Motion Cue Based Instance-level Moving Object Detection

Junjie Huang^{1,2}, Wei Zou^{1,2,3}, Zheng Zhu^{1,2},Jiagang Zhu^{1,2}

1. Institute of Automation, Chinese Academy of Sciences, Beijing, 100190

2. University of Chinese Academy of Sciences, Beijing, China

3. TainJin Intelligent Tech.Institute of CASIA Co.,Ltd, Tianjin, China

E-mail: {huangjunjie2016,wei.zou,zhuzheng2014,zhujiagang2015}@ia.ac.cn

Abstract: This paper studies the moving object detection, i.e., analyzing the amount, position and size of the moving objects in instance-level, which is meaningful for many computer vision problems. However, the existing methods are still not satisfying in accuracy, portability and speed. In this paper, we propose a novel framework which detects moving objects by analysis the consistency of the moving foreground. Instead of directly performing cluster algorithms on the moving foregound, we take two stages: analyzing the composition according to the local density of the moving foreground points and locating the targets by regressing some anchors. In this way, the proposed method doesn't need any training processes and can be efficiently performed to detect moving objects with arbitrary classes. Besides, we create our own publicly available dataset PDMOD with sufficient data, general challenges and convictive evaluation protocols to fill the scarcity of the evaluational datasets.

Key Words: Moving Object Detection, Dataset

1 Introduction

Object detection is a basic problem in computer vision. Driven by convolutional neural network (CNN) and deep learning, appearance-based object detection has achieved great success. Regression based methods[1] , region proposal based methods [2] and tracking methods [3][4] perform outstanding result both in public datasets[5, 6] and practical applications. However, they are data-driven and inconvenient in generalization as the models need to be trained again before applying to a new class. Detecting object based on motion cues can relieve this problem and is more feasible compared to appearance-based object detection in some applications like monitoring at night or tracking.

Aiming at detecting moving objects from complex scenes, many methods have been proposed [7, 8, 9, 10, 11]. Besides, some datasets have been collected and published [12, 13]. Among these works, a dominant paradigm for the output of moving object detection is the foreground masks. These masks consist of pixel-wise labels which provide a detailed discriminative result telling whether a pixel belongs to the moving foreground. However, it is less useful for the following questions like tracking and instance analysis, as there aren't any direct outputs indicating how many moving objects in the scenes, where are them, what are the sizes of them and which pixels belong to the same moving object, et.al.. Compared to the pixel-wise labels, these kinds of information are more useful for many subsequent processes. Thus, to a certain extent, the foreground masks obtained by aforementioned methods are unshaped and extra postprocessing is needed for obtaining the instance-level information of moving objects. We name this task as Instance-level Moving Object Detection (IMOD).

Recently, some works are contributed to this problem. Shen and Lu [14] taked the strategy of detecting objects based on appearance before classifying the objects' motion state, which limits their detection capacity to the target appearance. They used private data set and only tested their proposed method's classfication capacity. Siam et.al.[7] utilized CNN and deep learning to jointly learn both motion detection and target region proposal. The framework is designed and trained specially for vehicle detection instead of all class moving objects. They proposed the KITTI MOD dataset, whose test set only contains two video sequences and 497 annotated moving vehicles in total. As annotating specific for vehicle and missing the annotation for other moving objects in scenes, the dataset is only specific for the evaluation of their method. Besides, Zhou et.al.[15] detected and located the moving objects based on motion cues. Their method is based on stereo-vision system and requires up to 165 seconds to process each frame. Besides, quantitative analysis is missing in their experimental results. According to the above analysis, there are two aspects requirments in IMOD problem: Firstly a high-efficiency method which should be valid for single-view and for all class, and secondly a public challenging dataset for evaluation with convictive evaluational protocols.

To miss the first requirement, we propose a motion cue based robust framework which adopts the stategy of detecting the moving foreground before pursuing the instance-level information, as shown in Figure 1. We refer to the

This work is supported by Project of Development In Tianjin for Scientific Research Institutes Supported By Tianjin Goverment(project 16PTYJGX00050), and the National Nature Science Foundation of China(Grant No.61773374).

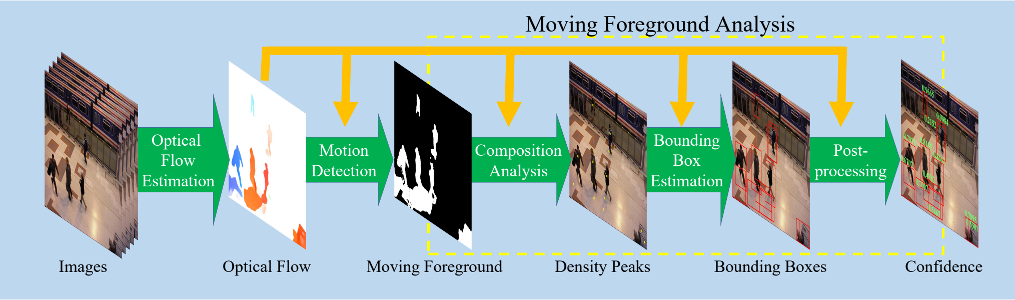


Figure 1: Visualization of the moving object detection framework.

motion detection works in [11] for moving foreground detection and concentrate on the moving foreground analysis in this work. Directly perform clustering algorithms on the moving foreground is highly discouraged, as they are time-consuming and not robust for this problem. We firstly imitate Alex Rodriguez and Alessandro Laio's idea of seaching peaks in density map [17] to efficiently analysis the foreground composition and obtain the potential object centers. After that, regression analysis is performed to accurately locate the edges of the targets. Finally, some priors are introduced to adjust the confidence scores of the detecting results. From coarse to fine, the method can be sped up to 20fps and equipped with better robustness.

To make the results more convictive, a new dataset is proposed, called Pedestrian Dataset for Moving Object Detection (PDMOD). It is extended from the pedestrian detection subset of CDnet2014[13] and has sufficient data and multiple common challenges. Inspired from single class object detection, we also create the toolkit for evaluating moving object detection. The PDMOD dataset and the evaluation toolkit have been made available at"https://github.com/HuangJunJie2017/PDMOD".

The contributions of this work are summarized as follows:

1. we construct a novel framework specific for instance-level moving object detection. The framework can be performed efficiently and is valid for arbitrary class moving object.

2. we create a publicly available dataset (PDMOD) with sufficient data, multiple common challenges and convitive evaluation protocol. s

The remainder of this paper is organized as follows. Our proposed analysis framework based on optical flow is introduced in Section 2 and its effectiveness is verified in Section 3 by comprehensive experiments. Besides, the proposed new dataset PDMOD and the relatived evaluational protocols are also introduced in Section 3. Finally, Section 4 is devoted to conclusions.

2 Methodology

Our strategy is to segment the motion detection result into different instances and obtain the instance-level information of moving objects. To this end, we construct the processing pipeline as shown in Figure 1. There are mainly four stages: motion detection, composition analysis, bounding box estimation and postprocessing. In the following, each procedure in this framework will be introduced in detail.

2.1 Motion Detection

Our moving object detection framework is based on the EMD[11], which can provide motion detection result $\{\mathbf{p}^{\mathrm{f}}\}$ as well as the optical flow while maintaining real-time property. We adopt the FN2-css-ft-sd optical flow estimation framework[11] and reduce the iterations of CRA[11] to 10. In this way, we speed up the EMD algorithm to 25fps.

2.2 Composition analysis

This subsection aims at figuring out how many moving objects in the scene, and at the same time, using some representative points to initially locate them. We consider this as a clustering problem. By using the image coordinate vector $\mathbf{p} = [\begin{array}{cc} x & y \end{array}]^T$ and the optical flow vector $\mathbf{f} = [\begin{array}{cc} u & v \end{array}]^T$ to construct a four-dimentional feature space, each pixel in an image is formulated into a point in this feature space. Inspired by Rodriguez A and Laio A [17], the peak points in the density maps are used to denote different individuals as shown in Figure 2.

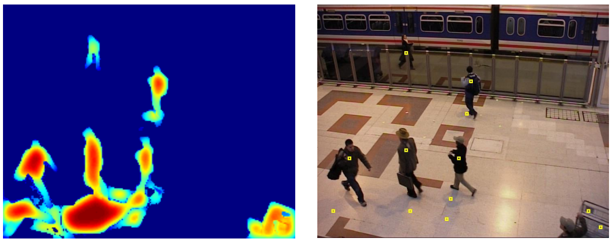


Figure 2: The local density map of the sample points is shown in the left. And Some major peaks are illustrated in the right.

At first, we sparsely sample points and only retain $\eta = 1/100$ of the total points, as sampling too much points contributes little to improve the system's performance, but causes a huge amount of computation. The computation complexity of formular 2 is proportional to the square of the sample point number. Then, we find out the foreground sample points $\{\mathbf{p}^{\mathrm{fs}}\}$ from all the sample points $\{\mathbf{p}^{\mathrm{s}}\}$ utilizing the foreground masks $\{\mathbf{p}^{\text{f}}\}$ provided by EMD [11]:

$$\{\mathbf{p}^{\text{fs}}\} = \{\mathbf{p}^{\text{s}}\} \cap \{\mathbf{p}^{\text{f}}\} \tag{1}$$

Subsequently, the local density ρ_i [17] of foreground point i is defined as

$$\rho_i = \sum_{j:\mathbf{p}_j \in \{\mathbf{p}^{\text{fs}}\}} \mathcal{R}(i,j) \tag{2}$$

where $\mathcal{R}$ is a similarity function that used to measure the contribution of foreground point j to the local density of foreground point i. $\mathcal{R}$ is defined as

$$\mathcal{R}(i,j) = \mathcal{S}(\mathbf{f}_i, \mathbf{f}_j, \lambda_f, \sigma_f)\mathcal{S}(\mathbf{p}_i, \mathbf{p}_j, \lambda_p, \sigma_p) \tag{3}$$

where $\mathcal{S}$ is a sigmoid function defined as:

$$\mathcal{S}(\mathbf{v}_1, \mathbf{v}_2, \lambda_v, \sigma_v) = 1/(1 + e^{\sigma_v(||\mathbf{v}_1 - \mathbf{v}_2||_1 - \lambda_v)}) \tag{4}$$

where λ is the cutoff distance and σ is the decay rate. We use abbreviation $\mathcal{S}_{\mathbf{v}_1,\mathbf{v}_2}$ instead in the following sections. $\mathcal{S}$ is expected to yield 1 when the first norm of the difference vector is less than the cutoff distance λ and yield 0 otherwise. Before being merged together, the optical flow feature and the coordinate feature are analyzed respectively to assign different proper parameters for these two kinds of feature.

Moreover, the distance δ_i from points of higher density [17] is defined as

$$\delta_i = \min_{j:\rho_j > \rho_i} (||\mathbf{p}_i - \mathbf{p}_j||_1) \tag{5}$$

where only the coordinate distance is taken into account. As shown in Figure 2, the coordinate distance is sufficient for figuring out the peaks. Finally, the set of peak points $\{\mathbf{p}^{\text{p}}\}$ is selected by the following criterion:

$$\{\mathbf{p}^{\text{p}}\} = \{\mathbf{p}^{\text{fs}} | \rho > \tau_r \wedge \delta > \tau_d)\} \tag{6}$$

where the isolated outliers are excluded by setting a threshold τ_r for the local density of the foreground points. Each peak point is considered as the geometric center of an independent moving object.

2.3 Bounding box estimation

As shown in Figure 2, the peak points obtained by formula (6) are coarse when they are considered as the geometric centers of the moving objects, due to the sparse sampling operation and the irregular shape of objects. In this section, we rely on the estimated peak points to get the bounding boxes of moving objects, being more accurate with size and aspect ratio messages.

To this end, as shown in the left of Figure 3, taking each peak point as the geometric center, we firstly set some anchor boxes $\{\mathbf{b}\}$ with different dimensions for reference. Unlike previous object detection methods [1] who employ fixed size anchor boxes, we utilize the prior that the size of a moving object is linear relative to the local density. The areas span three sizes

$$\{\frac{\lambda_s^{2k}\rho_i}{\eta}\}, k \in \{0,1,2\} \tag{7}$$

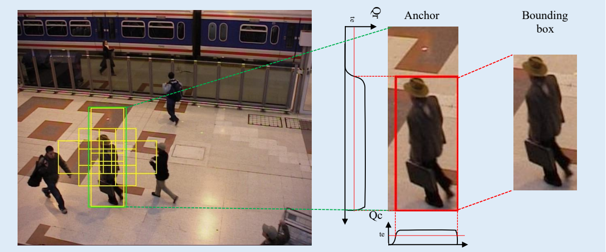


Figure 3: Visualization of bounding box estimation. The initial anchor boxes and the selected one are shown in the left. The refinement operation is illustrated in the middle and the bounding box result is in the right.

where η is defined in Section 2.2 and used as compensation for the sampling effect. In addition, the three aspect ratios are set to:

$$\{\lambda_a^k\}, k \in \{-1,0,1\} \tag{8}$$

Then the fitting degrees of the anchor boxes are evaluated to select the best one. Two aspects are considered during the scoring process:

1. The minimal anchor enclosing the moving.
2. Smmetrical content within the anchor.

According to these two considerations, two different initial scores are designed correspondingly: the content score $\mathcal{F}_c$ and the symmetry score $\mathcal{F}_s$:

$$\mathcal{F}_c = (1 + \beta_c) \sum_{j:\mathbf{p}_j^{\text{fs}} \in \mathbf{b}} \mathcal{S}_{\mathbf{f}_i,\mathbf{f}_j} - \frac{\beta_c}{\eta} b_w b_h \tag{9}$$

Where $\mathbf{b} = \begin{bmatrix} b_x & b_y & b_w & b_h \end{bmatrix}$ denotes the anchor box. b_x and b_y are the coordinates of the top left corner. b_w is the width, and b_h is the height. β_c is a penalty factor which is used to introduce the penalty term. In this way, the content score will peak when an anchor box is just big enough to completely cover a moving object.

As the used anchor box sizes are scattered, the penalty factors β_c needs to be small enough to avoid that the content score peaks in a small anchor size. However, this will reduce the ability of seperating objects which are moving side by side. This conflict problem is solved by introducing the prior that the contents in an anchor box should be central symmetric, and according to this, the symmetry score is defined as

$$\mathcal{F}_s = 1 - \frac{1}{2b_w b_h} \sum_{j:\mathbf{p}_j^{\text{fs}} \in \mathbf{b}} |\mathcal{S}_{\mathbf{f}_j,\mathbf{f}_p} - \mathcal{S}_{\tilde{\mathbf{f}}_j,\mathbf{f}_p}| \tag{10}$$

where $\tilde{\mathbf{f}}_j$ is the optical flow vector of the symmetry point:

$$\tilde{\mathbf{p}}_j = 2 \cdot \mathbf{p}^{\text{p}} - \mathbf{p}_j \tag{11}$$

$\mathcal{F}_s$ yields near to 1 when the content is highly central symmetric and 0 otherwise. The final score of an anchor is defined as

$$\mathcal{F} = \mathcal{F}_c \cdot \mathcal{F}_s \tag{12}$$

where $\mathcal{F}_s$ acts as another strong penalty factor. As a result, the one with the highest $\mathcal{F}$ score is selected.

As shown in Figure 3, the selected anchor boxes are usually slightly bigger than the desired bounding boxes. We shrink them by searching the content edges: Within the selected anchor box, a score is appended to each row $\mathcal{Q}_r(i)$ and each column $\mathcal{Q}_c(j)$. The row score $\mathcal{Q}_r(i)$ is defined as the maximal similarity between the peak point optical flow and the pixels' optical flow in this row.

$$\mathcal{Q}_r(i) = \max_{j:\mathbf{p}_j \in \mathbf{R}_i} \mathcal{S}_{\mathbf{f}_j, \mathbf{f}_p} \tag{13}$$

where $\mathbf{R}_i$ denotes the set of pixels in row i. Similarly, column score $\mathcal{Q}_c(j)$ is defined as

$$\mathcal{Q}_c(j) = \max_{i:\mathbf{p}_i \in \mathbf{R}_j} \mathcal{S}_{\mathbf{f}_i, \mathbf{f}_p} \tag{14}$$

By setting a threshold τ_e to the score curve, the intersection points which are closest to the anchor center are selected out as the edge of the final result. This operation is illustrates in Figure 3.

2.4 Postprocessing

Different confidence socres $\mathcal{C}$ are attached to the bounding boxes produced by the aforementioned processes. At first, we consider the prior that the detecting results are highly correlated between two consecutive frames. So the initial confidence socres of detecting results $\{\mathbf{d}\}_t$ in frame t are defined as

$$\mathcal{C}_{\mathbf{d}_i} = 0.3 + 0.7 \times \max_{j:\mathbf{d}_j \in \{\mathbf{d}\}_{t-1}} \mathrm{iou}(\mathbf{d}_i, \mathbf{d}_j) \tag{15}$$

where $\mathrm{iou}(\mathbf{d}_i, \mathbf{d}_j)$ is the intersection-over-union of two regions $\mathbf{d}_i$ and $\mathbf{d}_j$. A high confidence score is attached to the result which has high overlap with any results in the previous frame. Otherwise, a low initial score is attached the result considering it as a outlier or with low localization accuracy.

Besides, due to the arbitrary shape of moving objects, there may be more than one peak points corresponding to the same target. This will cause many false positive results. In this work, Soft-NMS [18] is performed to adjust the confidence scores. We sort all bounding boxes according to the score $\mathcal{F}$ defined in Section 2.3. The confidences of bounding boxes are adjusted by the following penalty function:

$$\mathcal{C}_{d_i} = \begin{cases} \mathcal{C}_{d_i}, & \text{if } \mathrm{iou}(\mathbf{d}_i, \mathbf{d}_l) < \tau_s \\ \mathcal{C}_{d_i}(1 - \mathrm{iou}(\mathbf{d}_i, \mathbf{d}_l)\mathcal{S}_{\mathbf{f}_i, \mathbf{f}_l}), & \text{if } \mathrm{iou}(\mathbf{d}_i, \mathbf{d}_l) \geq \tau_s \end{cases} \tag{16}$$

where $\mathbf{d}_l$ is the bounding boxes with higher $\mathcal{F}$ scores than $\mathbf{d}_i$ in frame t. After performing Soft-NMS [18], the confidence values of most false positive results are reduced to a low level. Figure 4 illustrates the effect of Soft-NMS algorithm.

3 Experiment

3.1 Dataset

Our proposed method is evaluated on the Pedestrian Detection Dataset(PDD), which is a subset of CDnet2014 [13].

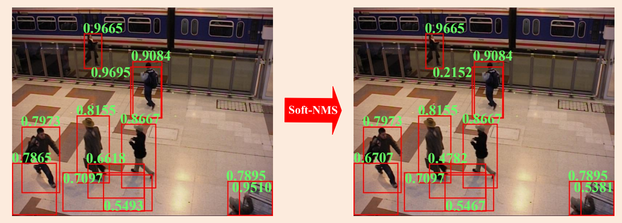


Figure 4: The effect of the Soft-NMS algorithm.

Algorithm 1 Moving Foreground Analysis

```
Input: image sequence $I_t$, moving foreground masks $\{\mathbf{p}^{\mathrm{f}}\}_t$,
  optical flow field $\{\mathbf{f}\}_t$;
sampling $I_t$ to obtain sample points $\{\mathbf{p}^{\mathrm{s}}\}_t$;
utilizing (1) to judge out foreground sample points $\{\mathbf{p}^{\mathrm{fs}}\}_t$ ;
utilizing (2) and (5) calculating $\rho_i$ and $\delta_i$ respectively;
utilizing (6) to judge out the density peak points $\{\mathbf{p}^{\mathrm{p}}\}_t$;
initializing anchors according to (7) and (8);
scoring the anchors according to (12) and selecting the one
  with highest score;
refining the selected anchor according to Figure 3;
initializing the confidence score according to (15);
performing Soft-NMS to refine the confidence score;
Output: bounding boxes set $\{\mathbf{d}\}_t$ and corresponding confi-
  dence score set $\{\mathcal{C}\}_t$.
```

PDD consists of ten video sequences which contain from 1099 to 7400 frames with spatial resolutions varying from 320 × 240 to 720 × 576. There are total 26248 frames with 14918 valid frames and total 21215 annotations in the form of rectangle bounding boxes. Different video sequences contain various challenging scenarios such as uncertain number of targets, irregular target shape, inconsistent motion speed inside target, hard shadow, and so on. Howerver, PDD is annotated specific to pedestrians and one fifth of them are static targets. We adjust PDD by removing the static annotations whose moving content accounts for less than 20% of the total area. As a result, containing 11653 valid frames and 16205 valid annotations, the adjusted dataset is dubbed as Pedestrian Dataset for Moving Object Detection(PDMOD).

3.2 Evaluation Metrics

As a single class object detection problem, we refer to the evaluation protocols which are introduced in the famous face detection dataset FDDB[19] and Wider-FACE [20]. In the first stage, we pay attention to the localization accuracy of the bounding box result. The confidence scores are ignored and all the detecting results are considered. Hungarian algorithm[21] is performed to find the best matching among the detecting results $\{\mathbf{d}\}_t$ and the annotations $\{\mathbf{a}\}_t$ in the same frame t. After that, the deteting result and the annotation in the same match are attached a match score $\mathcal{M}$, which is equal to the Intersection-over-Union score $\mathrm{iou}(\mathbf{d}, \mathbf{a})$ of this match. By setting different IoU thresholds τ_{iou}, the statistical score recall $\mathcal{R}$ is calculated by

$$\mathcal{R}(\tau_{\mathrm{iou}}) = \frac{1}{N_g} \sum_{i:\mathbf{d}_i \in \{\mathbf{d}\}} \chi(\mathcal{M}_i > \tau_{\mathrm{iou}}) \tag{17}$$

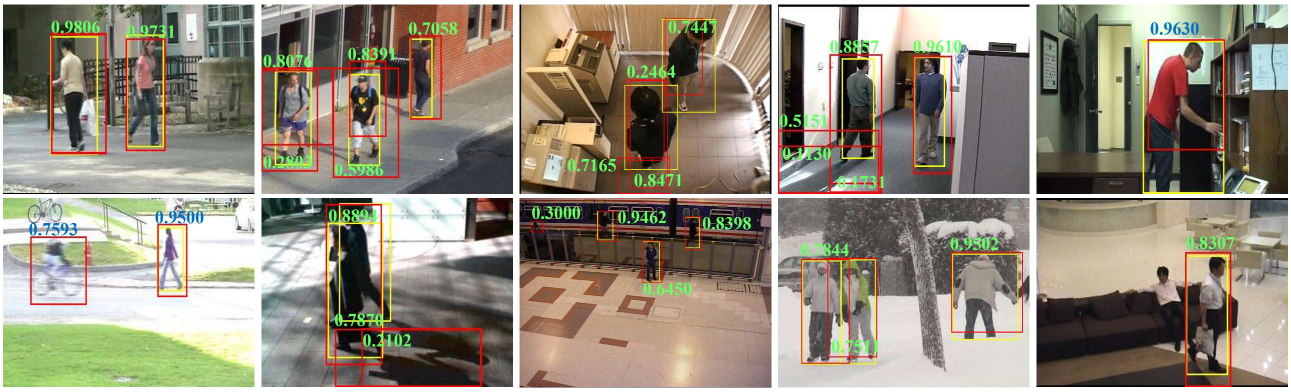


Figure 5: The bounding box results in some key frames. The yellow bounding boxes illustrate the ground truth annotations and the red bounding boxes illustrate the detecting results alone with confidence scores.

where $\chi(x) = 1$ if $x =$ true and $\chi(x) = 0$ otherwise. And the precision $\mathcal{P}$ is defined as

$$\mathcal{P}(\tau_{\text{iou}}) = \frac{1}{N_d} \sum_{i:\mathbf{a}_j \in \{\mathbf{a}\}} \chi(\mathcal{M}_j > \tau_{\text{iou}}) \tag{18}$$

In the second stage, we take confidence score into account and evaluate the detectability by using both ROC curve [19] and Precision-Recall curve [20]. The ROC curves and the Precision-Recall curves can tell the whole performance of the proposed methods.

3.3 Parameter setting

When analyzing composition, we set $\lambda_f = 10$, $\sigma_f = 0.3$, $\lambda_p = 25$ and $\sigma_p = 0.15$ in formula (3). $\tau_r = 3$, $\tau_d = 30$ in formular (6). When estimating bounding boxes, the parameters are set as $\lambda_s = 1.618$ in formula (7), $\lambda_a = 1.618$ in formula (8), $\beta_c = 0.01$ in formula (9). During postprocessing, $\tau_s = 0.3$ in formula (16).

3.4 Method for comparison

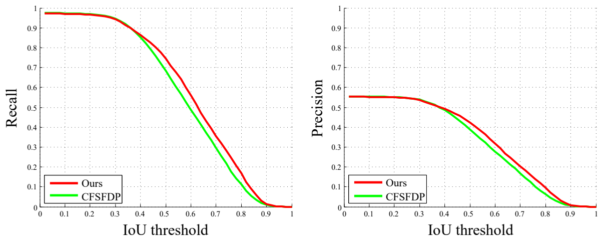
We compare the proposed method with the strategy that directly performing cluster algorithm CFSFDP [17]: The density peaks is selected out in the same way as ours. After that, each foreground sample point is attached to the nearest peak to form several clusters. The bounding boxes are obtained by finding the minimun rectangles that cover all the points in the same cluster. Finally, the confidence scores are calculated according to formular (15).

3.5 Qualitative result

Some bounding box results obtained by our proposed method are illustrated in Figure 5. Though the scenes contain different challenges, comparing the detecting results (the red bounding boxes) with the ground truths (the yellow bounding boxes), the proposed method outputs bounding boxes with high success rate and localization accuracy. Shadow and repeating detection cause most false positive results. As shown in the first image of the second row in Figure 5, there are a few frames which miss ground truth annotations, which also leads to false positive detections.

3.6 Quantitative result

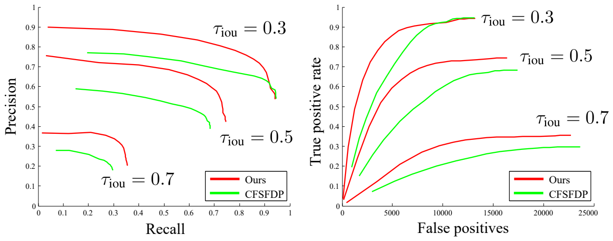
Figure 6 illustrates the $\mathcal{R}$-τ_{iou} curve and the $\mathcal{P}$-τ_{iou} curve of the proposed method and CFSFDP. The proposed method score 0.96 in $\mathcal{R}$ when using a low IoU threshold $\tau_{\text{iou}} < 0.3$. There is a 0.04 false negative rate which mostly due to the occlusion of multiple targets. A $\mathcal{P}$ score of 0.553 means that our proposed method score highly in $\mathcal{R}$ at the cost of a 0.447 false positive rate. Most false positive detecting results come from two respect: detecting the shadow as the moving object and producing more than one bounding boxes for a single target. When the IoU threshold exceed 0.3, both $\mathcal{R}$ score and $\mathcal{P}$ score drop down at a approximately linear rate. As the $\mathcal{R}$-τ_{iou} curve and the $\mathcal{P}$-τ_{iou} curve decays more slowly, the localization accuracy of the proposed method is more satisfying when compared to that of directly performing cluster algorithm (CFSFDP). Cluster algorithms have a trend to split the targets, which will cause a lower localization accuracy.



(a) Recall-τ_{iou} curves (b) Precision-τ_{iou} curves

Figure 6: The $\mathcal{R}$-τ_{iou} curves and the $\mathcal{P}$-τ_{iou} curves of our proposed method and CFSFDP[17].

By setting different IoU threshold τ_{iou} values, the P-R curves and the ROC curves of our proposed method is illustrated in Figure 7. The ceiling of the precision in P-R curves is caused by that the confidence score didn't tell any information about whether a bounding box is well positioning or whether the target is a shadow or a pedestrian. According to the P-R curves, the proposed method performs steadily when τ_{iou} maintains a low value, but quickly shrinks to the left bottom of the image when applying a higher τ_{iou} value. This is corresponding to change in the $\mathcal{R}$-τ_{iou} curve plotted in Figure 6. The proposed method



(a) Precision-Recall curves (b) ROC curves

Figure 7: The Precision-Recall curves and the ROC curves of the proposed method with respect to different τ_{iou} values.

performs more robust than directly performing cluster algorithm as it can attach different confidence scores to the bounding box results belong to the same target. By contrast, when a target is split into several parts, CFSFDP can't tell which part is better to be a delegate,.

3.7 Efficiency

We measure the computation time by Matlab on an Intel Core i5-7400 3.0GHz PC. Given the foreground masks and optical flow fields, it takes total 8.75 minutes for our method to process the PDMOD with 11653 valid frames and 28473 detecting results. The time consumption of a single frame (∼45ms) is linear relative to the target number in the scenes. When compared with other existing methods[7, 14, 15], the proposed method is the fastest one. As the foreground targets are processed independently, it can be further accelerated through GPU parallel processing.

4 Conclusion

Instance level moving object detection is a challenging but meaningful task. In this paper, we focus on this problem and propose an efficient strategy which can provide a certain level accuracy. The poposed framework doesn't need any training process and can be efficiently performed to detect any classes objects as long as they are moving in the sences. As the proposed method only takes optical flow as the only cue, the quality of its result rely on the accuracy of optical flow estimation to a great extent. Future works can introduce appearance cues to remove false positive detections caused by shadow and improve the localization accuracy. The proposed modified dataset can show the performance of instance-level moving object detection to a considerable height level. However, there are still some defects: the class of target is single, lack of the sences captured by non-stationary cameras and so on. Thus, a more convictive evaluational dataset is needed to be exploited.

REFERENCES

[1] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection, Proceedings of the IEEE conference on computer vision and pattern recognition, 779-788, 2016.

[2] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in neural information processing systems, 91-99, 2015.

[3] Zhu Z, Wu W, Zou W, et al. End-to-end flow correlation tracking with spatial-temporal attention[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 548-557.

[4] Zhu Z, Huang G, Zou W, et al. Uct: Learning unified convolutional networks for real-time visual tracking[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 1973-1982.

[5] Lin T Y, Maire M, Belongie S, et al. Microsoft coco: Common objects in context, European conference on computer vision. Springer, Cham, 740-755, 2014.

[6] Everingham M, Eslami S M A, Van Gool L, et al. The pascal visual object classes challenge: A retrospective, International journal of computer vision, 111(1): 98-136, 2015.

[7] Siam M, Mahgoub H, Zahran M, et al. MODNet: Moving Object Detection Network with Motion and Appearance for Autonomous Driving. arXiv:1709.04821, 2017.

[8] Tokmakov P, Alahari K, Schmid C, Learning motion patterns in videos, IEEE Conference on Computer Vision and Pattern Recognition, 531-539, 2017.

[9] Papazoglou A, Ferrari V, Fast object segmentation in unconstrained video, Proceedings of the IEEE International Conference on Computer Vision, 1777-1784, 2013.

[10] Jain S D, Xiong B, Grauman K, Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos, Proceedings of the IEEE conference on computer vision and pattern recognition, 1(2), 2017.

[11] Huang J, Zou W, Zhu J, et al. An Efficient Optical Flow Based Motion Detection Method for Non-stationary Scenes[J]. arXiv preprint arXiv:1811.08290, 2018.

[12] Perazzi F, Pont-Tuset J, McWilliams B, et al, A benchmark dataset and evaluation methodology for video object segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 724-732, 2016.

[13] Wang Y, Jodoin P M, Porikli F, et al, CDnet 2014: An expanded change detection benchmark dataset, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 387-394, 2014.

[14] Chen T, Lu S. Object-level motion detection from moving cameras, IEEE Transactions on Circuits and Systems for Video Technology, 27(11): 2333-2343, 2017.

[15] Zhou D, Frémont V, Quost B, et al. Moving object detection and segmentation in urban environments from a moving platform, Image and Vision Computing, 68: 76-87, 2017.

[16] Ilg E, Mayer N, Saikia T, et al. Flownet 2.0: Evolution of optical flow estimation with deep networks, IEEE conference on computer vision and pattern recognition, 2: 6, 2017.

[17] Rodriguez A, Laio A. Clustering by fast search and find of density peaks, Science, 344(6191): 1492-1496, 2014.

[18] Bodla N, Singh B, Chellappa R, et al. Soft-nms—improving object detection with one line of code, IEEE International Conference on Computer Vision, 5562-5570, 2017.

[19] Jain V, Learned-Miller E. Fddb: A benchmark for face detection in unconstrained settings, Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

[20] Yang S, Luo P, Loy C C, et al. Wider face: A face detection benchmark, Proceedings of the IEEE conference on computer vision and pattern recognition, 5525-5533, 2016.

[21] Kuhn H W. The Hungarian method for the assignment problem, Naval research logistics quarterly, 2(1): 83-97, 1955.