode as the new target center, and adjust the tracking window W^{t-1} to W^t.

Given new target location W^t, we assign those patches as positive when they are inside W_t and at the same time have confidence value $C(p_i)$ more than 0.5, others are assigned as negative. That is with classification, new sample sets P_F^t and P_B^t are constructed respectively. Now we are ready for updating.

4.2. Updating

With above mentioned tracking algorithm, we propose our update scheme. The process is two steps. The first one is to add new patches and corresponding relative spaces to target model set, on purpose of replacing those "bad" ones, meanwhile keep the "good" ones. We need a criterion to tell that, inside the model set P_F^0, which ones are good enough to be kept, and which ones need to be replaced. Error of each classifier is used as such criterion here. Each c_i is applied on P_F^t and P_B^t to get the error e_i^t. Then every model patch in P_F^0 has the probability e_i^t to be placed by new one randomly picked from P_F^t. Then we will have a new set of model patches, still denoted as P_F^0. It is natural that the strong classifier for next frame should be adjusted. This is the second step of update, and similar to the operation in Algorithm 1.

Fig. 2. The confidence map, in which the intensity of the pixel represent the value of confidence, more whiter more higher the value is.

5. EXPERIMENTAL RESULTS

In the experiments, two types of features are used, color histogram and histogram of oriented gradient[18]. Of course, under our framework, other types of features can be easily integrated. Patch size is the proportion of the target size. And we use approximately 100 uniformly distributed random patches to model each target. We do experiments on several sequences and compare the results to other methods. Some of the results are shown in Figure 3 and Figure 4. Figure 3 shows the comparison results of our method (right column) to basic meanshift (left column) on a PETS 01 sequence. A pedestrian, with three major types of appearance (bag, coat and trousers) is walking past a parking lot, where the background cars are very distracting. As the tracking window is a bit larger than the real object, some background information is contained in the model construction step. In the meanshift method, the whole target is modeled with single histogram, and is easily distracted by the background. While in our method, the combination of partial information (relative space) makes the tracker more robust and discriminative. In Figure 4, we compare our method to [13]. Two methods both use patch based idea with update scheme. The result shows that our method is more resistant to model drift.

6. CONCLUSION

In this paper, we have proposed an efficient and robust tracking algorithm by boosting the relative spaces. We have demonstrated our algorithm for tracking moving object in various challenging sequences. By boosting the relative space of each model patch, we can explore the subclass information of the target, and at the same time construct a strong classifier for tracking the whole target. The problem of large intra-class variance is handled well under such framework. Moreover, an update scheme is naturally devised to make the tracker more robust in case the tracking scene changes significantly. It learns adaptively and also avoids model drift effectively.

7. REFERENCES

[1] M.J. Lucena, J. Fuertes, and N.P Blanca, "Real-time tracking using multiple target models," *Proc. Iberian Conference on Computer Vision and Image Analysis*, vol. 1, pp. 20–27, 2005.

[2] S.M. Nejhum, J. Ho, and M.H. Yang, "Visual tracking with his-

tograms and articulating blocks," *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[3] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on lie algebra," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 728–735, 2006.

[4] Z. Yin and R. Collins, "Spatial divide and conquer with motion cues for tracking through clutter," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 570–577, 2006.

[5] S Ali, V. Reilly, and M. Shah, "Motion and appearance contexts for tracking and re-acquiring targets in aerial video," *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[6] M. Isard and A. Blake, "CONDENSATION-conditional density propagation for visual tracking," *International Journal on Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[7] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[8] R. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 27, no. 10, pp. 1631–1643, 2005.

[9] S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 29, no. 2, pp. 261–271, 2007.

[10] Z. Yin and R. Collins, "Spatial divide and conquer with motion cues for tracking through clutter," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 570–577, 2006.

[11] H. Grabner and H. Bischorf, "On-line boosting and vision," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 260–267, 2005.

[12] F. Tang, S. Brennan, Q. Zhao, and H. Tao, "Co-tracking using semi-supervised support vector machines," *Internationl Conference on Computer Vision*, 2007.

[13] L. Lu and D.G. Hager, "A nonparametric treatment for location/segmentation based visual tracking," *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[14] Y. Ouyang, M. Tang, J. Wang, H. Lu, and S. Ma, "Boosting relative spaces for categorizing objects with large intra-class variation," *Proc. 16th ACM International Conference on Multimedia*, pp. 663–666, 2008.

[15] R.A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, pp. 179–188, 1936.

[16] L. Laptev, "Improvements of object detection using boosted histograms," *British Machine Vision Conference*, 2006.

[17] P. Viola and M. Jones, "Rapid object ddetection using a boosted cascade of simple features," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 511–518, 2001.

[18] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, 2005.

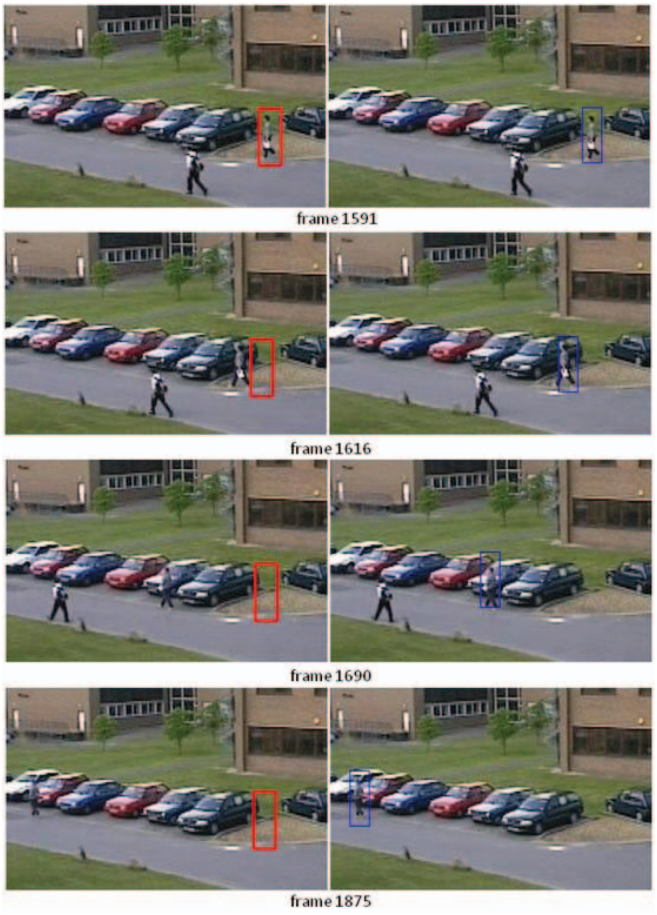

Fig. 3. Left column: Tracking results of basic meanshift. Right column: Tracking results of our method.

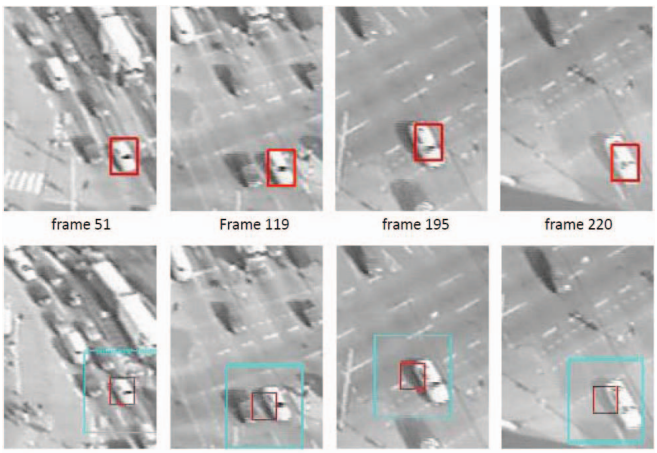

Fig. 4. Comparison results. Top row: Tracking results of our method. Bottom row: Tracking results by [13]'s method. Both methods are random patch based and with update scheme.