ROBUST MULTI-PATCH TRACKING

Shanxin Yuan[⋆] *Jun Miao*[⋆] *Laiyun Qing*[⋆,†]

[⋆] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[†]University of Chinese Academy of Sciences, Beijing 100190, China

ABSTRACT

In this paper, we propose a robust and fast multi-patch visual tracking algorithm within the Bayesian inference framework. The target template is initialized by selecting the object in the first frame manually and dividing it into small patches. For one certain frame, target candidates are sampled with the state transition model. Each candidate is divided into patches in the same way as the target template. By comparing the candidate's patches with the corresponding template patches, we can get the candidate's likelihood. The tracking result is the candidate which Maximum a Posteriori estimation. After that, tracking is continued using the Bayesian state inference and template update. Our approach can handle appearance variation, occlusion, illumination change, scale variation, rotation and cluttered background. The tracker is fast and performs favorably against several state-of-the-art trackers on challenging sequences.

Index Terms— Visual tracking, Bayesian inference, multi-patch, fast, robust

1. INTRODUCTION

Visual tracking is an important topic in the computer vision community with a broad range of applications, including surveillance, vehicle navigation, activity analysis and human-computer interaction. Recently, it has been addressed in real-world scenarios rather than the lab environment by many computer vision researchers [1]. In this paper, we propose a robust and fast multi-patch tracking algorithm within the Bayesian inference framework by making use of local patches of the tracking target. The challenges we deal with are occlusion, illumination change, scale variation, rotation and cluttered background (See Fig.1).

The patch representation is a well known technique in visual tracking field. Adam *et al.* [2] proposed a patch-based tracker to handle occlusions using a representation with histograms of local patches. Each patch votes for possible locations by comparing its histogram with the corresponding one in the template. Nevertheless, the template is not updated and thereby the tracker is sensitive to large appearance variations and illumination change. Shu *et al.* [3] learned part-based person-specific SVM classifiers which capture the articulations of human body. With the patch-based model, they can handle partial occlusions. However, their method depend much on the human detector and person-specific classifiers which need a training stage. Our multi-patch tracker explores the patches' locality and confidence values. By selecting discriminative patches and rejecting those occluded ones, our tracker can handle partial occlusion robustly. The proposed approach need no specific model for the target and no training stage, thus it can be used to track generic objects.

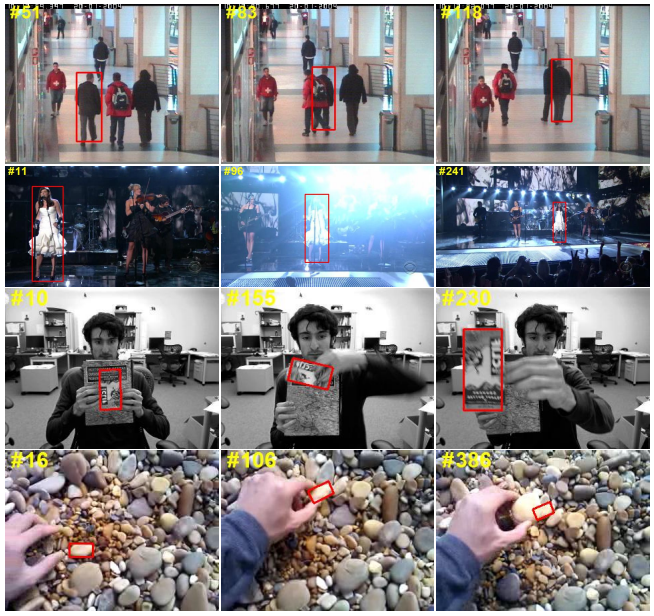

Fig. 1. The challenges our tracker deal with. They are heavy occlusions (*Caviar*), illumination change (*Singer1*), scale variation (*Singer1, Cliffbar*), rotation (*Cliffbar, Stone*) and cluttered background (*Stone*). The rows from top to bottom are *Caviar, Singer1, Cliffbar* and *Stone*, respectively. The red boxes are the ground truth.

Numerous works [4, 5, 6, 7] have demonstrated that adaptive appearance models, which evolve during the tracking process, are essential for the trackers to adapt to appearance

variations and illumination change. Ross *et al.* [5] presented an incremental visual tracker that learns a low-dimensional subspace representation, efficiently adapting to significant variations of the object's appearance and the surrounding illumination. Nevertheless, their holistic appearance model is less effective in dealing with heavy occlusion. Babenko *et al.* [6] proposed a Multiple Instance Learning method to update the appearance model through choosing positive and negative bags, with the true target included in the positive bag. Kalal *et al.* [7] used P-N learning, which is an approach for processing of labeled and unlabeled data, to explore the structure of the unlabeled samples. The positive and negative constraints that they used help to select potential samples for update. Our template updating method takes occlusions into account, the occluded patches are not used to account for the appearance change of the tracking target.

Our approach has the following advantages:

1). Since the only feature we use are the intensity values, our tracker is fast and can run at 12 frames per second.

2). The multi-patch representation of the target can handle partial occlusions effectively, allow selecting discriminative patches and guide the template update.

3). The state transition model considers rotation and scale variation explicitly.

2. ROBUST MULTI-PATCH TRACKING

In this section, we present our robust multi-patch tracking algorithm in detail. Our tracker contains four modules: object representation, distance calculation, occlusion handling and template update.

2.1. Bayesian Object Tracking Formulation

In this paper, object tracking is considered as a Bayesian inference task in a Markov model with hidden state variables [5]. Given a target's observation set $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_t\}$ up to the *t*-th frame, the hidden state variable $\mathbf{x}_t$ can be estimated recursively,

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1}, \tag{1}$$

where $p(\mathbf{y}_t|\mathbf{x}_t)$ is the observation model that estimates the likelihood of the observation $\mathbf{y}_t$ predicted by hidden state $\mathbf{x}_t$. $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the transition model which describes the temporal correlation of the target states between two consecutive frames. In this work, the hidden target state is defined as $\mathbf{x} = (x, y, \theta, s, \alpha, \phi)$, where x, y denote the center location of the target, θ, s, α, ϕ denote rotation angle, scale, aspect ratio and skew respectively. We model the target motion between two consecutive frames by applying the affine transformation with six parameters. Our state transition model is assumed to be Gaussian distributed:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = N(\mathbf{x}_t; \mathbf{x}_{t-1}, \Psi), \tag{2}$$

where Ψ is a diagonal covariance matrix whose elements are the standard deviations for the six parameters. With the posteriori probability $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ computed by the observation model and the transition model, the optimal state of the tracked target can be obtained by Maximum A Posteriori estimation over N samples at time t by

$$\hat{\mathbf{x}}_t = \arg\max_{\mathbf{x}_t^i} p(\mathbf{x}_t^i|\mathbf{y}_{1:t}), \quad i = 1, 2, ..., N, \tag{3}$$

where $\mathbf{x}_t^i$ represents the i-th sample of the object state $\mathbf{x}_t$.

The likelihood of the observation $\mathbf{y}_t^i$ predicted by state $\mathbf{x}_t^i$ is inversely proportional to its distance,

$$p(\mathbf{y}_t^i|\mathbf{x}_t^i) \propto \frac{1}{D^i}, \quad i = 1, 2, ..., N, \tag{4}$$

where D^i denotes the distance between observation $\mathbf{y}_t^i$ and target template $\mathbf{T}$.

2.2. One Frame Tracking

In each frame, N candidates are sampled around the tracking result in the previous frame with the transition model described in the previous section.

2.2.1. Object Representation

The tracking object's observation is represented by a local patch-based model in our tracking algorithm. For simplicity and efficiency, we use only the intensity information to represent the local patches. To build the target template $\mathbf{T}$, the tracking object is selected manually from the initial frame and normalized to the canonical size of 32×32. Overlapped sliding windows are used on the normalized target to obtain M patches, each of which is converted to a vector $\mathbf{q}_j \in R^{l \times 1}, j = 1, 2, ..., M$, where l denotes the size of the patch. The target template can be represented by

$$\mathbf{T} = (\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_M)$$

When a new frame arrives we sample N target candidates through the transition model and every candidate is represented in the same way as the target template, i.e.

$$\mathbf{y}^i = (\mathbf{p}_1^i, \mathbf{p}_2^i, ..., \mathbf{p}_M^i), \quad i = 1, 2, ..., N$$

where $\mathbf{p}_j^i \in R^{l \times 1}$ denotes the j-th patch of the i-th candidate.

2.2.2. Distance Calculation

For target candidate $\mathbf{y}^i = (\mathbf{p}_1^i, \mathbf{p}_2^i, ..., \mathbf{p}_M^i), i = 1, 2, ..., N$, we calculate the distance D^i between $\mathbf{y}^i$ and the target template $\mathbf{T}$ in this section. The patch distances $\|\mathbf{p}_j^i -$

$\mathbf{q}_j\|_2, j = 1, 2, ..., M$, between the candidate's patches $\mathbf{p}_j^i$ and the corresponding template patches $\mathbf{q}_j$ are calculated first, then the $K(K < M)$ smallest patch distances are accumulated,

$$D^i = \sum_{j=1}^{K} \|\mathbf{p}_{s(j)}^i - \mathbf{q}_{s(j)}\|_2, \quad i = 1, 2, ..., N, \tag{5}$$

where $\|\cdot\|_2$ denotes ℓ_2-norm, $s(j)$ is the index of the candidate's patch whose distance value is the j-th smallest one.

2.2.3. Occlusion Handling

In order to deal with occlusions, the occluded patches are excluded when calculating the distance D^i. To find an occluded patch, our approach make use of the distance value between the candidate patch and the corresponding one from the target template. A patch is regarded as occluded if its distance is larger than a threshold d_0.

For each target candidate $\mathbf{y}^i, i = 1, 2, ..., N$, a vector $\mathbf{O}^i \in R^{M\times 1}$ is obtained to indicate the occlusion. Each element value of $\mathbf{O}^i$ denotes the occlusion condition of the corresponding patch and is obtained by

$$O_j^i = \begin{cases} 1 & if \quad d_j^i > d_0 \\ 0 & otherwise \end{cases} \tag{6}$$

where O_j^i is the j-th patch's occlusion indicator, $d_j^i = \|p_j^i - q_j\|_2, j = 1, 2, ..., M$ is the Euclidean distance between the j-th patch of the candidate $\mathbf{y}^i$ and the corresponding template patch, d_0 is the predefined threshold.

For all the N occlusion indicating vectors $\mathbf{O}^i, i \in 1 : N$, we select

$$\mathbf{O} = \arg\min_{\mathbf{O}^i} \sum_{j=1}^{M} O_j^i, \tag{7}$$

to represent the occlusion condition of the state $\mathbf{x}_t$. In other word, $\mathbf{O}$ is used to model the truth occlusion condition of the target.

By integrating the occlusion indicating vector $\mathbf{O}$ into (5), a more reasonable distance function is obtained,

$$D^i = \sum_{j=1}^{K} (1 - O_{s(j)}) \|\mathbf{p}_{s(j)}^i - \mathbf{q}_{s(j)}\|_2, \tag{8}$$

where $O_j, j = 1, 2, ...M$ denotes the j-th element of $\mathbf{O}$.

2.2.4. Template Update

Since the tracking target's appearance and the surrounding illumination often change significantly, a template update module is necessary. We use a simple technique similar to what Zhong [8] used. In order to recover the object from occlusion and let the tracker adapt to the appearance variation, illumination change, our target template is updated by

$$\mathbf{T}_n = \mu \mathbf{T}_f + (1 - \mu)\mathbf{T}_p, \quad if \quad \tau_n < \tau_0, \tag{9}$$

where the current target template $\mathbf{T}_n$ is composed of the template at the first frame $\mathbf{T}_f$ and the previous tracking result $\mathbf{T}_p$ according to the weights assigned by the constant μ. The variable $\tau_n = |\mathbf{O}|$ denotes the occluded patch number of the tracking result in the current frame. Template update is performed every five frames as long as τ_n is smaller than a predefined constant τ_0. Simple as it is, our template update module can preserve the target appearance from the initial frame and take appearance variation, illumination change into consideration.

Table 1. Average center location error (in pixel). The best and second best are shown in red and blue fonts.

	IVT	$\ell 1$	PN	VTD	MIL	Frag	Our
Caviar1	45.2	119.9	5.6	**3.9**	48.5	5.7	**1.3**
Stone	**2.2**	19.2	8.0	31.4	32.3	65.9	**2.0**
Caviar2	8.6	**3.2**	8.5	4.7	70.3	5.6	**3.3**
Caviar	66.2	65.9	**53.0**	60.9	83.9	94.2	**3.2**
Singer1	8.5	4.6	32.7	**4.1**	15.2	22.0	**4.0**
Occlusion1	9.2	6.5	17.6	11.1	32.3	**5.6**	**3.8**
Occlusion2	**10.2**	11.1	18.6	10.4	14.1	15.5	**7.0**
Car11	**2.1**	33.3	25.1	27.1	43.5	63.9	**3.8**
Cliffbar	24.8	49.6	**11.3**	34.6	13.3	48.7	**2.2**
Jumping	36.8	92.4	**3.6**	63.0	9.9	58.4	**3.5**

Table 2. Average overlap rate base on [9]. The best and second best results are shown in red and blue fonts.

	IVT	$\ell 1$	PN	VTD	MIL	Frag	Our
Caviar1	0.28	0.28	0.70	**0.83**	0.25	0.68	**0.89**
Stone	**0.66**	0.29	0.41	0.42	0.32	0.15	**0.60**
Caviar2	0.45	**0.81**	0.66	0.67	0.26	0.56	**0.78**
Caviar	**0.21**	0.20	**0.21**	0.19	0.19	0.19	**0.86**
Singer1	0.66	0.70	0.41	**0.79**	0.34	0.34	**0.84**
Occlusion1	0.85	0.88	0.65	0.77	0.59	**0.90**	**0.93**
Occlusion2	0.59	**0.67**	0.49	0.59	0.61	0.60	**0.77**
Car11	**0.81**	0.44	0.38	0.43	0.17	0.09	**0.72**
Cliffbar	**0.56**	0.20	0.38	0.33	0.46	0.13	**0.80**
Jumping	0.28	0.09	**0.69**	0.08	0.53	0.14	**0.74**

3. EXPERIMENTAL RESULTS

In this section we present our experimental results. Ten image sequences are used, they provide a wide range of significant challenges including heavy occlusion, motion blur, large illumination change, scale variation, rotation, pose change and cluttered background.

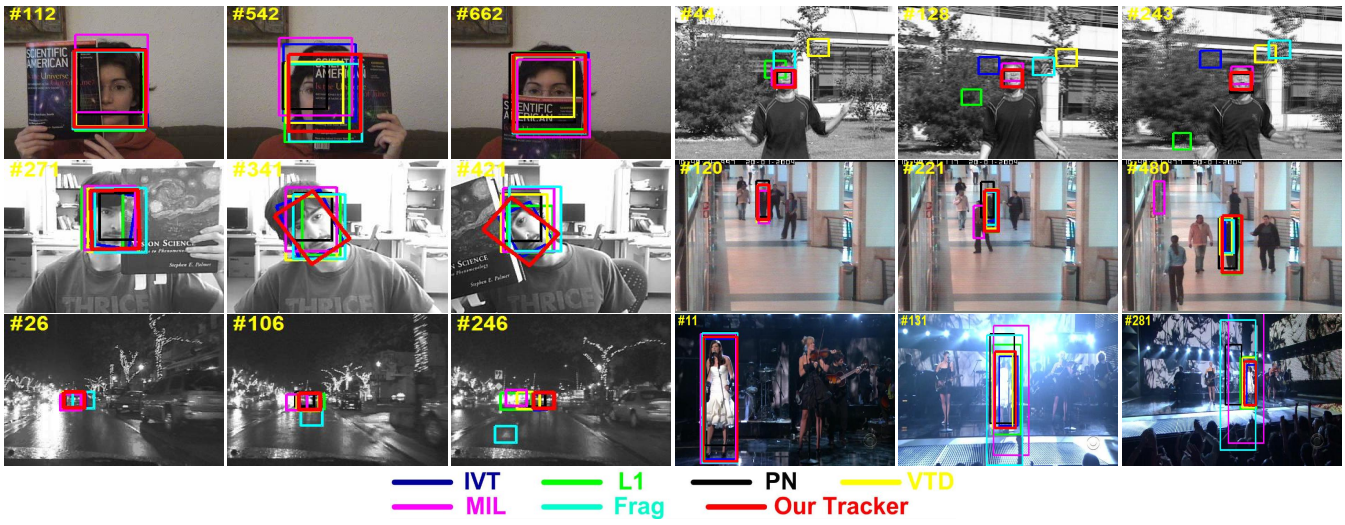

Fig. 2. Qualitative comparison with six state-of-the-art tackers. From left to right, the first row is *Occlusion1* and *Jumping*, the second row is *Occlusion2* and *Caviar2*, the third row is *Car11* and *Singer1*.

Our tracker is evaluated against six state-of-the art trackers with the same initial position of the target. These trackers are FragTrack [2], IVT [5], MIL [6], $\ell 1$ [10], PN [7]and VTD [11].

Our tracker is implemented in MATLAB and runs at 12 frames per second on a 3.10 GHz Dual Core PC with 2GB memory. Each observation is normalized to 32×32 pixels and then divided into overlapped 8×8 local patches with 4 pixels as the step length, the total number of the local patches is $((32-8)/4+1)\times((32-8)/4+1)) = 49$. We sample $N = 600$ target candidates for each frame through the transition model. We choose the top $K = 45$ patches with the smallest patch distances to calculate the candidate's distance D^i, in order to prevent degeneration, at least 36 patch distances are used regardless of the occlusion condition $\mathbf{O}$. The variables d_0 in Eq.6 and τ_0 in Eq.9 are fixed to 0.5 and 4 respectively. The update rate μ in Eq.9 is fixed to 0.95.

Quantitative Comparison: We use two evaluation criteria to quantitatively assess our tracker's performance. The first one is the center location error which measures the difference between the tracked and the ground truth center locations. The second one is the overlapping rate $score = \frac{area(R_T \bigcap R_G)}{area(R_T \bigcup R_G)}$ [9], where R_T and R_G are the tracking result and the corresponding ground truth, respectively, when the score is above 0.5 the object is regarded as being successfully tracked. The results are shown in Table 1 and Table 2. Overall, our tracker performs favorably against state-of-the-art trackers.

Qualitative Comparison: Qualitative comparison is shown in Fig 2. Results on sequence *Occlusion1, Occlusion2, Caviar2* show that our occlusion handling module works very well. When calculating candidate distances, only those un-occluded patches are used. *Singer1* and *Car11* provide challenges like illumination change and cluttered background, *Jumping* shows the motion blur. Our tracker can handle these challenges thanks to the template update module. Although moving objects in sequence *Singer1, Occlusion2* and *Caviar2* experience scale and rotation change, our tracker can robustly track them with the explicit state transition model.

4. CONCLUSIONS AND DISCUSSION

In this paper we propose a robust and fast multi-patch tracking algorithm. Local patch representation is adopted with an occlusion handling module which enables our tracker to better handle heavy occlusion and to guide the template update. Online update scheme enhances the proposed method to adaptively account for appearance variation and illumination change in dynamic scenes. Scale change and rotation are treated explicitly through the state transition model. Qualitative and quantitative evaluations on ten challenging sequences demonstrate the robustness of our tracker in dealing with heavy occlusion,motion blur, large illumination change, scale variation, object rotation and cluttered background.

The current model can not handle the cases like large geometric appearance change and severe out-of-plan rotation, which would be improved in our future work.

5. ACKNOWLEDGEMENTS

This research is partially sponsored by National Basic Research Program of China (No.2009CB320900), and Natural Science Foundation of China (Nos.61070116, 61070149, 61001108, 61175115, and 61272320).

6. REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm Computing Surveys (CSUR)*, vol. 38, no. 4, pp. 13, 2006.

[2] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2006, vol. 1, pp. 798–805.

[3] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, "Part-based multiple-person tracking with partial occlusion handling," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 1815–1821.

[4] A.D. Jepson, D.J. Fleet, and T.F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296–1311, 2003.

[5] D.A. Ross, J. Lim, R.S. Lin, and M.H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1, pp. 125–141, 2008.

[6] B. Babenko, M.H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.

[7] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-n learning: Bootstrapping binary classifiers by structural constraints," in *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 49–56.

[8] W. Zhong, H. Lu, and M.H. Yang, "Robust object tracking via sparsity-based collaborative model," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 1838–1845.

[9] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[10] X. Mei and H. Ling, "Robust visual tracking using ℓ 1 minimization," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 1436–1443.

[11] J. Kwon and K.M. Lee, "Visual tracking decomposition," in *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 1269–1276.