A UNIQUE TARGET REPRESENTATION AND VOTING MECHANISM FOR VISUAL TRACKING

Changlin Xiao, Alper Yilmaz

The Ohio State University
Photogrammetric Computer Vision Laboratory
Columbus, OH, USA, 43210

ABSTRACT

One of the main problems in visual tracking is how to robustly represent the target. In this paper, we propose a simple yet efficient method that provides a unique ”target representation” by generating discriminative non-uniform subspaces from the feature space which we refer to as cells. Each cell is attributed with a measure that highlights how likely it describes the target or the background. In addition, we keep a codebook of spatial locations of the features which are mapped to the cell similar to that of the R-Table in generalized Hough transform. Using the uniqueness measure as weight, the target center is estimated by using a modified Hough voting scheme to address non-rigid deformations. In the experiment, we use color as the pixel's descriptor and demonstrate comparable performance on the Online Tracking Benchmark (OTB) dataset respect to the other state-of-the-art.

Index Terms— computer vision, visual tracking, voting

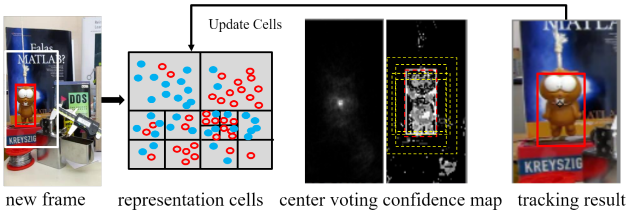


Fig. 1: The flow diagram of the proposed tracking framework. For each new frame, we extract descriptors for features within the search window(white rectangle) generated based on past target location. The mapped cells and their likelihood measures are used to generate a target center voting map and an appearance confidence map (red circle means the feature from target, the blue dot means the feature from background). Finally, the descriptors within the tracked bounding box are used to update the cells.

1. INTRODUCTION

The visual trackers can be categorized into two classes: generative and discriminative. Generative methods focus on how to represent the target by feature descriptions [1, 2, 3]. These trackers, however, are vulnerable to background clutter and occlusions. On the other hand, the discriminative trackers consider the distinctiveness between the target and the background and they train classifiers to distinguish features [4, 5, 6]. However, online or incremental training are time-consuming and slow down the tracker throughput.

Recently, a correlation filter based tracker (KCF) have achieved robust tracking performance with high speed[7]. The KCF tracker only considers the target's previous appearance, and it fails in cases of large scale and appearance changes. Also, deep-learning, such as CNN has become popular [8, 9, 10]. These methods use the neural network and classifier to combine feature selection and classification steps together. However, the online training is still a problem.

So, in this paper, we seek the answer to the problem of finding an efficient method to generate discriminative information from limited data while eliminating the training of classifiers. There are several trackers have done some works, like in [11], Collins et al. use histograms to estimate the feature distribution for background and target separately. In [12], Horst et al. use the color histogram to represent colors and use them to prevent the drift problem. Contrast to these methods, we directly estimate the distribution of features in the target and background using a voting scheme, also, we include spatial information and use Hough voting to estimate the center of the target.

Also, not similar to the generative or discriminative trackers, we dynamically represent the target just by its ”unique parts” instead of the complete target. In our paper, the unique parts are defined as have features that only or most on target. This is acquired by dividing the feature space into a number of small cells and estimate how likely the features in the cell are from target or background. Besides that, the spatial information of each feature is assigned to each cell as a codebook to facilitate a target center voting scheme. So, for any feature in the new frame, we can estimate how likely it belongs to target and generate a confidence map as well as a voting map similar to generalized Hough voting scheme [13, 14]. Finally, each cell's likelihood measure and codebook are up-

978-1-5090-2175-8/17/$31.00 ©2017 IEEE

dated based on the tracking target. The flow diagram of the tracking algorithm is given in Fig.1.

The main contributions of this paper are: 1. Represent target by unique parts and offers an efficient way to estimate the uniqueness of a feature. 2. Offer a novel tracking method using feature's confidence and spatial information to estimate target area and center.

2. METHODOLOGY

2.1. Unique Target Representation

Unlike the previous work, we define unique target representation as a set of cells in the feature space that contain the uniqueness of features and a spatial codebook. The cells are generated by recursively dividing the feature space until each cell either mostly represents the target or background. Actually, a cell C_i can be considered as a cluster of features, then, the task of extracting unique representation cells becomes finding unique clusters.

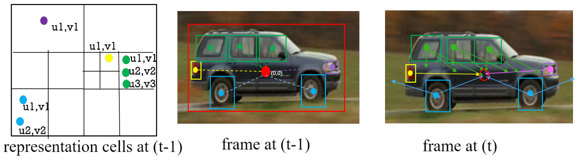
Let the target be composed of a set of feature with descriptors $T = (f_1, f_2, f_3, \ldots)$ and its immediate background features $B = (f_1, f_2, f_3, \ldots)$. In the first frame, the initial target region can offer the initial labels l_i for each feature. Then, these features can generate vote in the feature space in favor of its label l_i. We consider a vote in favor of the target as positive $\beta^{+}(f_i) = \{1 | l_i = target\}$. On the contrary, a background vote is considered as negative $\beta^{-}(f_i) = \{-1 | l_i = background\}$ which in turns unlearns the cell. As a consequence, the uniqueness of a cell is measured by:

$$d\left(C_i\right) = \frac{1}{n} \sum_{j=0}^{n} \beta^{\{-,+\}}\left(f_j\right), \tag{1}$$

where n is the number of features mapped to the cell, $-1 \leq d\left(C\right) \leq 1$. The closer $d\left(C_i\right)$ to 1, the more the cell represents the target; otherwise, the closer $d\left(C_i\right)$ to -1, the more it represents the background. Hence, a unique target representation is generated as the set of cells that have positive uniqueness:

$$U_{target} = \left\{C_i | d(C_i) > \theta_u\right\}, \tag{2}$$

where θ_u represents the threshold that is used to select unique cells. In this model, robust target tracking can only happen when the unique representation U_{target} contains many cells with values closer to 1. This requirement can be easily met when the cells are smaller, however, it will cause over fitting and is not robust to noise. Hence, we resolve this problem by introducing a divide and conquer strategy to dynamically partition the feature space into cells of different sizes. The algorithm starts by evenly dividing the feature space into 2^k cells, k is the dimensionality of the feature space. The algorithm proceeds by checking a cells uniqueness against threshold, θ_u, and recursively divides it into 2^k sub-cells until the uniqueness criteria is met. The recursive division is concluded when the number of mapped descriptors N_i is less than a number N_t or its uniqueness is larger than θ_u.



representation cells at (t-1) frame at (t-1) frame at (t)

Fig. 2: The spatial codebook recording and voting. The codebook records each feature's distance to target center as (u,v) and help to find the new center by voting all (u,v) in the cell.

Checking Feature Descriptor: In order to have a discriminative representation, it is imperative to have cells that support the target appearance, and that a good number of feature mapping to these cells. For this purpose, we test if the feature descriptor can represent most parts of the target by:

$$F_e\left(B_x, I\right) = \frac{S_t - S_b}{S_x}, \tag{3}$$

where B_x is the initial target box, I is the image, S_x is the size of initial target box, S_t is the number of feature identified as target in the target box, S_b is the target feature identified as target outside the target box. Using this equation, we can have three different decisions on trackability:

$$F_e(B_x, I) = \begin{cases} > 0.5 & \text{Trackable} \\ > 0.2 & \text{May be tracked} \\ \text{otherwise} & \text{Cannot be tracked} \end{cases} \tag{4}$$

2.2. Target Tracking

Center Estimation: In order to encode spatial information, we consider encoding the spatial information, $\mathbf{s}_i = (u_i, v_i)^{\top}$ carried by each individual feature f_i relative to target's center. Considering that a cell contains multiple individual features, their respective spatial locations generate a spatial codebook which serves as a look-up table similar to the R-Table GHT. In contrast to the GHT voting, we consider a weighted voting approach which relates to the uniqueness of the target representation. In Fig.2, we display the voting method for the target. The coordinates with the highest vote represent the target center. For the tracking in the new frame, we estimate the best location of the target center u by:

$$score\left(u\right) = \sum_i d(C(f_i)) \sum_{s_k \in C(f_i)} g\left(u, s_k\right) \tag{5}$$

where $C\left(f_i\right)$ is the cell which f_i belongs to. $g(u, s_k)$ is indication function, when s_k vote to u it gives 1, otherwise 0.

The non-rigid motion may cause the voting center gathering as area instead of point. For this, we use Gaussian filter to regenerate the voting map and find the center as:

$$\hat{u}_t = \arg\max\ \left(Score(u_i)\right), \tag{6}$$

$$score\left(u_i\right) = \sum_{u_k \in u_i} w\left(\mathrm{u}_k\right) * score\left(u_k\right), \tag{7}$$

$$w\left(\mathrm{u}_k\right) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\left(\mathrm{u}_k - \mathrm{u}_i\right)\right)), \tag{8}$$

where u_i is the target center candidate, u_k is voting center which around u_i. If the final center score smaller than u_v, the image center is assumed to be voting center. Also, for unique cells, the number of features mapped from the target shouldn't drastically change. Based on this assumption, if

$$\frac{\mathrm{Size}\left(\mathrm{C}_{old}\right)}{\mathrm{Size}\left(\mathrm{C}_{new}\right)} > \theta_h,\ or\ , \frac{\mathrm{Size}\left(\mathrm{C}_{old}\right)}{\mathrm{Size}\left(\mathrm{C}_{new}\right)} < \theta_l, \tag{9}$$

the unique features in cell C is considered not to be trusted and are removed from the representation. Here θ_h and θ_l are high and low threshold defining allowed size changes.

Target Area Estimation: In order to estimate the target area, we generate a confidence map from the appearance and spatial information encoded in the target representation. Let μ_t be the new target center and s_{t-1} be the target scale at frame $t-1$. Each pixel (u_i, v_i) within the search window centered around μ_t generates a descriptor f_i with uniqueness value $\theta_u \leq d(C_i) \leq 1$. In addition to appearance, the spatial codebook for C_i provides all acceptable displacements $\mathbf{s}_j$ for the mapped features. These domain $\mathbf{s}_j$ and range $d(C_i)$ information provide a means to compute the likelihood of the feature to be a part of the target or the background:

$$v(f_i) = d(C_i) \times \exp\left(-\frac{(u_i - \hat{u})^2 + (v_i - \hat{v})^2)}{2\sigma^2}\right) \tag{10}$$

where $(\hat{u}, \hat{v}) = \min_{(u_j, v_j)}(u_i - u_j)^2 + (v_i - v_j)^2$ for $(u_j, v_j) \in \mathbf{s}_i$ and $-1 < v\left(f_i\right) < 1$. This likelihood value will be computed for each feature in the new frame and resulting a confidence map.

Target location can be presented as (s_x, s_y, μ_x, μ_y), where s is scale per axis and μ is the target center. At each new frame, the true target state then becomes the location that maximizes the posterior estimate given by:

$$\hat{X}_t = \arg\max p(X_t|Y_{1:t})$$

$$= \arg\max p(Y_t|X_t) \int p(X_t|X_{t-1}) p(X_{t-1}|Y_{1:t-1}) dX_{t-1} \tag{11}$$

where the new state $\hat{X}_t$ depends on past previous observations $Y_{1:t}$. This formulation can be divided by two parts: the observation model and the motion model. For the observation model, we use the confidence map: $p(Y_t|X_t) \propto C(X_t)$. As for the motion model, we consider the flow of the target center $(\dot{\mu}_x, \dot{\mu}_y)$ and changes in the target scale $(\dot{s}_x, \dot{s}_y)$ which is modeled by normal distribution $p(X_t|X_{t-1}) \propto N(\dot{\mu}_x, \dot{\mu}_y, \dot{s}_x, \dot{s}_y; \psi)$, where ψ is the covariance matrix. In our implementation, we compute the maximum posteriori estimate by testing a series of possible scale changes given the target's previous scale, like yellow bounding boxes in Fig. 1. The score of multiple scales can be swiftly computed using integral image formulation [15] without increasing computational cost.

2.3. Representation Update

Whether to update the target representation is deponent on how well the previous representation U_{object} is preserved in the new frame. In order to facilitate this, we estimate a new representation, $\hat{U}$, using the new frame and the estimated target state. We measure the tracking performance Λ as the similarity between U_{object} and $\hat{U}$ by:

$$\Lambda(U_{object}, \hat{U}) = \frac{1}{n} \sum_{i=1} \delta(C_{U_{object}}(i), C_{\hat{U}}(i)) \tag{12}$$

where $\delta(.)$ is delta dirac function and $\Lambda(.)$ ranges from 0 to 1 respectively denoting bad to good performance.We achieve the representation update by updating each cell using:

$$C_{new}(i) = \Lambda^k C_{\hat{U}}(i) + (1 - \Lambda^k) C_{U_{object}} \tag{13}$$

where the performance measure serves as the learning factor for the target in the new frame. The dynamic updating algorithm can help to avoid the damage of bad tracking and help to recover to true target.

3. EXPERIMENTS

In the experiments, we use color as the descriptor of a pixel(see setting details in supplemental materials). We test the performance of the proposed approach using a large dataset called online tracker benchmark (OTB) [16]. We also compared the proposed tracker with other popular trackers including 29 trackers in [16] and KCF [7]. Since our tracker in the experiment uses color, we compared it with another color based state-of-the-art trackers: TGPR [17].

3.1. Experimental Result

As mentioned in section 2.1, the proposed method can check color's trackability of the target for the provided sequences. After test the trackability, the tracking results from (OTB) is given in Fig. 3. As we can see, the proposed approach is the best tracker compare to all others. Especially in fast motion, motion blur and occlusion sequences(see details in supplemental materials).

Similar to the idea of patch based tracking, proposed approach can track the center of the target which is partly covered, like in Fig.4. The proposed representation can estimate the target center with high accuracy which results in high confidence state estimation. In the case when the occlusions

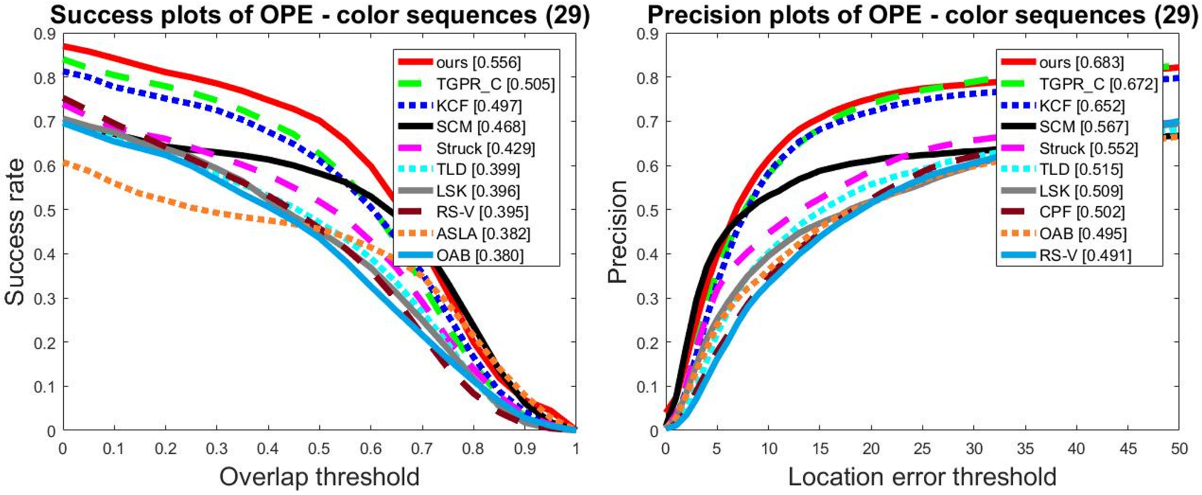


Fig. 3: The comparative tracking results generated using the OTB tools.

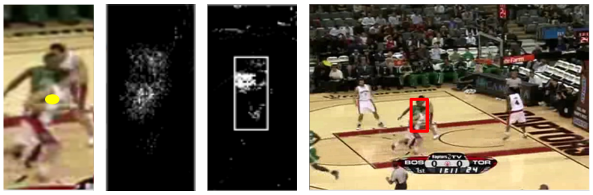


Fig. 4: Tracking under occlusion. Despite the target is covered by another player, the voted center is correct and the resulting confidence map provides adequate information to estimate the target state.

happen, the evaluation scores used to update the target representation are lower than none-occlusion cases, which inhibits model update, hence does not damage representation. This property allows the tracker to reacquire the target back once the occlusion is completed.

An implicit advantage of the unique target representation is to suppress cells with confusing descriptors and only focus on unique cells. Like in Fig.5, the color from the green shorts of the runner are confused with the grass. As can be seen in corresponding confidence maps, the confusing cells in the representation are suppressed resulting in a cleaner target model in the following frames.

The center estimation in our tracker is based on the codebook which can be affected by target deformation and out the plane rotation. Considering that, the final result is not solely depended on the voting center but also the confidence map, an appearance that is coded in the unique cell still helps to estimate the target center like in Fig.6, where the target constantly undergoes out of plane rotations and non-rigid deformations.

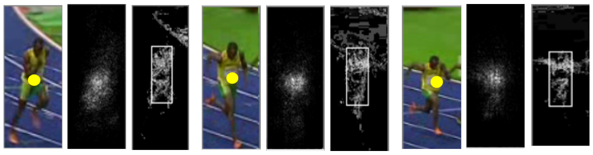


Fig. 5: Illustration of the removal of unique cells in the representation which are contaminated by background clutter. In the middle, green color in the background and on target make green no more unique to target. After updating, the green is removed from the unique target representation.

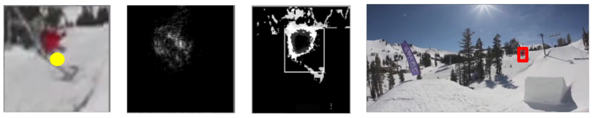


Fig. 6: Illustration of tracking with target deformations and out plane rotations. When these cases occur the center voting scheme may not have clear center and make the image center as candidate. But the confidence map provides a good approximation of the target location (red box in the last image) despite center voting errors.

4. DISCUSSION AND CONCLUSION

In this paper, we introduce a novel target tracking method that uses unique target representation and their spatial locations as codebook in feature space. We offer an efficient way to estimate the likelihood of feature to be target or background and generates a confidence map to find the range of the target while using voting codebook to estimate the target center. The experimental results using a simple color descriptor shows comparably superior performance to other state-of-the-art tracking methods.

5. REFERENCES

[1] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, "Tracking-learning-detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1409–1422, 2012.

[2] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan, "Object detection with discriminatively trained part-based models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1627–1645, 2010.

[3] Jongwoo Lim, David A Ross, Ruei-Sung Lin, and Ming-Hsuan Yang, "Incremental learning for visual tracking," in *Advances in neural information processing systems*, 2004, pp. 793–800.

[4] Shai Avidan, "Support vector tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 8, pp. 1064–1072, 2004.

[5] Sam Hare, Amir Saffari, and Philip HS Torr, "Struck: Structured output tracking with kernels," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 263–270.

[6] Yancheng Bai and Ming Tang, "Robust tracking via weakly supervised ranking svm," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1854–1861.

[7] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[8] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu, "Visual tracking with fully convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3119–3127.

[9] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3074–3082.

[10] Naiyan Wang, Siyi Li, Abhinav Gupta, and Dit-Yan Yeung, "Transferring rich feature hierarchies for robust visual tracking," *arXiv preprint arXiv:1501.04587*, 2015.

[11] Robert T Collins, Yanxi Liu, and Marius Leordeanu, "Online selection of discriminative tracking features," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1631–1643, 2005.

[12] Horst Possegger, Thomas Mauthner, and Horst Bischof, "In defense of color-based model-free tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2113–2120.

[13] Bastian Leibe, Ales Leonardis, and Bernt Schiele, "Combined object categorization and segmentation with an implicit shape model," in *Workshop on statistical learning in computer vision, ECCV*, 2004, vol. 2, p. 7.

[14] Dana H Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[15] Franklin C Crow, "Summed-area tables for texture mapping," *ACM SIGGRAPH computer graphics*, vol. 18, no. 3, pp. 207–212, 1984.

[16] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2411–2418.

[17] Jin Gao, Haibin Ling, Weiming Hu, and Junliang Xing, "Transfer learning based visual tracking with gaussian processes regression," in *European Conference on Computer Vision*. Springer, 2014, pp. 188–203.