Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 6329864, 9 pages
https://doi.org/10.1155/2017/6329864

Research Article

Convolutional Shallow Features for Performance Improvement of Histogram of Oriented Gradients in Visual Object Tracking

Suryo Adhi Wibowo, Hansoo Lee, Eun Kyeong Kim, and Sungshin Kim

Department of Electrical and Computer Engineering, Pusan National University, Busan, Republic of Korea

Correspondence should be addressed to Sungshin Kim; sskim@pusan.ac.kr

Received 24 August 2017; Accepted 6 December 2017; Published 26 December 2017

Academic Editor: Vitaly Kober

Histogram of oriented gradients (HOG) is a feature descriptor typically used for object detection. For object tracking, this feature has certain drawbacks when the target object is influenced by a change in motion or size. In this paper, the use of convolutional shallow features is proposed to improve the performance of HOG feature-based object tracking. Because the proposed method works based on a correlation filter, the response maps for each feature are summed in order to obtain the final response map. The location of the target object is then predicted based on the maximum value of the optimized final response map. Further, a model update is used to overcome the change in appearance of the target object during tracking. A performance evaluation of the proposed method is obtained by using Visual Object Tracking 2015 (VOT2015) benchmark dataset and its protocols. The results are then provided based on their accuracy-robustness (AR) rank. Furthermore, through a comparison with several state-of-the-art tracking algorithms, the proposed method was shown to achieve the highest rank in terms of accuracy and a third rank for robustness. In addition, the proposed method significantly improves the robustness of HOG-based features.

1. Introduction

In the field of computer vision research, the basic problem of visual tracking has been studied. Given an initial state, the task of visual tracking is estimating the trajectory of the target object in an image sequence. We can implement visual/object tracking, in several types of applications including surveillance systems [1], human-computer interaction (HCI) systems, unmanned aerial vehicle (UAV) systems, robotics, and three-dimensional (3D) reconstruction [2]. Further, it is difficult to implement a visual tracking algorithm that has excellent performance in terms of both accuracy and robustness. Several problems such as changes in illumination, motion, and size, as well as occlusions and camera motion, may cause tracking failures. For this reason, visual tracking has become a significant research topic in the area of computer vision.

Over the past several years, visual tracking based on online learning has made excellent progress. The key idea with this type of tracking is how to exploit the boosting classifier. Because the framework uses a boosting classifier, it can be categorized as a discriminative approach, and thus the computation time should be fast and feature extraction that can be computed rapidly is required. For example, the Adaboost classifier is implemented in an object tracking algorithm in [3]. In their research, the authors used Haar-like features. Further, a boosting classifier has also been used in the learning of multiple instances for object tracking [4]. The authors also use Haar-like features to represent the target object. The development of face tracking system based on this work was proposed by [5]. Recently, Wang et al. [6] proposed multiple instance learning based on the use of a patch, where an object is divided into many blocks. Unfortunately, extensive experiments using a benchmark dataset have not been performed on their research. Furthermore, a discriminative approach using a boosting classifier has a limitation related to the region used for searching the target object.

Based on this limitation, generative approach has been proposed. In [7], the authors proposed the use of fuzzy coding histogram features with a point representation for handling tracking failures. By adopting compressive sensing, new features representing the target object may be obtained.

Such features are called sparse coefficient vectors, which are usually combined with a particle filter for motion estimation. Several examples using a sparse coefficient vector combined with particle filter can be found in [8–10]. Such features have shown good results with regard to the problem of occlusions, although several other issues and challenging problems exist in visual tracking. Further, because these methods are based on a particle filter, issues related to the computation time still remain.

To address such issues through a generative approach, a correlation filter has been proposed. This method works based on a Fourier transform, and to improve the computation time, a fast Fourier transform (FFT) is used. Correlation-filter-based visual tracking was initially proposed by Bolme et al. [11]. They proposed a way to implement a correlation filter during visual tracking and handle changes in appearance adaptively. To do so, they used a simple linear classification to solve the problem, which is a limitation of their work. Further, a correlation filter that operates efficiently was proposed by Henriques et al. [12]. The efficiency of this filter was achieved through computations using a circulant matrix combined with a kernel. To represent the target object, histogram of oriented gradients (HOG) features were used. In [13], a HOG feature-based correlation filter is also applied. In their method, they focused on how to handle the problem of a change in size during visual tracking. Other features for correlation-filter-based visual tracking, such as adaptive color features, were used in [14], and recently a fusion between color histogram features and HOG features with distractor handling was proposed in [15]. Unfortunately, these works have a limitation in that they use only a one-dimensional (1D) feature map.

From this reason, a multiscale feature-map-based correlation filter was proposed in [16]. In their method, they used color features for simplicity. Unfortunately, although multiscale feature maps have been successfully implemented, their performance still needs to be improved. In this research, we propose how to improve a HOG-feature-based Visual Object Tracking algorithm using convolutional shallow (CS) features. Because we use both HOG and CS features, the problem is how to integrate the two owing to their different resolutions. Further, to handle the problem of a change in size of the target object, an estimation of the scale is computed after the location estimation of the target object is achieved. After the scale estimation of the target object is computed, several parameters of the proposed method need to be updated to handle the changes in appearance of the target object during tracking. Furthermore, extensive experiments were conducted using the Visual Object Tracking 2015 (VOT2015) benchmark dataset. In addition, we also conducted a comparison among the proposed method, the proposed method using only HOG features, and the proposed method using only CS features. The purpose of this comparison is to prove that the proposed method has advantages for challenging problems in Visual Object Tracking over the use of a single type of feature.

The rest of this paper is organized as follows. Section 2 discusses the proposed method. Parameter updates are described in Section 3. Further, Section 4 discusses the experiment results. Finally, Section 5 provides some concluding remarks.

2. Proposed Method

In this section, our proposed method is described. Deep learning has been rapidly developing in recent years, particularly in the area of computer vision research. In addition, one of the methods used in deep learning is the application of a convolutional neural network (CNN). The architecture of a CNN usually consists of several layers, including convolutional layers, normalization layers, and pooling layers. Moreover, convolutional layers usually consist of several layers: from a shallow layer to the deepest layer. Although the deepest layer provides the best results for image classification, in this research, we used a shallow layer because it provides more favorable information than the deepest layer for object tracking owing to the fact that the information from a shallow layer of a pretrained CNN only requires a small number operations as compared to the deepest layer. Based on this fact, the information from a shallow layer can still represent the input, which will be more useful for the case of object tracking. For detailed information regarding the architecture of the pretrained used in the present research, refer to [17]. Further, the framework of the proposed method is represented in Figure 1.

Starting with frame $(t + 1)$, the search area for the target object is defined based on an expansion of the result from frame (t). From this area, feature extraction is conducted and for this step, we use two features: HOG and CS features. For this reason, we define x_{hog} and x_{cs} for smooth results of the HOG and CS features extraction, respectively. In addition, a cosine window is used for the smoothing process.

The next step is the interpolation process. The purposes of the interpolation process are to estimate the output more accurately and to achieve an integration of multiresolution feature maps. Further, the interpolation model can be defined as follows.

$$R^n \{x^n\} (t_1, t_2) = \sum_{l=0}^{L_n - 1} x^n [l] \, k_n \left(t_1 - \frac{T_1}{L_n} l \right) k_n \left(t_2 - \frac{T_2}{L_n} l \right), \tag{1}$$

where L_n is the number of feature channels, T_1 and T_2 are related to the size of the feature map, and $t_1 \in [0, T_1)$, $t_2 \in [0, T_2)$, and $k_n(\cdot)$ represent interpolation functions. Because two features are used, we also have two interpolation models, where R^n_{hog} is the interpolation model for x_{hog} and R^n_{cs} is the interpolation model for x_{cs}.

After the interpolation models are obtained, the next step is obtaining a response map for each feature. Because the proposed method is based on a correlation filter, each response map can be obtained through a convolution between the interpolation model and the correlation filter. This computation can be expressed as follows.

$$O = F^{-1} \left(\sum_{n=1}^{N} \overline{R}^n \odot \overline{h}^{n*} \right), \tag{2}$$

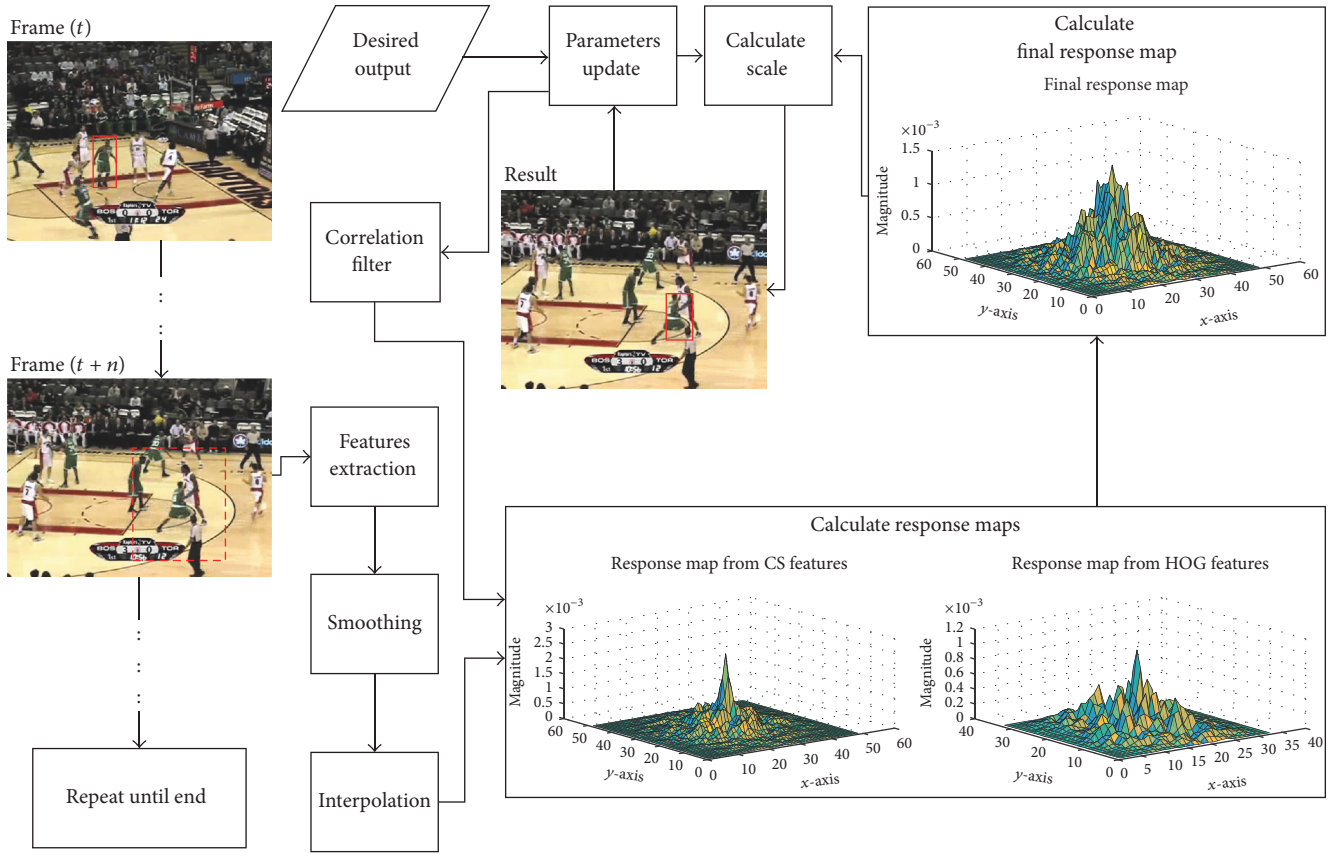

Figure 1: Framework of the proposed method.

where F^{-1}, $\overline{R}$, $\overline{h}$, $*$, and $\odot$ are an inverse Fourier transform, interpolation model in the Fourier domain, correlation filter in the Fourier domain, complex conjugate value, and elementwise multiplication, respectively. The variable $O = R \otimes h$, and as can be seen in (2), for a fast computation, a Fourier transform is used; thus, elementwise multiplication can also be applied. For this reason, (1) can be transformed into the Fourier domain as follows.

$$\overline{R^n\{x^n\}}[t_1, t_2] = \sum_{l=0}^{L_n-1} x^n[l]\, e^{-j(2\pi/L_n)l(a_1+a_2)}\overline{k}_n[a_1, a_2], \quad (3)$$

where $(a_1, a_2) \in \mathbb{Z}$.

Remembering that we use two features, based on (2), we have two types of correlation filters, where h_{cs} is the correlation filter from the CS features, and h_{hog} is the correlation filter from the HOG features. For this reason, two response maps O_{cs} and O_{hog} can be obtained. The final response map can then be calculated using

$$O_{final} = O_{cs} + O_{hog}, \quad (4)$$

where O_{cs} is the response map from the CS features, and O_{hog} is the response map from the HOG features. After O_{final} is obtained, we can estimate the location of the target object by finding the maximum value of O_{final}. As shown in Figure 2, there are three types of response maps: a response map from the proposed method using only CS features, a response map from the proposed method using only HOG features, and a response map from the proposed method. The response map O_{hog} is not sharper than the response map O_{cs}. This shape may provide an incorrect decision when we estimate the location of the target object because the maximum value of the response map has a small difference with the second-maximum value. However, when we combine the response map O_{hog} with the response map O_{cs}, the shape of O_{final} is sharper than O_{hog}. This shape makes the location estimation of the target object more robust than O_{hog}.

Further, given the location estimation of the target object, we can conduct a scale estimation of the target object because, during tracking, the scale of the target object may change, and in order to handle this problem, a scale estimation of the target object is required. Therefore, we should estimate the scale of the target object using (5) as follows.

$$O_{cscales} = F^{-1}\left(\frac{\sum_{i=1}^{I}\overline{\beta}_{t+1}^{i*} \odot \overline{b}_{t+1}^{i}}{\gamma_{t+1} + W_1}\right), \quad (5)$$

where $\overline{\beta}_{t+1}$ is the numerator in the Fourier domain, $\overline{b}_{t+1}$ is the feature sample based on the scale factor in the Fourier domain, γ_{t+1} is the denominator, and W_1 is the weight parameter controlling the regularization term in (5). The selected scale can be obtained by calculating the maximum value from $O_{cscales}$.

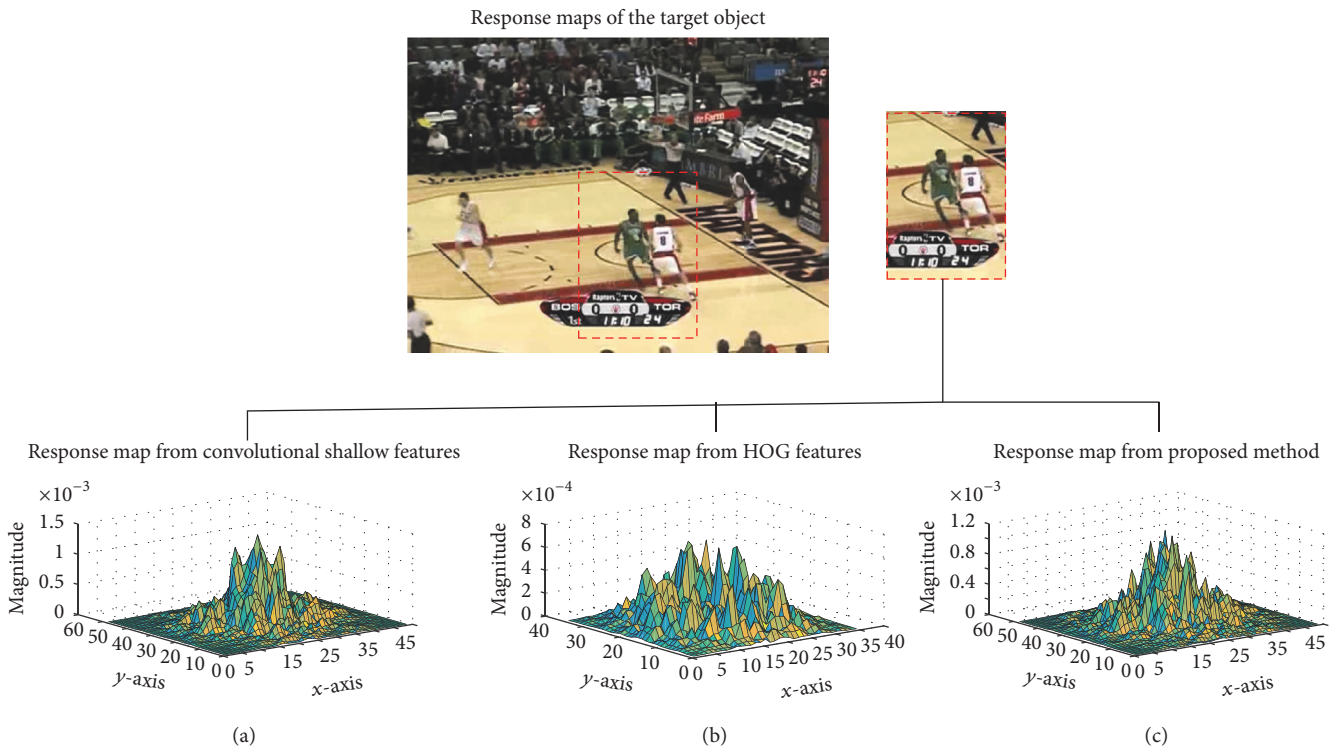

Figure 2: Response map representation: the proposed method using only CS features (a), the proposed method using only HOG features (b), and the proposed method (c).

3. Parameters Update

During tracking, the target object usually changes its appearance, which should be handled to make the tracking algorithm more robust. Because the proposed method is based on a correlation filter, the correlation filter needs to learn the desired output to handle the changes in appearance of the target object. This learning process can be achieved by solving minimization problem as follows.

$$B\{x\} = \sum_{n=1}^{N} h^n \otimes R^n\{x^n\},$$
$$\min_h \sum_{m=1}^{M} W_2^m \left\| B\{x^m\} - y^m \right\|_2^2 + \sum_{n=1}^{N} \left\| W_3 \otimes h^n \right\|_2^2, \tag{6}$$

where W_2 is the weight parameter used to control the sample pairs, M is the number of sample pairs, y is the desired output, W_3 is the weight parameter used to control the regularization term, and $\otimes$ is the convolution operator. Further, because it is used to control sample pairs, parameter W_2 should be updated for each frame. To update this parameter, we can use the following equation:

$$W_2 = \begin{cases} \left(W_2(\mathrm{id}_0) = 1-\mu\right), \left(W_2(\mathrm{id}_1) = \mu\right), & \text{if } t < 3 \\ W_2(\mathrm{id}_1) = \dfrac{W_2(\mathrm{id}_0)}{(1-\mu)}, & \text{otherwise,} \end{cases} \tag{7}$$

where μ is the learning rate, and id_0 is the index id_1 from the previous frame.

For the scale estimation, two parameters, the denominator $\overline{\gamma}_{t+1}$ and the numerator $\overline{\beta}_{t+1}$, need to be updated. Further, the numerator $\overline{\beta}_{t+1}$ can be obtained using (8) as follows:

$$\overline{\beta}_{t+1} = W_4 \overline{\beta}_0 + W_5 \left(\overline{y} \odot \overline{b}_t^*\right), \tag{8}$$

where W_4 is a weight parameter, $W_5 = 1 - W_4$, and $\overline{\beta}_0$ is the numerator from the initial frame. Furthermore, the denominator $\overline{\gamma}_{t+1}$ can be obtained using (9) as follows:

$$\overline{\gamma}_{t+1} = W_4 \overline{\gamma}_0 + W_5 \sum_i \overline{b}_t \odot \overline{b}_t^*, \tag{9}$$

where γ_0 is the denominator from the initial frame. In addition, an equation conducted in the Fourier domain is symbolized by using the upper-line.

4. Experimental Results

In this section, the experimental results are described to validate the proposed method. Using the VOT2015 benchmark dataset, the proposed method was compared with 55 state-of-the-art tracking algorithms. These 55 state-of-the-art tracking algorithms are as follows: ACT [18], amt [19], AOGTracker [19], ASMS [20], baseline [19], bdf [19], cmil [19], CMT [19], CT [21], DAT [22], DFT [19], DSST [13], dtracker [19], fct [19], fot [23], FragTrack [19], ggt [19], HMMTxD [19], HT [19],

IVT [24], kcf_mtsa [19], KCF2 [19], kcfdp [19], kcfv2 [19], L1APG [9], LGT [25], loft_lite [19], LT_FLO [26], LPMT [15], matflow [19], MCT [19], MEEM [27], MIL [4], mkcf_plus [19], muster [28], mvcft [19], ncc [29], OAB [19], OACF [19], PKLTF [19], rajssc [19], RobStruck [19], s3Tracker [19], samf [30], SCBT [31], sKCF [19], sme [19], SODLT [32], srat [19], STC [33], struck [34], sumshift [35], TGPR [36], tric [19], and zhang [19]. Further, to prove the advantages of the proposed method, a comparison among the proposed method, the proposed method using only CS features, and the proposed method using only HOG features was also conducted.

The VOT2015 benchmark dataset consists of 60 videos that have certain problems including camera motion, changes in illumination, motion changes, occlusions, and size changes, as well as videos under normal conditions (empty). In addition, to evaluate the performance in terms of the accuracy and robustness of the tracking algorithm, each video uses its protocols based on the area under curve (AUC). For more details regarding these protocols, refer to [37]. Further, the parameters used by the proposed method, N, W_1, W_3, W_4, and μ, have values of 1, 0.01, 0.001, 0.002, and 0.008, respectively. Parameter I is equal to 15, where each scale has a difference of 0.02. Parameter W_2 is equal to zero at the initial frame. Furthermore, we implemented the proposed method by using MATLAB on a 3.3 GHz i5-4590 with 4 GB of RAM.

The convolutional layer is one of many layers contained in the convolutional neural network (CNN). According to the references from [38, 39], the dot product computations of the output of the neurons with the local regions in the input (i.e., an image) are performed. The results from these computations are represented in the volume. For example, if we use the filter which has size 60×60 and the number of the filters is three, then the results of these computations in volume become $60 \times 60 \times 3$. Further, these results are categorized as the result in the first convolutional layer. For the second convolutional layer, it can be obtained with similar computation with the first convolutional layer. The differences between the second convolutional layer and the first convolutional layer are on the part of the input, the size of the filter, and the number of the filters. The input of the second convolutional layer can be as only the output of the first convolutional layer or the output of the first convolutional layer combined with the computations of the normalized layer and the pooling layer. Further, the size of the filter in the second convolutional layer is smaller than the size of the filter in the first convolutional layer. If we use the result from the first convolutional layer as the features, it can be called convolutional shallow (CS) features. In this research, we used CS features from pretrained CNN that has been proposed by the authors in [17].

Several illustrations of the comparison results among the proposed method, the proposed method using only CS features, and the proposed method using only HOG features are provided in Figure 3. In addition, as shown in Figure 4, for the camera motion problem, the proposed method achieved the highest rank in terms of accuracy and second rank in terms of robustness, whereas the proposed method using only HOG features achieved the highest rank in accuracy but ranked 43rd in robustness. The proposed method using only CS features achieves the same accuracy and robustness ranks as the proposed method. Further, the accuracy and robustness ranks of the DFT tracker are also the same as those of the proposed method. This tracker uses an image descriptor based on distributing fields, and the approach maintains the pixel value information when the objective function is smoothed. Top ranks for both accuracy and robustness were achieved by the rajssc tracker, which is based on a correlation filter and uses a block circulant-structure combined with a Gaussian space response for representing the target object.

For an empty label problem, that is, normal conditions, the proposed method achieved the highest rank for both accuracy and robustness. Meanwhile the proposed method using only HOG features achieved the first rank in accuracy and 45th rank for robustness. The proposed method using only CS features achieved the first rank in accuracy and second rank in robustness. These results indicate that combining CS and HOG features can make the tracking algorithm more robust than using CS or HOG features alone. The struck and samf trackers achieved the same rank as the proposed method, where the struck tracker is based on a kernelized structured output support vector machine, and the samf tracker is based on kernelized correlation filter that efficiently utilizes a scale adaptive method.

Further, for changes in illumination, as shown in Figure 4, both the proposed method and the proposed method using only CS features rank first in both accuracy and robustness. Meanwhile, the proposed method using only HOG features ranks first in accuracy and ninth in robustness. These results indicate that CS features are more useful than HOG features for the problem of changes in illumination. Combining CS and HOG features has no significant influence on this particular problem. However, the OACF tracker, which is based on a correlation filter combined with a red-green-blue (RGB) histogram, and also uses an adaptive scaling method, ranking first in accuracy and fourth in robustness, which are the same ranks as the rajssc tracker.

For motion changes, the proposed method achieved the highest rank for both accuracy and robustness. The proposed method using only CS features also achieved the highest rank in accuracy; however, for robustness, the method achieved second rank. Further, the proposed method using only HOG features also achieved the first rank in accuracy. Unfortunately, this method achieved a robustness ranking 44th. Based on this, the proposed method shows a superior performance than using only CS features or only HOG features. Furthermore, combining HOG and CS features proves that CS features may improve the robustness performance significantly compared to using only HOG features. The rajssc tracker has the same rank as the proposed method. For the s3Tracker tracking algorithm, which is based on an RGB histogram to represent the target object and also uses an aspect ratio selection. Moreover, an accuracy ranking second and a robustness ranking third were achieved by the LPMT tracker, which is based on a correlation filter with distractor handling.

The next problem addressed is occlusions, where the target object is fully or partially occluded. For this problem, the proposed method achieved the highest rank in terms of

Figure 3: Comparison of the results from several sequences on the VOT2015 benchmark dataset. The blue rectangle represents the result from the proposed method, the red rectangle represents the result from the proposed method using only CS features, the green rectangle represents the result from the proposed method using only HOG features, and the yellow rectangle represents the ground-truth.

accuracy. Unfortunately, for the robustness, it only achieved a 16th ranking, which is much lower than when using only CS features, where the proposed method achieved the highest rank in accuracy and a robustness second ranking. Based on this evidence, the proposed method using only CS features is more robust than both the proposed method and the proposed method using only HOG features. This is because when the target is occluded and the correlation filter is updated, the response from the CS features is more similar to the target object than the others. For this reason, for the occlusion problem, the CS features are more robust than both the proposed method and the proposed method using only HOG features. On the other hand, the proposed method using only HOG features ranks first in terms of accuracy and 27th in robustness. Top ranks in both accuracy and robustness were achieved by the rajssc and sme trackers, the latter of which is a tracking algorithm that operates based on a score function for selecting the candidate from multiple experts.

The final problem defined in the VOT2015 benchmark dataset is a change in size. For this problem, the proposed method shows better results than the others, achieving the highest rank in both accuracy and robustness. Meanwhile, the proposed method using only CS features achieved the highest rank in accuracy and a robustness 11th ranking. Further, the proposed method using only HOG features achieved the highest rank in accuracy and a robustness 50th ranking. These results show that a combination of CS and HOG features may increase the robustness significantly. Other

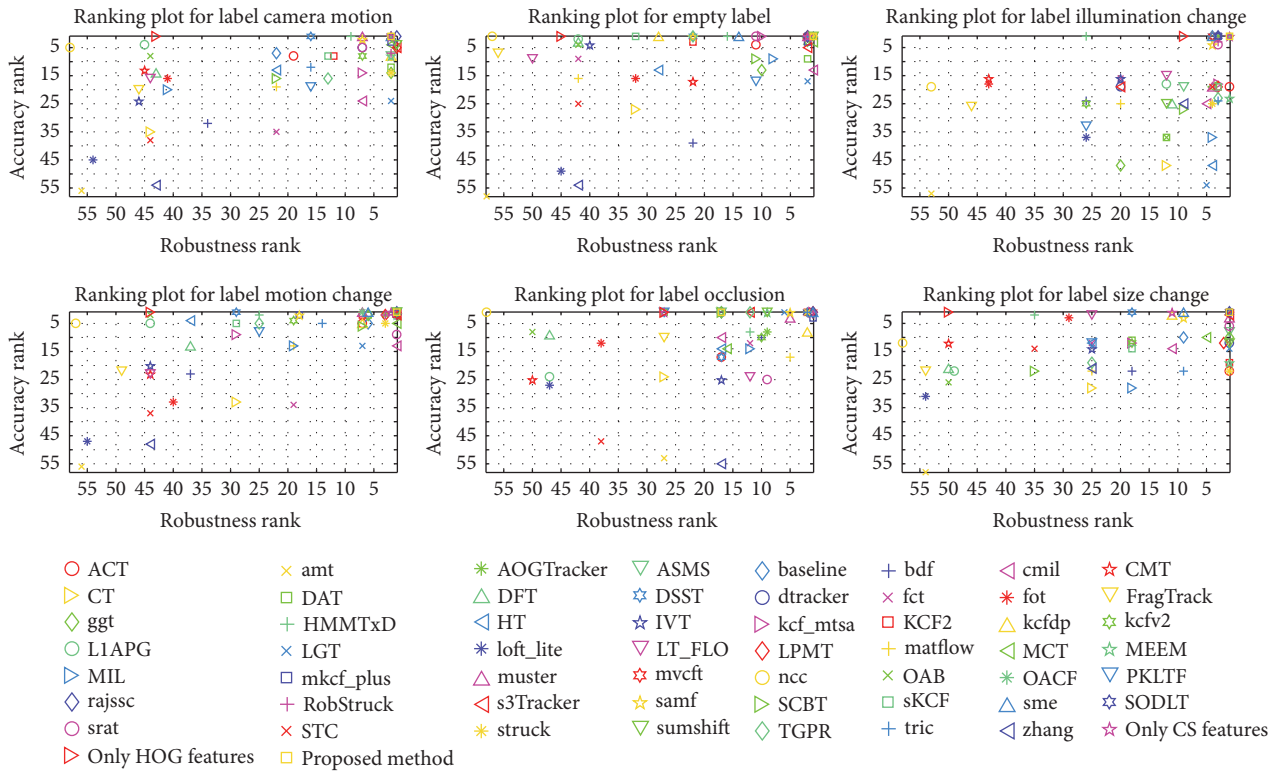

FIGURE 4: AR ranking plot for camera motion problems, normal conditions (empty), changes in illumination, changes in motion, occlusions, and changes in size. The proposed method was compared with the proposed method using only HOG features, the proposed method using only CS features, and other state-of-the-art tracking algorithms.

state-of-the-art tracking algorithms, s3Tracker and muster, achieved second and third ranks for accuracy, respectively, and the highest rank for robustness. These results are shown in Figure 4.

Finally, after AR ranks from several problems including camera motion, changes in illumination, motion changes, occlusions, and size changes, as well as videos under normal conditions (empty) are obtained, we can summarize the results and then ranking plot for experiment baseline (pooled) can be obtained. This ranking plot is obtained by concatenating the results from all sequences and creating a single rank list. As shown in Figure 5, the proposed method achieved the highest rank in accuracy and second rank in robustness. Meanwhile the proposed method using only CS features achieved the highest rank in accuracy and a robustness seventh ranking. The proposed method using only HOG features achieved the highest rank for accuracy but unfortunately achieved 45th rank for robustness. This reinforces the idea that combining CS and HOG features makes the tracking algorithm more robust and is very useful when developing a tracking algorithm. However, for the sme and sumshift trackers, both, achieved the highest rank for accuracy and ranked fourth in the robustness. Finally, the proposed method achieved a computation time of about 15 fps.

5. Conclusion

This paper described how to improve the performance of HOG features-based Visual Object Tracking algorithm. The proposed method combines a response map between the HOG and CS features. The CS features are computed from a shallow layer of a pretrained CNN with the input. In addition, to handle the differences in resolution, an interpolation approach is used. Further, experiments were conducted using the VOT2015 benchmark dataset, which consists extensively of 60 different videos. The results indicated that the proposed method significantly improves the robustness performance of a HOG feature-based approach. In addition, based on a comparison with many other state-of-the-art tracking algorithms, the proposed method achieved the highest rank in terms of accuracy and a third rank for robustness.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by BK21PLUS, Creative Human Resource Development Program for IT Convergence.

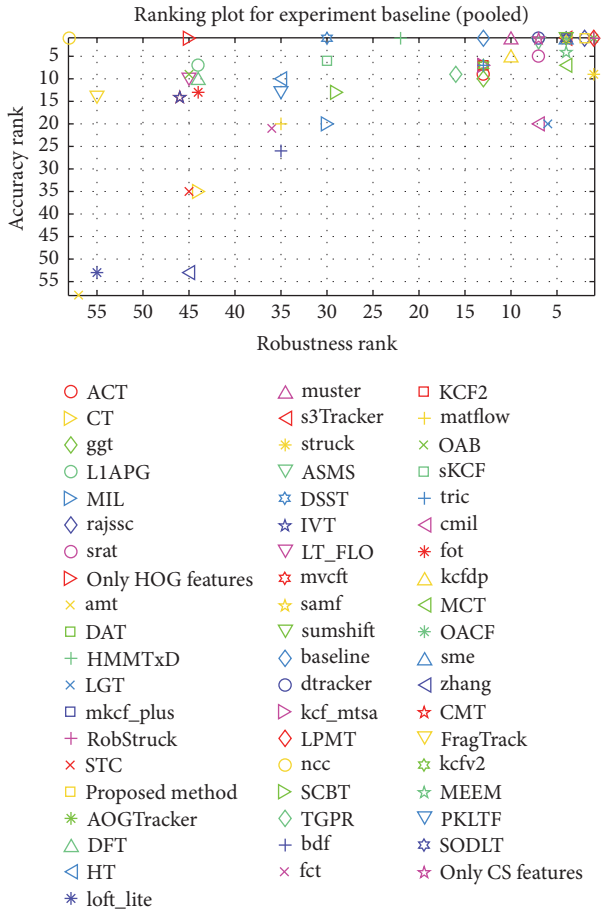

FIGURE 5: AR ranking plot baseline (pooled) for the proposed method, the proposed method using only HOG features, the proposed method using only CS features, and other state-of-the-art tracking algorithms.

References

[1] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: an experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, July 2014.

[2] S. A. Wibowo and S. Kim, "Three-dimensional face point cloud smoothing based on modified anisotropic diffusion method," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 14, no. 2, pp. 84–90, 2014.

[3] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proceedings of the British Machine Vision Conference (BMVC '06)*, vol. 1, pp. 6.1-6.10, BMVA Press, Edinburgh, UK, September 2006.

[4] 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.

[5] S. A. Wibowo, H. Lee, E. K. Kim, and S. Kim, "Performance analysis of online weighted multiple instance learning for single face tracking at outdoor environment," in *Proceedings of The 16th International Symposium on Advanced Intelligent Systems*, pp. 1257–1264, 2015.

[6] Z. Wang, L. Wang, and H. Zhang, "Patch based multiple instance learning algorithm for object tracking," *Computational Intelligence and Neuroscience*, vol. 2017, pp. 1–7, 2017.

[7] S. A. Wibowo, H. Lee, E. K. Kim, T. Kwon, and S. Kim, "Tracking failures detection and correction for face tracking by detection approach based on fuzzy coding histogram and point representation," in *Proceedings of the International Conference on Fuzzy Theory and Its Applications (iFUZZY '15)*, pp. 34–39, Taiwan, November 2015.

[8] X. Mei, H. Ling, Y. Wu, E. P. Blasch, and L. Bai, "Efficient minimum error bounded particle resampling L1 tracker with occlusion detection," *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2661–2675, 2013.

[9] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1830–1837, Providence, RI, USA, June 2012.

[10] S. A. Wibowo, H. Lee, E. K. Kim, and S. Kim, "Fast Generative Approach Based on Sparse Representation for Visual Tracking," in *Proceedings of the 8th Joint International Conference on Soft Computing and Intelligent Systems and 17th International Symposium on Advanced Intelligent Systems (SCIS-ISIS '16)*, pp. 778–783, Japan, August 2016.

[11] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1401–1409, San Francisco, Calif, USA, 2010.

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

[13] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proceedings of the 25th British Machine Vision Conference (BMVC '14)*, pp. 1–11, Linköping, Sweden, September 2014.

[14] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1090–1097, IEEE, Columbus, Ohio, USA, June 2014.

[15] S. A. Wibowo, H. Lee, E. K. Kim, and S. Kim, "Visual tracking based on complementary learners with distractor handling," *Mathematical Problems in Engineering*, vol. 2017, pp. 1–13, 2017.

[16] S. A. Wibowo, H. Lee, E. K. Kim, and S. Kim, "Multi-scale color features based on correlation filter for visual tracking," in *Proceedings of the 1st International Conference on Signals and Systems*, pp. 272–277, 2017.

[17] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: delving deep into convolutional nets," in *Proceedings of the 25th British Machine Vision Conference (BMVC '14)*, pp. 1–11, Nottingham, UK, September 2014.

[18] M. Felsberg, "Enhanced distribution field tracking using channel representations," in *Proceedings of the IEEE International Conference on Computer Vision Workshop (ICCVW '13)*, pp. 121–128, 2013.

[19] M. Kristan, J. Matas, A. Leonardis et al., "The Visual Object Tracking VOT2015 Challenge Results," in *Proceedings of The IEEE International Conference on Computer Vision Workshop (ICCVW '15)*, pp. 1–23, 2015.

[20] T. Vojir, J. Noskova, and J. Matas, "Robust scale-adaptive mean-shift for tracking," *Pattern Recognition Letters*, vol. 49, pp. 250–258, 2014.

[21] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proceedings of the 12th European Conference on Computer Vision (ECCV '12)*, pp. 864–877, 2012.

[22] H. Possegger, T. Mauthner, and H. Bischof, "In defense of color-based model-free tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 2113–2120, Boston, Mass, USA, June 2015.

[23] T. Vojir and J. Matas, "Robustifying the flock of trackers," in *Proceedings of the Computer Vision Winter Workshop*, pp. 91–97, 2011.

[24] 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. 1–3, pp. 125–141, 2008.

[25] L. Čehovin, M. Kristan, and A. Leonardis, "Robust visual tracking using an adaptive coupled-layer visual model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 941–953, 2013.

[26] K. Lebeda, S. Hadfield, J. Matas, and R. Bowden, "Long-term tracking through failure cases," in *Proceedings of the 14th IEEE International Conference on Computer Vision Workshops (ICCVW '13)*, pp. 153–160, Sydney, Australia, December 2013.

[27] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: robust visual tracking via multiple experts using entropy," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Patern Recognition (CVPR '14)*, pp. 188–203, 2014.

[28] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "MUlti-Store Tracker (MUSTer): a cognitive psychology inspired approach to object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 749–755, Boston, Mass, USA, June 2015.

[29] K. Briechle and U. D. Hanebeck, "Template matching using fast normalized cross correlation," *SPIE*, vol. 4387, pp. 95–102, March 2001.

[30] L. Yang and Z. Jianke, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proceedings of the European Conference on Computer Vision Workshop (ECCV '14)*, pp. 254–265, 2014.

[31] S. Moujtahid, S. Duffner, and A. Baskurt, "Classifying global scene context for on-line multiple tracker," in *Proceedings of the British Machine Vision Conference (BMVC '15)*, pp. 1–12, 2015.

[32] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung, "Transferring rich feature hierarchies for robust visual tracking," *CoRR*, vol. abs/1501.04587, pp. 1–9, 2015.

[33] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *Proceedings of the European Conference on Computer Vision Workshop (ECCV '14)*, pp. 127–141, 2014.

[34] S. Hare, S. Golodetz, A. Saffari et al., "Struck: structured output tracking with kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2096–2109, October 2016.

[35] J.-Y. Lee and W. Yu, "Visual tracking by partition-based histogram backprojection and maximum support criteria," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '11)*, pp. 749–758, IEEE, Phuket, Thailand, December 2011.

[36] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfering learning based visual tracking with gaussian processes regression," in *Proceedings of the European Conference on Computer Vision Workshop (ECCV '14)*, pp. 188–203, 2014.

[37] M. Kristan, J. Matas, A. Leonardis et al., "A novel performance evaluation methodology for single-target trackers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2137–2155, 2016.

[38] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, Lake Tahoe, Nev, USA, December 2012.

[39] cs231n.github.io/convolutional-networks/.