applied sciences

Article

Visual Tracking of Small Unmanned Aerial Vehicles Based on Object Proposal Voting

Jin Hong and Junseok Kwon *

School of Computer Science and Engineering, Chung-Ang University, Seoul 156-756, Korea; yap030@naver.com
* Correspondence: jskwon@cau.ac.kr; Tel.: +82-10-8693-7455

Abstract: In this paper, we propose a novel visual tracking method for unmanned aerial vehicles (UAVs) in aerial scenery. To track the UAVs robustly, we present a new object proposal method that can accurately determine the object regions that are likely to exist. The proposed object proposal method is robust to small objects and severe background clutter. For this, we vote on candidate areas of the object and increase or decrease the weight of the area accordingly. Thus, the method can accurately propose the object areas that can be used to track small-sized UAVs with the assumption that their motion is smooth over time. Experimental results verify that UAVs are accurately tracked even when they are very small and the background is complex. The proposed method qualitatively and quantitatively delivers state-of-the-art performance in comparison with conventional object proposal-based methods.

Keywords: unmanned aerial vehicles; object tracking; object proposal voting

Citation: Hong, J.; Kwon, J. Visual Tracking of Small Unmanned Aerial Vehicles Based on Object Proposal Voting. *Appl. Sci.* **2021**, *11*, 953. https://doi.org/10.3390/app11030953

Academic Editor: Rubén Usamentiaga

Received: 4 December 2020
Accepted: 19 January 2021
Published: 21 January 2021

1. Introduction

The objective of object tracking is to estimate the exact location and size of an object of interest in an image over time, where the object can be more accurately tracked if only the areas with a high probability of object existence are considered in the image. Subsequently, conventional visual trackers have employed object proposal methods [1]. The goal of object proposal was to determine if object regions exist given an image and, if they exist, to estimate positions and scales of these object regions. To localize the object regions, conventional object proposal methods [2] search for a set of bounding boxes of object candidate regions instead of examining all positions in an image. However, these object proposal methods are very sensitive to complex background clutter in the image and cannot accurately propose areas of small-sized objects. Moreover, if the objects and their background share similar appearances, conventional approaches frequently propose the background as the object area. Consequently, the visual tracking accuracy reduces owing to inaccurate object proposals.

To overcome this issue, we propose a visual tracking method that can track small unmanned aerial vehicles (UAVs) [3–11] using a new object proposal algorithm. To make object proposals robust to various factors that interfere with visual tracking, we conduct object proposal voting (Figure 1b) where areas with high probabilities of objects being presented have more votes, whereas areas where objects are unlikely to exist have fewer votes. Based on these proposal results, a small UAV can be accurately tracked under various challenging visual tracking environments. For example, as shown in Figure 1a, the target object marked with a red circle is very small and its appearance is very similar to that of its background. Thus, conventional methods easily propose background areas as object areas, as shown in the green boxes of Figure 1c. In contrast, our method generates the object proposal voting image in Figure 1b, where object regions have more votes, which produce areas in the image that are brighter than the background regions. By thresholding the object proposal voting image, our method proposes a single object area, as shown in the blue box in Figure 1d. Based on the estimated object region, our method can accurately tack the target.

The contributions of our method are summarized as follows:

- We propose a novel object proposal algorithm called object proposal voting (OPV). As a result that voting strategies are typically insensitive to noise, the proposed OPV is robust to background clutter and can detect small-sized objects.
- We present a visual tracking system based on the proposed OPV to track small-sized UAVs, such as drones in real-world environments.

The remainder of this paper is organized as follows. We relate conventional object proposal methods with the proposed method in Section 2. Section 3 describes object proposal baselines, while Section 4 proposes a novel OPV method. In Section 5, we present a new visual tracking method based on the proposed OPV. Section 6 describes experimental results and Section 7 concludes our paper.

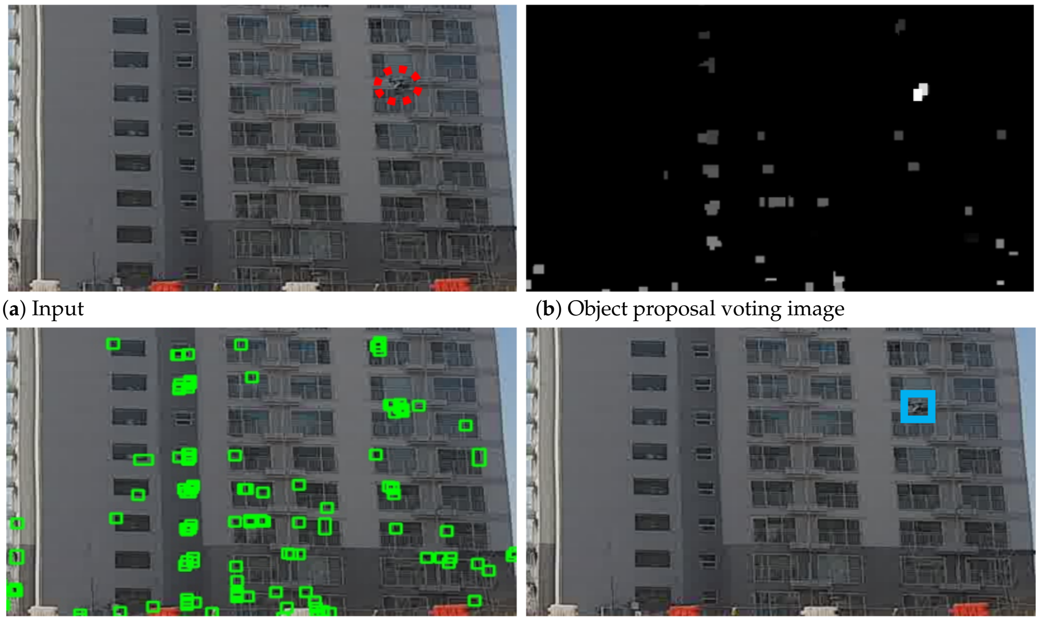

(**a**) Input
(**b**) Object proposal voting image
(**c**) Conventional result
(**d**) Our result

Figure 1. Basic concept of the proposed method.

2. Related Work

2.1. Object Proposal Methods

To propose object-like areas, BING [2] assumed that objects could be represented by closed boundaries. EdgeBox [12] extracted edge information in an image and attempted to find closed edges. Rantalankila et al. [13] presented a method that generates object segmentation proposals. Hosang et al. [14] summarized the advantages of several object proposal methods. Kwon and Lee [1] presented a new boundary extraction method for object proposals. Recently, deep learning-based object proposal approaches have been actively studied [15]. For example, Pirinen and Sminchisescu [16] used a deep reinforcement learning method to implement region proposal networks for object proposals. In contrast to these methods, our method is robust to severe background clutter because the method uses a voting scheme. In addition, the proposed voting algorithm can be integrated into existing object proposal frameworks including the methods described earlier.

2.2. Object Proposal-Based Visual Trackers

Hua et al. [17] introduced a visual tracking method based on object proposals. However, no explicit mechanism for ignoring background clutter exists. Liang et al. [18] adopted BING object proposal methods for visual tracking. Kwon and Lee [1] used EdgeField object proposal algorithms for data association between consecutive frames. However, the performance of object proposals in these cases was very sensitive to the background clutter. In contrast, our method overcomes this issue by using a novel object proposal algorithm.

2.3. Generic Visual Trackers

SCM [19] combined sparsity-based discriminative classifiers with sparsity-based generative models to cover appearance changes in target objects. Struck [20] presented a visual tracking method based on a structured output support vector machine, which enables to couple label prediction with object position estimation. TLD [21] transformed visual tracking problems into a combination of tracking, learning, and detection problems. Tracking aims to associate similar detection results across consecutive frames, whereas detection determines target areas. Learning updates detectors to reduce errors. ASLA [22] represented target appearances using a novel alignment-pooling method, in which partial and spatial structures of target appearances are exploited to deal with occlusion and geometric deformation. CXT [23] exploited distracters and supporters using sequential randomized forests. Distracters make visual trackers to drift into background regions, thus they should be avoided during visual tracking. Supporters help visual trackers to find foreground regions, thus they can improve visual tracking accuracy. VTS [24] sampled multiple visual trackers using Markov Chain Monte Carlo methods by proposing different appearance, motion, state, and observation models, in which these trackers run interactively to track target objects in real-world scenarios. VTD [25] decomposed conventional visual tracking models into different combinations of basic appearance and motion models, in which each combination deals with a particular appearance and motion changes in target objects. CSK [26] employed circulant matrices for visual tracking, which enables fast learning and detection. LSK [27] selected representative bases from a static sparse dictionary for visual tracking, which is updated adaptively over time. LOT [28] proposed a visual tracking method based on a probabilistic model to handle different variations especially in deformable objects. SiamDW [29] used deeper and wider neural networks for visual tracking, in which receptive fields and network strides are adaptively controlled via residual modules. SiamRPN++ [30] adopted ResNet architectures with layer-wise and depth-wise feature aggravation for multi-level feature fusion. SINTop [31] required no appearance model updating for visual tracking and simply found the best patch, which is mostly similar to the initial target patch, using a trained matching function. DAT [32] presented attention-based visual trackers using a reciprocative learning algorithm, which focus more on temporally representative features. TADT [33] introduced regression and ranking losses with Siamese networks to generate target-aware and scale-sensitive features for visual tracking. ECO-HC [34] reduced the number of network parameters and computational complexity in visual tracking by utilizing factorized convolution operators and compact generative models. COT [35] developed continuous convolution filters defined in in a continuous spatial domain, which can fuse multi-level deep features. In contrast, our method uses a simple object proposal voting scheme without complex deep neural architectures for visual tracking.

3. Object Proposal Baseline

The object is characterized by a closed boundary when its edge component is extracted. Subsequently, if the closed curve can be accurately found in the image, candidate object regions can be determined precisely. Therefore, it is necessary to find a closed curve more accurately to propose an object region candidate group. Hence, in this study, EdgeField [1] is adopted, which smoothens the energy distribution of the objective function by blurring the image. This blurring process does not cause images to lose their details because we change a single channel-based image (i.e., gray image) to multiple channel-based images (i.e., thresholded binary images at multiple intensity levels), and each channel preserves its own information even after blurring. To create multiple channels, we set the threshold of an image at the l-th intensity level at time t, I_t^l, which yields multiple thresholded images, as follows:

$$I_t^l(\cdot) = \begin{cases} 255 & \text{if } I_t(\cdot) \geq l \\ 0 & \text{if otherwise} \end{cases}. \tag{1}$$

Subsequently, we apply the Gaussian blurring to each thresholded image, as follows:

$$I_b^{l,t} = I_t^l \circledast G_\sigma, \tag{2}$$

where G_σ denotes a Gaussian kernel with variance σ and $\circledast$ is a convolution operation. Note that blurring smoothens the energy landscape for edge extraction, which indicates a geometric shape of the objective function for edge extraction. As the last step, the closed boundaries in $I_b^{l,t}$ are found by conventional contour detection functions *Contour* (e.g., [36]).

$$\mathcal{O}^{l,t} = \left\{O_j^{l,t}\right\}_{j=1}^{|\mathcal{O}^{l,t}|} \sim Contour\left(I_b^{l,t}\right), \tag{3}$$

where $\mathcal{O}^{l,t}$ denotes a set of object proposals obtained for an input image $I_b^{l,t}$ and $|\mathcal{O}^{l,t}|$ denotes the number of object proposals. The contour detection functions *Contour* in (3) produce binary images using border-following algorithms, in which connected pixels for object regions (background regions or holes) have values of 1 (0). From this binary image, the function extracts borders that split 1 and 0 regions, which yield border description images. Subsequently, each border in border description images are converted into bounding boxes that compactly fit the corresponding borders.

4. Object Proposal Voting

The object proposal algorithm presented in the previous section has difficulty in finding appropriate object regions, if severe background clutter exists. Moreover, small-sized object regions are rarely proposed in this algorithm. To overcome these problems, a new object proposal voting algorithm is proposed.

Voting for a new object proposal consists of three steps:

- Step 1: The object candidate area more voted by the object proposal baseline (OPB) increases the weight. The weight is initialized to zero for all areas, and then, a value of 10 is assigned to a specific area whenever that area receives one vote.
- Step 2: All the weights by voting are computed and are expressed in the form of an image with values ranging from 0 to 255. In this case, if the weight exceeds 255, it is expressed as 255. This is called the voting image. Figure 1b shows the voting image.
- Step 3: An object proposal using the OPB is performed on the voting image again, as shown in Figure 1d. As a result that the area with many votes is bright and the area with less votes is dark, as shown in Figure 1b, object areas can be proposed more accurately and robustly against the background.

The procedure outlined earlier can be formulated as follows. The vote grids $\{g_k\}_{k=1}^{|\mathcal{V}|}$ are determined by dividing an input image into a regular grid, where each grid has the 5 × 5 size of pixels in all the experiments. A vote histogram $\mathcal{V}$ has multiples bins $\{b_k\}_{k=1}^{|\mathcal{V}|}$, in which each bin corresponds to each grid. Then, $b_k = [g_k; w_k]$ with $g_k \in \mathbb{R}^2$ as the k-th vote grid and $w_k \in \mathbb{R}$ as the corresponding vote weight. If the center position of the object candidate area, which is obtained by the OPB, belongs to a particular bin g_k, we increase the corresponding vote weight w_k by an amount of a predefined value c.

$$w_k \leftarrow \max[255, w_k + c], \tag{4}$$

where c is set to 10 for all the experiments and all vote weights $\{w_k\}_{k=1}^{|\mathcal{V}|}$ are initialized to zero. The voting image I_v is constructed by assigning each vote weight for each bin as a value of the corresponding image grid, in which the value cannot be larger than 255 due to (4).

$$I_v \sim I(g_k) = w_k, \text{ for } k = 1, \cdots, |\mathcal{V}|, \tag{5}$$

where $I(g_k)$ assigns the value of w_k to grid g_k of an image. As the last step, the closed boundaries in I_v are found using *Contour* in (3):

$$\mathcal{O}_v = \{O_j\}_{j=1}^{|\mathcal{O}_v|} \sim Contour(I_v). \tag{6}$$

5. Visual Tracking Based on Object Proposal Voting

Object tracking is performed for the object candidate areas proposed by the object proposal voting method. It is formulated based on the assumption that the position of an object in a frame changes minimally in the adjacent frame. Thus, if the object position in the current frame is known, the area closest to the object position in the current frame is selected from candidate areas for object proposal in the next frame. In addition to tracking the object by the similarity of the object positions in adjacent frames, more accurate results can be obtained by assessing the similarity of the object appearances in adjacent frames. For this purpose, we adopt a Markov chain Monte Carlo data association algorithm [37] that can be run in realtime. We associate similar object proposals in terms of positions and appearances across consecutive frames:

$$p(O_i, O_j) = exp(-d(O_i, O_j)), \tag{7}$$

where O_i and O_j are the i-th and j-th object proposals in the current and next frames, respectively. The dissimilarity function d is designed as follows:

$$\begin{aligned} &d(O_i, O_j) = \\ &\left|g(O_i) - g(O_j)\right|_2 + \left|a(O_i) - a(O_j)\right|_2 + \left|w(O_i) - w(O_j)\right|_2, \end{aligned} \tag{8}$$

where the first and second terms measure the $l2$-norm distance between two object proposals in terms of their positions and appearances, respectively. In (8), $g(\cdot)$ returns (x, y)-center positions of bounding boxes, which are obtained by the proposed object proposal voting method. $a(\cdot)$ returns appearance feature vectors of image patches described by the aforementioned bounding boxes. We obtain appearance feature vectors from the 14-th feature map of the VGG-19 network [38], which is pre-trained using the ImageNet dataset [39]. Each layer contains different features with various sizes, in which early convolutional layers typically extract local (low level) features and later layers exhibit global (high-level) features. Based on this observation, we use the 14-th feature map of the VGG-19 network, because it can contain both local and global properties. $w(\cdot)$ returns voting weights computer by (4), in which voting weights have high values if we vote the corresponding regions many times. As a result that identical objects have similar voting weights across consecutive frames, we also measure the $l2$-norm distance between voting weights using the third term in (8). For visual tracking, we choose the best pair of object proposals, which maximizes $p(O_i, O_j)$ in (7):

$$(\hat{O}_i, \hat{O}_j) = \underset{i,j}{\operatorname{argmax}}\ p(O_i, O_j). \tag{9}$$

Algorithm 1 illustrates a whole pipeline of the proposed visual method based on our OPV. Table 1 summarizes notations used in this paper.

Algorithm 1 Visual Tracking Based on OPV.

```
Input: $\mathcal{O}_v^{t-1}$
Output: $\mathcal{O}_v^t$ and $(\hat{O}_i, \hat{O}_j)$
Obtain $\mathcal{O}_v^t$ using (6) at a current frame $t$.
for $i = 1$ to $|\mathcal{O}_v^{t-1}|$ do
    Sample $O_i$ from $\mathcal{O}_v^{t-1}$.
    for $j = 1$ to $|\mathcal{O}_v^t|$ do
        Sample $O_j$ from $\mathcal{O}_v^t$.
        Compute $d(O_i, O_j)$ using (8).
    end for
end for
Find $(\hat{O}_i, \hat{O}_j)$ using (9).
```

Table 1. Description on notations.

Notation	Description
I_t^l	Thresholded image at the l-th intensity level at time t
$I_b^{l,t}$	Gaussian blurred image of I_t^l
$O_j^{l,t}$	The j-th closed boundary of $I_b^{l,t}$
$\mathcal{O}^{l,t}$	A set of all closed boundaries in $I_b^{l,t}$
g_k	The k-th vote grid
b_k	The k-th histogram bin
w_k	Vote weight in b_k
$\mathcal{V}$	Vote histogram
$\lvert\mathcal{V}\rvert$	The number of histogram bins
I_v	Voting image
$\mathcal{O}_v$	A set of all closed boundaries in I_v

6. Experiment

To evaluate the proposed method (**OPV**: object proposal voting), we used our six UAV datasets and a standard benchmark dataset (OTB) [40]. We compared the OPV method with state-of-the-art deep learning-based visual trackers, SiamDW [29], SiamRPN++ [30], SIN-Top [31], DAT [32], TADT [33], ECO-HC [34], and COT [35]. In addition, we also compared the OPV method with non-learning-based visual trackers, including SCM [19], Struck [20], TLD [21], ASLA [22], CXT [23], VTS [24], VTD [25], CSK [26], LSK [27], and LOT [28].

We adopted three evaluation metrics, which are precision plot, success plot, and area under the curve (AUC) [40]. To illustrate the precision plot, we first compute the distance between center locations of predicted and ground truth bounding boxes. Subsequently, we determine the number of frames, in which the distance is less than a specific threshold. To obtain the success plot, we first compute the overlap ratio of the predicted and ground truth bounding boxes. Subsequently, we determine the number of frames, in which the overlap ratio is greater than a specific threshold. The AUC metric indicates the entire area under the success plot.

6.1. Ablation Study

We verified the effectiveness of the proposed object proposal voting in performing visual tracking. For this experiment, we adjust the parameter c in (4); a small value of c causes the voting process to make a smaller contribution to the object proposal results. Table 2 shows the AUC of the proposed OPV using the OTB dataset, which verifies that the proposed voting process considerably enhances the object proposal accuracy, and thus improves the visual tracking. We also examined the contributions of the position, appearance, and vote weight terms in (8) to visual tracking using the OTB dataset. For this experiment, we used different combinations of position, appearance, and vote weight terms. As shown in Table 3, using the voting weight to calculate the distance enhances the visual tracking performance.

Table 2. Visual tracking performance of the proposed OPV according to values of c in (8).

	1	**5**	**10**	**15**	**20**
AUC	0.631	0.645	0.706	0.707	0.706

Table 3. Visual tracking performance of the proposed object proposal voting (OPV) according to different settings for the terms in (8). OPV_g, OPV_{g+a}, and OPV_{g+a+w} denote the proposed visual tracker using position, position+appearance, and position+appearance+vote weight terms, respectively.

	$\boldsymbol{OPV_g}$	$\boldsymbol{OPV_{g+a}}$	$\boldsymbol{OPV_{g+a+w}}$
AUC	0.674	0.683	0.706

6.2. Quantitative Comparisons

Figure 2 shows a quantitative comparison of the proposed OPV with state-of-the-art non-deep learning-based visual trackers. Our visual tracker considerably outperforms other methods, of which performance mainly comes from the effectiveness of the proposed OPV. Figure 3 shows a quantitative comparison of the proposed OPV with state-of-the-art deep learning-based visual trackers. Our visual tracker is comparable to other methods; our method is simple owing to complex deep neural networks not being employed.

Table 4 compared the OPV with other visual trackers using our UAV datasets, which verified that the OPV outperforms other trackers if either small targets or severe backgrounds exist. Other trackers easily missed the targets.

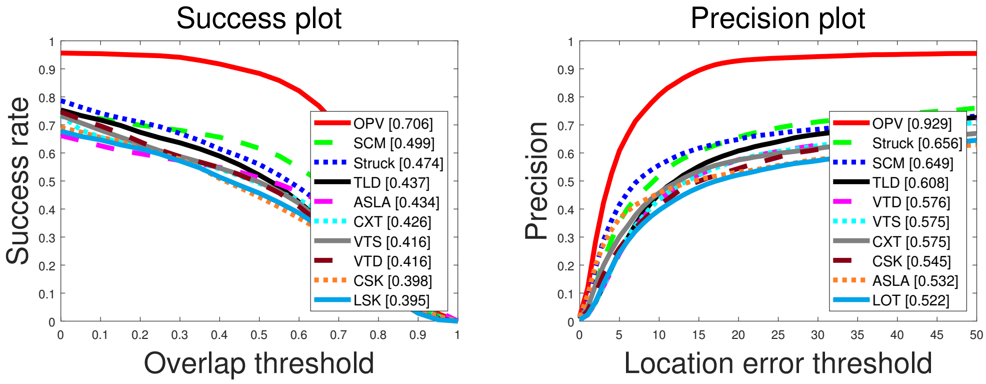


Figure 2. Quantitative comparison on non-deep learning-based methods using the OTB dataset. We compared the proposed method (OPV) with Sparsity-based Collaborative Model (SCM), STRUCtured output tracking with Kernels (Struck), Tracking-Learning-Detection (TLD), Adaptive Structural Local sparse Appearance (ASLA), ConteXt Tracker (CXT), Visual Tracker Sampler (VTS), visual trackibg decomposition (VTD), Circulant Structure of tracking-by-detection with Kernels (CSK), Local Sparse appearance model and K-selection (LSK), and Locally Orderless Tracking (LOT).

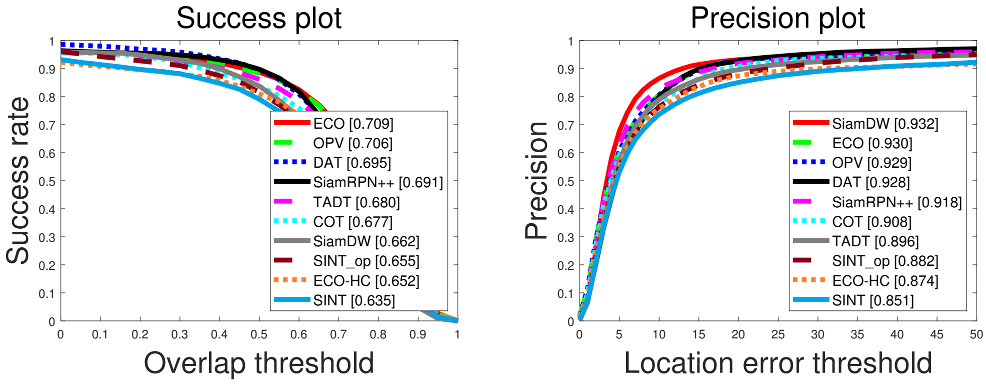


Figure 3. Quantitative comparison on deep learning-based methods using the OTB dataset. We compared the proposed method (OPV) with SIAMese network Deeper and Wider (SiamDW), SIAMese Region Proposal Network (SiamRPN++), Siamese INstance search Tracker (SINT), Deep Attentive Tracking (DAT), Target-Aware Deep Tracking (TADT), Efficient Convolution Operators (ECO), and Continuous Convolution Operator Tracker (C-COT).

Table 4. Quantitative comparison using our unmanned aerial vehicle (UAV) datasets. The best results are indicated in bold type.

	SiamRPN++	DAT	SiamDW	ECO	OPV
AUC	0.638	0.651	0.649	0.657	**0.693**

Table 5 compared computational costs of several visual tracking methods in terms of FPS. The proposed OPV shows real-time performance and is considerably faster than other deep learning-based visual trackers. This property mainly stems from the simplicity of the proposed visual tracker, in which our method does not require complex deep neural network architectures.

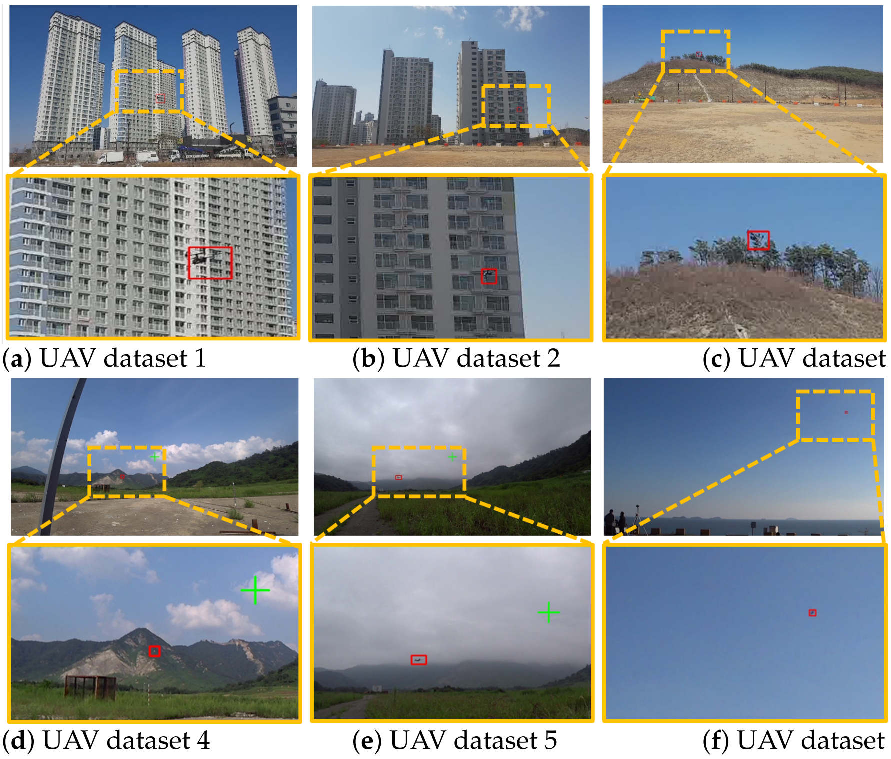
Table 5. Computational costs of OPV in terms of frames per seconds (FPS).

	ECI	C-COT	SINT	OPV
FPS	7	1	5	87

6.3. Qualitative Comparisons

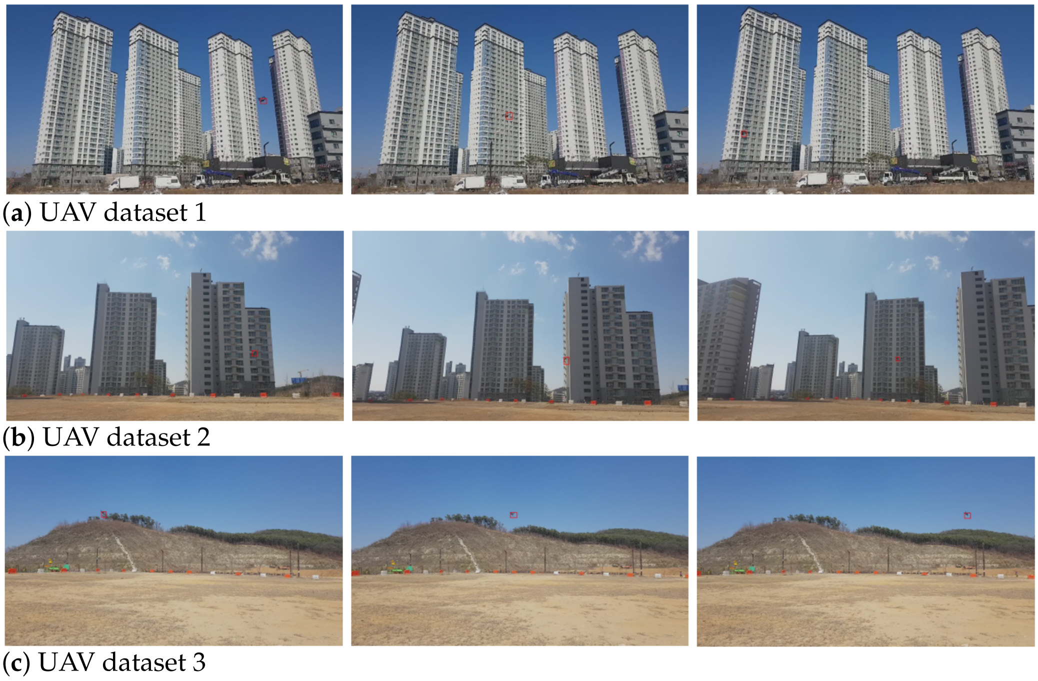
Figure 4 presents sample results of the proposed OPV using our UAV datasets, which contain very small UAVs under various visual tracking environments (e.g., severe background clutter (UAV datasets 1, 2, 3, 4, and 5), large scale changes (UAV datasets 4 and 5), and moving cameras (UAV datasets 1 and 2)). As shown in Figure 4, the proposed OPV can accurately track small targets, even if the background regions contain many objects that appear similar to those of the targets.

Figure 5 shows visual tracking results over time for several UAV datasets. Our method accurately detected and tracked UAVs, in which these objects could disappear and appear again due to UAVs' and camera motions. As shown in Figure 6a,b, although UAVs change their sizes, our method robustly tracker the targets. As shown in Figure 6c, although UAVs have very irregular motions, our method successfully tracker the targets.


(**a**) UAV dataset 1 (**b**) UAV dataset 2 (**c**) UAV dataset 3

(**d**) UAV dataset 4 (**e**) UAV dataset 5 (**f**) UAV dataset 6

Figure 4. Qualitative results 1 using our UAV datasets. Red boxes denote visual tracking results of the proposed method.


(**a**) UAV dataset 1

(**b**) UAV dataset 2

(**c**) UAV dataset 3

Figure 5. Qualitative results 2 using our UAV datasets. Red boxes denote visual tracking results of the proposed method.

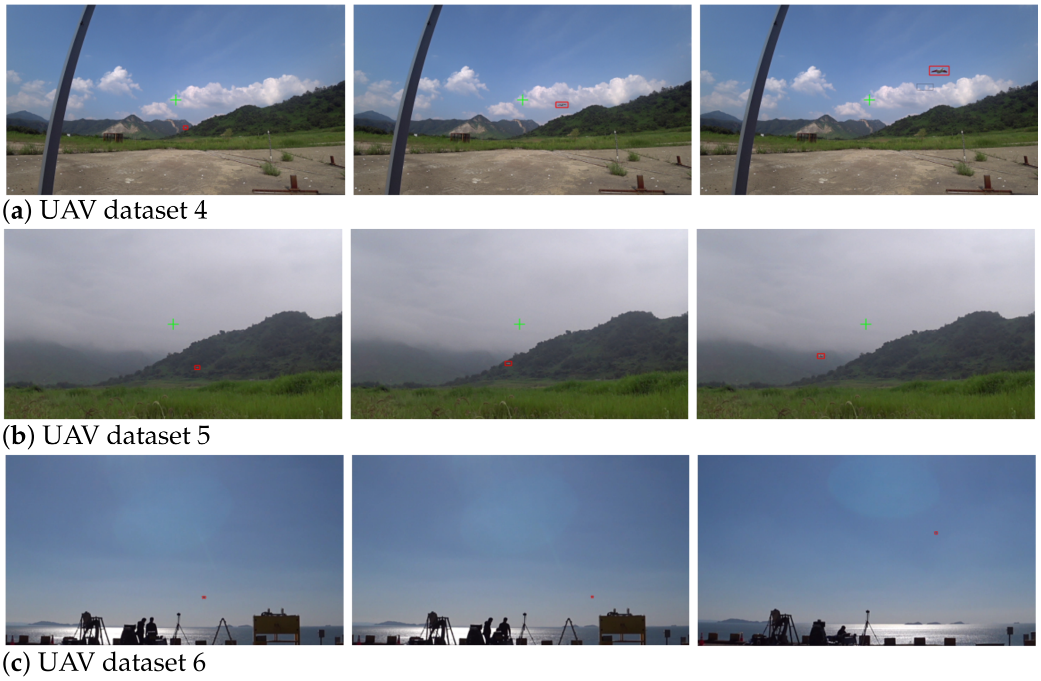


Figure 6. Qualitative results 3 using our UAV datasets. Red boxes denote visual tracking results of the proposed method.

Figure 7 demonstrates that the proposed method can successfully track UAVs with medium or large size, in which large objects are relatively insensitive to background clutters and object proposal methods can accurately estimate object candidate regions.

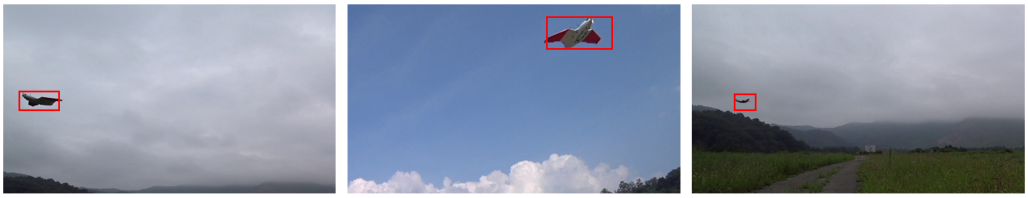


Figure 7. Qualitative results of UAVs with relatively medium or large size. Red boxes denote visual tracking results of the proposed method.

7. Conclusions

In this paper, we presented a novel object proposal method called OPV for visual tracking. OPV can enable the proposed visual tracker to track a small object regardless of severe background clutter. Experimental results verify that our method outperforms other conventional methods.

Author Contributions: Conceptualization, J.H. and J.K.; validation, J.H.; writing—original draft preparation, J.H.; writing—review and editing, J.H. and J.K.; supervision, J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the Chung-Ang University Graduate Research Scholarship Grants in 2020 and partly supported by NRF-2020R1C1C1004907.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to security issues.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kwon, J.; Lee, H. Visual Tracking Based on Edge Field with Object Proposal Association. *IVT* **2018**, *69*, 22–32. [CrossRef]
2. Cheng, M.; Zhang, Z.; Lin, W.; Torr, P. BING: Binarized normed gradients for objectness estimation at 300fps. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
3. Mesquita, D.B.; dos Santos, R.F.; Macharet, D.G.; Campos, M.F.M.; Nascimento, E.R. Fully Convolutional Siamese Autoencoder for Change Detection in UAV Aerial Images. *IEEE Geosci. Remote. Sens. Lett.* **2020**, *17*, 1455–1459. [CrossRef]
4. Wei, C.; Xia, H.; Qiao, Y. Fast Unmanned Aerial Vehicle Image Matching Combining Geometric Information and Feature Similarity. *IEEE Geosci. Remote. Sens. Lett.* **2020**, 1–5. [CrossRef]
5. Kitano, B.T.; Mendes, C.C.T.; Geus, A.R.; Oliveira, H.C.; Souza, J.R. Corn Plant Counting Using Deep Learning and UAV Images. *IEEE Geosci. Remote. Sens. Lett.* **2019**, 1–5. [CrossRef]
6. Huang, Y.; Liu, F.; Chen, Z.; Li, J.; Hong, W. An Improved Map-Drift Algorithm for Unmanned Aerial Vehicle SAR Imaging. *IEEE Geosci. Remote. Sens. Lett.* **2020**, 1–5. [CrossRef]
7. Berra, E.F.; Gaulton, R.; Barr, S. Commercial Off-the-Shelf Digital Cameras on Unmanned Aerial Vehicles for Multitemporal Monitoring of Vegetation Reflectance and NDVI. *IEEE Geosci. Remote. Sens. Lett.* **2017**, *55*, 4878–4886. [CrossRef]
8. Oh, B.; Guo, X.; Wan, F.; Toh, K.; Lin, Z. Micro-Doppler Mini-UAV Classification Using Empirical-Mode Decomposition Features. *IEEE Geosci. Remote. Sens. Lett.* **2018**, *15*, 227–231. [CrossRef]
9. Chiang, K.; Tsai, G.; Li, Y.; El-Sheimy, N. Development of LiDAR-Based UAV System for Environment Reconstruction. *IEEE Geosci. Remote. Sens. Lett.* **2017**, *14*, 1790–1794. [CrossRef]
10. Tetila, E.; Machado, B.; Belete, N.A.; Guimaraes, D.; Pistori, H. Identification of Soybean Foliar Diseases Using Unmanned Aerial Vehicle Images. *IEEE Geosci. Remote. Sens. Lett.* **2017**, *14*, 2190–2194.
11. Mandal, M.; Shah, M.; Meena, P.; Devi, S.; Vipparthi, S.K. AVDNet: A Small-Sized Vehicle Detection Network for Aerial Visual Data. *IEEE Geosci. Remote. Sens. Lett.* **2020**, *17*, 494–498. [CrossRef]
12. Zitnick, C.L.; Dollar, P. Edge boxes: Locating object proposals from edges. In Proceedings of the ECCV, Zurich, Switzerland, 6–12 September 2014.
13. Rantalankila, P.; Kannala, J.; Rahtu, E. Generating object segmentation proposals using global and local search. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
14. Hosang, J.H.; Benenson, R.; Dollar, P.; Schiele, B. What makes for effective detection proposals? *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 814–830. [CrossRef] [PubMed]
15. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep Learning for Generic Object Detection: A Survey. *Int J. Comput. Vis.* **2020**, *128*, 261–318. [CrossRef]
16. Pirinen, A.; Sminchisescu, C. Deep Reinforcement Learning of Region Proposal Networks for Object Detection. In Proceedings of the CVPR, Salt Lake City, UT, USA, 18–22 June 2018.
17. Hua, Y.; Alahari, K.; Schmid, C. Online object tracking with proposal selection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
18. Liang, P.; Pang, Y.; Liao, C.; Mei, X.; Ling, H. Adaptive objectness for object tracking. *SPL* **2016**, *23*, 949–953. [CrossRef]
19. Zhong, W.; Lu, H.; Yang, M. Robust object tracking via sparsity-based collaborative model. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012.
20. Hare, S.; Saffari, A.; Torr, P. Struck: Structured output tracking with kernels. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Brazil, 6–13 November 2011.
21. Kalal, Z.; Matas, J.; Mikolajczyk, K. P-n learning: Bootstrapping binary classifiers by structural constraints. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010.
22. Jia, X.; Lu, H.; Yang, M. Visual tracking via adaptive structural local sparse appearance model. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012.
23. Dinh, T.; Vo, N.; Medioni, G. Context tracker: Exploring supporters and distracters in unconstrained environments. In Proceedings of the CVPR 2011, Providence, RI, USA, 20–25 June 2011.
24. Kwon, J.; Lee, K. Tracking by sampling trackers. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Brazil, 6–13 November 2011.
25. Kwon, J.; Lee, K. Visual tracking decomposition. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010.
26. Henriques, J.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. In Proceedings of the 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012.
27. Liu, B.; Huang, J.; Yang, L.; Kulikowsk, C. Robust tracking using local sparse appearance model and k-selection. In Proceedings of the CVPR 2011, Providence, RI, USA, 20–25 June 2011.
28. Oron, S.; Bar-Hillel, A.; Levi, D.; Avidan, S. Locally orderless tracking. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012.
29. Zhang, Z.; Peng, H. Deeper and Wider Siamese Networks for Real-Time Visual Tracking. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–17 June 2019.
30. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks. In Proceedings of the CVPR, Salt Lake City, UT, USA, 18–22 June 2018.

31. Tao, R.; Gavves, E.; Smeulders, A.W. Siamese Instance Search for Tracking. In Proceedings of the CVPR, Las Vegas, NV, USA, 26 June–1 July 2016.
32. Pu, S.; Song, Y.; Ma, C.; Zhang, H.; Yang, M.H. Deep Attentive Tracking via Reciprocative Learning. In Proceedings of the NIPS, Montreal, QC, Canada, 3–8 December 2018.
33. Li, X.; Ma, C.; Wu, B.; He, Z.; Yang, M.H. Target-Aware Deep Tracking. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–17 June 2019.
34. Danelljan, M.; Bhat, G.; Shahbaz Khan, F.; Felsberg, M. ECO: Efficient Convolution Operators for Tracking. In Proceedings of the CVPR, Honolulu, HI, USA, 21–26 July 2017.
35. Danelljan, M.; Robinson, A.; Khan, F.; Felsberg, M. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016.
36. Suzuki, S.; Abe, K. Topological structural analysis of digitized binary images by border following. *CVGIP* **1985**, *30*, 32–46.
37. Oh, S.; Russell, S.; Sastry, S. Markov chain monte carlo data association for general multiple-target tracking problems. In Proceedings of the 2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601), Nassau, Bahamas, 14–17 December 2004.
38. Simonyan, K.; Gavves, E.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the ICLR, San Diego, CA, USA, 7–9 May 2015.
39. Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; Li, F.F. ImageNet: A LargeScale Hierarchical Image Database. In Proceedings of the CVPR, Miami, FL, USA, 20–25 June 2009.
40. Wu, Y.; Lim, J.; Yang, M.H. Online object tracking: A benchmark. In Proceedings of the CVPR, Portland, OR, USA, 23–28 June 2013.