A Video Shot Occlusion Detection Algorithm Based on the Abnormal Fluctuation of Depth Information

Junhua Liao, *Graduate Student Member, IEEE*, Haihan Duan, *Student Member, IEEE*, Wanbing Zhao, Kanghui Feng, Yanbing Yang, *Member, IEEE*, and Liangyin Chen, *Member, IEEE*

***Abstract*— To make the video more attractive, original video materials usually need postprocessing by video editors, especially to eliminate low-quality abnormal clips, which seriously affect the visual effect. One of the main reasons for the low-quality abnormal clips is that there are occluders that accidentally break into the shot to occlude the protagonist, resulting in the loss of the video protagonist's information. However, it is time-consuming and laborious to manually find shot occlusion clips, so computer vision technology can be used to assist editors in completing this work. The previous solutions directly utilize neural networks to detect shot occlusion, so their performance is affected by the size and quality of the dataset. In contrast, inspired by the change of depth information in the frame caused by the occluder breaking into the shot, we propose an algorithm for video shot occlusion detection based on the fluctuation of depth information. This algorithm does not need occlusion data training and can detect shot occlusion well only by capturing the abnormal fluctuations of the frame depth information. Additionally, to overcome the defect in that the first video shot occlusion detection (VSOD) dataset released in our conference publication can only verify the sensitivity of detection methods, we expand the VSOD dataset to evaluate the comprehensive performance of detection algorithms. The plentiful experimental results show that, compared with state-of-the-art occlusion detection methods and self-designed baseline methods, our algorithm significantly improves the comprehensive performance of video shot occlusion detection. Furthermore, through verification on datasets with different data types and distributions, our shot occlusion detection algorithm can maintain an occlusion event recall of over 95%, while the false positive rate does not exceed 3%, demonstrating good generalization ability. To promote reproducible research, the code and dataset are available at https://github.com/Junhua-Liao/VSOD.**

***Index Terms*— Dataset, occlusion detection, human–computer interaction, automatic video editing.**

Manuscript received 28 December 2022; revised 20 April 2023 and 3 June 2023; accepted 10 July 2023. Date of publication 13 July 2023; date of current version 7 March 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62072319 and in part by the Department of Science and Technology of Sichuan Province under Grant 2022YFG0041. This article was recommended by Associate Editor Z. Tao. *(Corresponding author: Liangyin Chen.)*

Junhua Liao, Wanbing Zhao, and Kanghui Feng are with the College of Computer Science, Sichuan University, Chengdu 610044, China (e-mail: liaojunhua@stu.scu.edu.cn; wanbingzhao@stu.scu.edu.cn; fengkanghui@stu.scu.edu.cn).

Haihan Duan is with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen 518172, China (e-mail: haihanduan@link.cuhk.edu.cn).

Yanbing Yang and Liangyin Chen are with the College of Computer Science and the Institute for Industrial Internet Research, Sichuan University, Chengdu 610044, China (e-mail: yangyanbing@scu.edu.cn; chenliangyin@scu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2023.3295243.

Digital Object Identifier 10.1109/TCSVT.2023.3295243

I. Introduction

With the popularity of affordable and reliable cameras, the massive amount of videos that people shoot to record information has imposed a significant burden on editors. To assist or replace editors, there has been much research on automatic video editing [1], [2], [3], [4], [5], [6]. These studies mainly focus on video editing rules, and most default input videos are well shot materials with satisfactory quality. However, due to complex and changeable shooting environments, the quality of video materials is uneven, so removing low-quality anomaly clips should be the first and most crucial step toward achieving complete automation of video editing. In anomaly detection based on automatic video editing, shot occlusion detection is a meaningful and challenging new task [7]. Shot occlusion refers to the phenomenon where extraneous objects intrude into the shooting frame, resulting in the protagonist being occluded in the shooting process. Fig. 1 (a) and (c) show normal and occluded frames, respectively. Compared with Fig. 1(a), the protagonist's information in Fig. 1(c) is lost. To ensure the integrity of the protagonist's information in the finished film, finding and removing shot occlusion clips from video materials has always been one of the basic jobs of editors in post-processing. Therefore, it is indispensable to utilize computer vision technology to automatically detect shot occlusion clips in videos to reduce the workload of editors.

Most of the recent occlusion detection algorithms are based on neural networks [9], [10], [11], [12], [13], [14], [15], [16], so our previous work [7] also used neural networks to directly detect shot occlusion and released the VSOD dataset, which is the first dataset used specifically for the video shot occlusion detection task. However, this data-driven method [7] is prone to overfitting the limited training data, resulting in its excellent performance in specific scenarios under the same distribution as the training set and poor generalization ability in the wild. Photographers usually shoot in open areas, so the occlusion clips caused by occluders constitute a relatively minor proportion of the overall video. Since the average video duration of the previous VSOD dataset [7] is less than 4 seconds and the occlusion frame occupies most of the time, it can only verify the sensitivity of the algorithms, while the evaluation of the shot occlusion detection algorithms needs to comprehensively consider multiple indicators such as accuracy, sensitivity, and false positive rate. In summary, for complex and diverse shot occlusion, it is difficult to collect

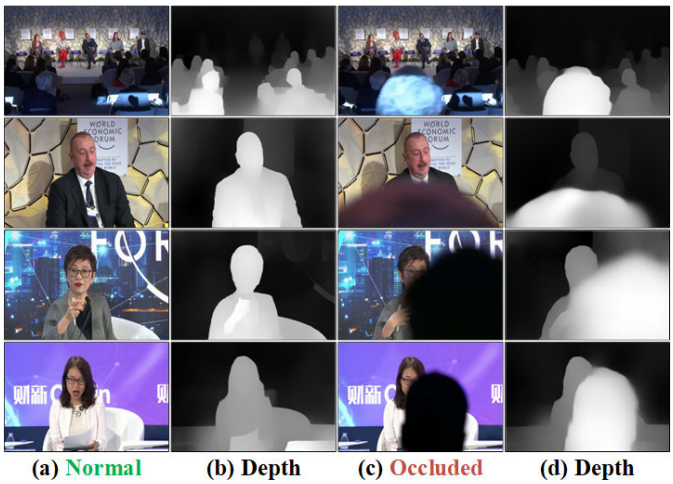

Fig. 1. Examples of normal and occluded frames and corresponding depth maps. (b) and (d) show the results predicted by the depth estimation model DPT [8]. The brighter area is closer to the shot.

abundant occlusion data to drive the neural networks to achieve good generalization ability. It is necessary to evaluate the comprehensive performance of the shot occlusion detection algorithms with more realistic data.

In this work, we seek to gain insights and address the challenging video shot occlusion detection problem. Because the occluder is closer to the shot than the protagonist and depth estimation methods are more sensitive to objects close to the shot, the appearance of the occluder results in significant changes in the frame depth information predicted by the depth estimation methods. This view is supported by Fig. 1(b) and Fig. 1(d). According to this phenomenon, we propose a shot occlusion detection algorithm based on the abnormal fluctuation of depth information. This non-data-driven algorithm is designed based on the commonality that occluders cause the change in depth information, and experimental results show that it has better generalization ability than methods driven by limited data. Moreover, we release version 2.0 of the VSOD dataset, which is composed of long videos that include shot occlusion, and most of the time in the video is normal. The expanded VSOD dataset can be used to more comprehensively evaluate the performance of the shot occlusion detection algorithms. We believe that the new VSOD dataset will further facilitate future research in this field.

In this work, we extend our previous conference publication [7], which proposes a video shot occlusion detection model built with 3D convolutional neural networks and contributes version 1.0 of the VSOD dataset. Unlike the conference publication that builds a neural network to directly detect shot occlusion, we try to design a video shot occlusion detection algorithm based on the physical principle of shot occlusion. Additionally, we contribute version 2.0 of the VSOD dataset to overcome the limitations of version 1.0.

The main contributions of this work are as follows:

- We propose an algorithm for detecting shot occlusion by leveraging abnormal fluctuations of depth information.
- We release version 2.0 of the VSOD dataset, including 200 active videos with shot occlusion, with a total duration of 25 hours. To the best of our knowledge, it is the first dataset for video shot occlusion detection.
- The experimental results on the VSOD dataset show that our proposed shot occlusion detection algorithm can drastically boost detection performance and achieve new state-of-the-art performance for this task. Moreover, this algorithm can maintain a recall of occlusion events of over 95% and a false positive rate of less than 3% on datasets with different data types and distributions, demonstrating good generalization ability.

The structure of this work is organized as follows: Section II introduces the related works; Section III describes the proposed video shot occlusion detection algorithm based on abnormal fluctuations of depth information; Section IV introduces the VSOD dataset we contributed; Section V demonstrates the experiments; Section VI discusses the limitation of this work and future work; and Section VII summarizes all the contributions of this work and provides the conclusion.

II. Related Works

A. Occlusion Detection

After making major breakthroughs in basic computer vision tasks such as image classification [17], object detection [18], and semantic segmentation [19], researchers have begun paying attention to some special scenarios, such as occlusion. Since occlusion can seriously affect the accuracy of recognition, it has received extensive attention in many tasks and has extended many new problems [9], [10], [11], [12], [13], [14], [15], [16]. Hou et al. [13] used the similarity between frame region features and video region features to judge whether occlusion occurred in the person re-identification task. In facial landmark detection, Zhu et al. [20] used a distillation module to infer the occlusion probability of each position in high-level features. Chi et al. [11] introduced a mask-guided module into a pedestrian detector in crowded scenes to enhance the discrimination ability of the occluded features. Song et al. [9] used the adaptive anchor initialization provided by the confidence-aware calibration for occluded pedestrian detection. Lazarow et al. [10] solved the problem of poor correlation between the ordering of instances with detection confidence and the natural occlusion relationship in panoptic segmentation by modeling two instance masks. Xu et al. [15] proposed an effective occlusion-aware network to handle occlusion for video deblurring.

Typically, most tasks define the occlusion scenario as the task target object being occluded. In contrast, the definition of occlusion in the shot occlusion detection task is more special. In a normal frame, the protagonist may also be occluded by a table, water bottle, or microphone in front of him/her. Within the shot occlusion detection task, the “uninvited guest” is considered an occluder that intrudes upon the frame and obstructs the visibility of the protagonist. Due to the different definitions of occlusion in different tasks, the corresponding design ideas for occlusion detection likewise vary. Therefore, directly migrating the state-of-the-art occlusion detection methods from other tasks to this task may not have satisfactory performance. We need to design detection methods according

to the characteristics of shot occlusion. Our previous work [7] detected shot occlusion by building a neural network model. However, the robustness of this data-driven parametric method is limited by the size and quality of the dataset, so we propose a non-data-driven solution to enhance the robustness of the shot occlusion detection method.

B. Depth Estimation

Depth estimation is one of the basic tasks in computer vision, which is widely used in object detection [21], semantic segmentation [22], simultaneous localization and mapping [23], and other fields. In recent years, depth estimation based on deep learning has made remarkable progress [24], [25], [26], [27]. In this process, deep datasets such as NYU depth [28], KITTI [29], and TUM [30] play an important role in driving neural networks. These datasets collected by depth sensors involve very limited scenarios, resulting in poor generalization ability of depth estimation methods trained on these datasets. To solve this problem, Chen et al. [24] crowdsourced a depth dataset, DIW, for wild images. Then, researchers used web stereo image, web stereo video, and internet video's Structure-from-Motion to generate wild depth datasets such as ReDWeb [31], WSVD [32], and YouTube3D [33]. Driven by these datasets, the depth estimation methods have broader practical scenarios.

Despite the existence of numerous depth datasets, they are individually biased and incomplete. To allow these datasets to complement each other to support the stable depth estimation of real images in different scenarios, Ranftl et al. proposed a depth estimation method called MiDaS [27]. This method was trained on a special MIX5 dataset (constructed by four public depth datasets [31], [32], [34], [35] and a homemade depth dataset using 3D movies), and its robustness was verified on the test set (constructed by six public depth datasets [24], [28], [29], [30], [36], [37]). Then, Ranftl et al. introduced five additional datasets [38], [39], [40], [41], [42] based on MIX5, constructed the largest monocular depth estimation training dataset MIX6 containing 1.4 million images, and proposed the dense prediction transformer (DPT) [8], which achieved state-of-the-art performance. This robust depth estimation model can provide effective depth information for different types of videos in the video editing task.

III. Algorithm

To enhance the robustness of the video shot occlusion detection algorithm, we design the detection algorithm according to the physical phenomenon of abnormal fluctuations of frame depth information caused by occluders. The performance of this non-data-driven algorithm is not affected by the size and quality of the training set. The detection process of our proposed video shot occlusion detection algorithm includes four steps. (1) The video frame sequence is preprocessed with a robust depth estimation model to obtain the depth map sequence as the algorithm's input. (2) The foreground sequence is extracted from the depth map sequence according to the dynamic depth threshold calculated by the Otsu algorithm. (3) The foreground sequence is used to calculate the foreground singleton-over-union we designed, that is, the proportion of the foreground of the singleton in the sequence to the union of the foreground of the entire sequence. (4) The foreground singleton-over-union of the whole video is analyzed to obtain the dynamic optimal occlusion threshold to predict the time at which shot occlusion occurs. Subsequent subsections provide detailed elaboration on each part of this video shot occlusion detection algorithm.

A. Data Preprocessing

Firstly, we need to preprocess the original video to obtain the depth information as the input of our algorithm. Since video editing tasks involve a wide range of scenarios, a highly robust depth estimation model is required to provide the depth information of videos shot in different scenes. As described in Section II-B, depth estimation technology is relatively mature, so we directly choose the DPT [8] with strong robustness driven by the depth estimation dataset containing 1.4 million images to provide depth information for our algorithm. Owing to the exceptional performance of the DPT, many studies [43], [44], [45], [46], [47] have chosen it for data preprocessing. Some depth maps predicted by this robust depth estimation model are shown in Fig. 1.

The preprocessed depth map sequence D_i is represented as:

$$D_i = \{d_1^i, d_2^i, \dots d_{n-1}^i, d_n^i\} \tag{1}$$

where i represents the ith second of the video, n is the frame rate of the video, and d_n^i denotes the nth frame depth map d of the ith second. The size of the d is 192×192 pixels.

B. Depth Foreground Segmentation

The depth map d maps different depth levels in the original picture to the depth interval of $[0, 255]$. Within this interval, the value 0 denotes the farthest distance from the shot, whereas the value 255 represents the closest distance to the shot. To further refine the depth sequence information, we employ the Otsu algorithm [48] to derive the global optimal depth threshold t_d for segmenting the foreground f and background b of the depth map d. f is the position of objects close to the camera, while b is the position of the remaining objects and the background. f, which includes the foreground information closest to the camera, is calculated as follows:

$$f(x, y) = \begin{cases} 1, & if\ d(x, y) > t_d \\ 0, & if\ d(x, y) \leq t_d \end{cases} \tag{2}$$

where x and y represent the abscissa and ordinate of the depth map d, respectively, and their values range from $[0, 191]$.

Subsequently, the foreground sequence F_i composed of f is sent to the next stage of the video shot occlusion detection algorithm. This sequence is defined as follows:

$$F_i = \{f_1^i, f_2^i, \dots f_{n-1}^i, f_n^i\} \tag{3}$$

where f_n^i indicates the nth frame f of the ith second.

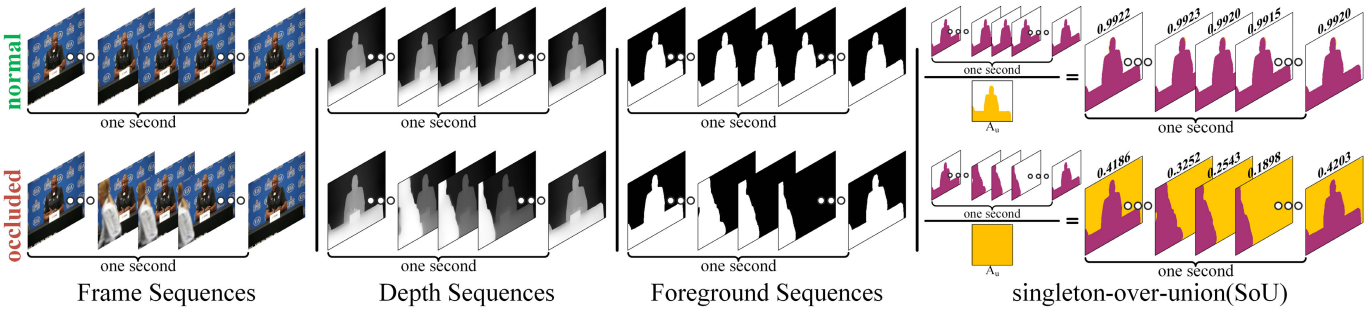

Fig. 2. A schematic illustration of the design principle of foreground singleton-over-union (SoU). SoU represents the proportion of the foreground area of the singleton in the sequence to the union of the foreground areas of the entire sequence, and the sequence length defaults to 1 second. Under normal conditions, SoU is in a high-value range close to 1, while under occlusion conditions, its value will significantly decrease.

C. Foreground Singleton-Over-Union

Photographers usually place the protagonist in the center of the picture by adjusting the shot, so the active area of the protagonist in the shot is relatively fixed. However, an occluder will produce a large-span movement track while it breaks into and leaves the shot. Since the occluder is closer to the shot than the protagonist, the motion of the occluder will cause considerable changes in the close-range area information recorded by the foreground f. Therefore, we design foreground singleton-over-union (SoU) to capture this information change. The SoU is calculated as follows.

First, we calculate the union area A_u^i of the foreground sequence F_i. A_u^i represents the area recorded by any foreground in F_i, defined as follows:

$$A_u^i(x,y) = \begin{cases} 1, & if \ \sum_{j=1}^{n} f_j^i(x,y) > 0 \\ 0, & if \ \sum_{j=1}^{n} f_j^i(x,y) = 0 \end{cases} \tag{4}$$

where x and y represent the abscissa and ordinate of f, respectively, and their values range from $[0, 191]$. n denotes the total number of frames in the ith second.

Second, SoU_i is calculated according to F_i and A_u^i.

$$SoU_i = \{\frac{sum(f_1^i)}{sum(A_u^i)}, \frac{sum(f_2^i)}{sum(A_u^i)}, \dots \frac{sum(f_{n-1}^i)}{sum(A_u^i)}, \frac{sum(f_n^i)}{sum(A_u^i)}\} \tag{5}$$

where $sum()$ represents the total number of non-zero pixel points in f_n^i and A_u^i.

Finally, the SoU_i sequence per second constitutes the sequence S for the entire video. S is defined as follows:

$$S = \{SoU_1, SoU_2, \dots SoU_{m-1}, SoU_m\} \tag{6}$$

where m is the total number of seconds in the video.

To facilitate understanding, a schematic illustration depicting the design principle of SoU is presented in Fig. 2. Theoretically, as the SoU value approaches 1, it indicates that the objects within the shot are in a state of relative stability and normalcy. On the contrary, it means that the occluder may intrude into the shot and move violently. Furthermore, we corroborate the viability of SoU via experiments. Fig. 3 shows the variation in SoU in the sample videos. When the shot is in a normal state, SoU maintains a high value and exhibits fluctuations that correspond to the action of the protagonist. Conversely, in the event of shot occlusion, SoU experiences a rapid and substantial decline. Therefore, our algorithm is capable of detecting shot occlusion via the identification of abnormal fluctuations present within the S.

Algorithm 1 Dynamic Occlusion Threshold Algorithm

Input: SoU sequence S
Output: optimal occlusion threshold t_o

$max(S)$: maximum value of S; $min(S)$: minimum value of S; $mean(S)$: mean value of S; $length(S)$: length of S
$mid(S) = \frac{max(S)+min(S)}{2}$
$t_{max} = max(mean(S), mid(S))$
$t_{min} = min(mean(S), mid(S))$
$\alpha = (1 - mean(S))^{length(S)}$
$t_n = (max(S) - 0.1) + 0.1 \times \alpha$
if $min(S) > t_n$ **then**
  $t_o = 0$
else
  $t_o = t_{min} + (t_{max} - t_{min}) \times \alpha$
end if
Return t_o

D. Optimal Occlusion Threshold Selection

After obtaining the video SoU sequence S, we need to determine a threshold to divide the normal and occluded shots. Due to the different motions of the protagonist and the occluder in different videos, the fluctuation amplitudes of SoU are also different. In this case, the fixed threshold can not achieve satisfactory results, so we propose a dynamic occlusion threshold algorithm to find the optimal occlusion threshold t_o. The calculation process is shown in Algorithm 1.

This dynamic occlusion threshold algorithm first calculates the maximum value $max(S)$, minimum value $min(S)$, mean value $mean(S)$, length $length(S)$, and middle value $mid(S)$ of the input sequence S. Second, the upper limit of the occlusion threshold is determined by selecting the maximum value from $mean(S)$ and $mid(S)$, while the lower limit is set to the minimum value. Then, the algorithm calculates the amplification factor α of the occlusion threshold to determine the optimal occlusion threshold t_o within the threshold interval.

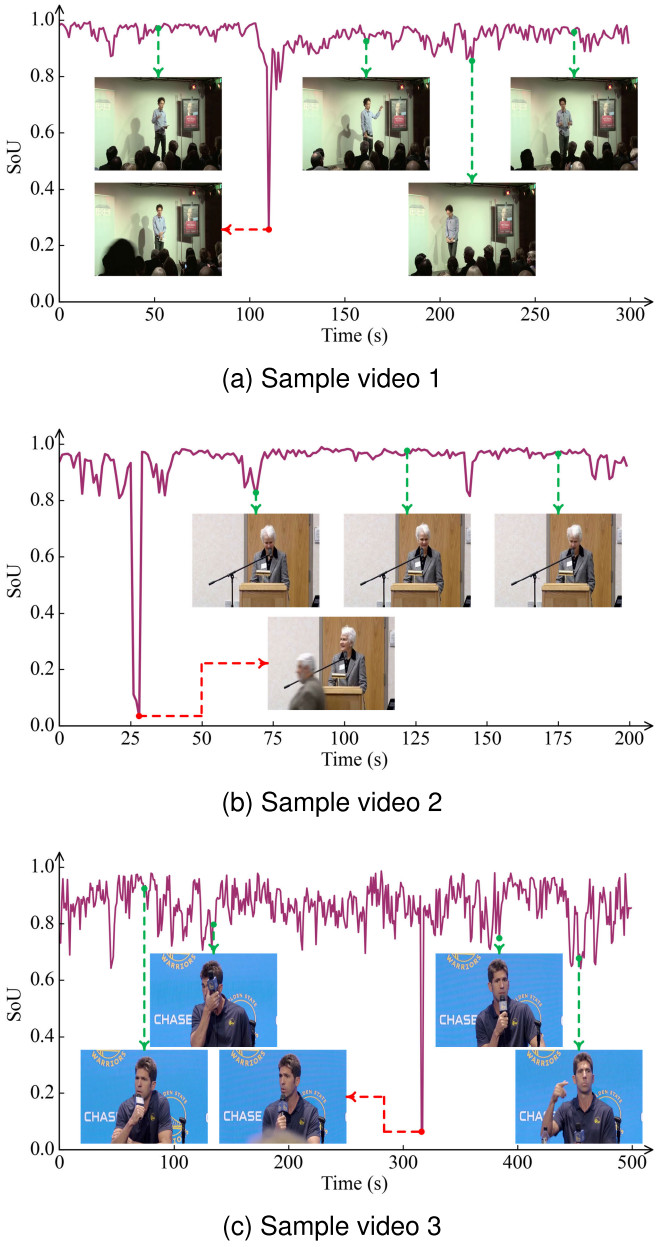

Fig. 3. Changes in SoU for different videos. The green dots represent normal frames, and the red dots represent occluded frames. The SoU value at the ith second of the video is represented by the minimum value in the SoU_i.

For videos of equal duration, a larger mean value $mean(S)$ signifies that the overall SoU is more stable. In this case, a small threshold is sufficient to ensure the recognition rate of shot occlusion while minimizing the likelihood of misjudging fluctuations in SoU resulting from the protagonist's movements as occlusion. Conversely, this indicates that the SoU of the entire video is exhibiting significant fluctuations, necessitating a large α to increase the threshold to avoid missed detection. When the mean value $mean(S)$ is the same, a longer video exhibits greater stability in terms of its SoU and, consequently, possesses a relatively smaller α. We use $mean(S)$ and $length(S)$ to construct functions that satisfy the above rules to derive the amplification factor α, as shown in Fig. 4. Finally, we obtain the optimal occlusion threshold t_o in

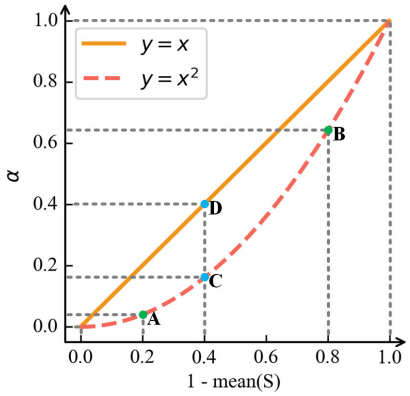

Fig. 4. Examples of amplification factor α for different videos. Videos A, B, and C have a duration of two minutes, but the mean values are 0.8, 0.2, and 0.6 respectively. Video D lasts for one minute with a mean value of 0.6.

Fig. 5. Examples of different occlusion detection datasets.

the occlusion threshold interval according to α. Nevertheless, not all videos exist shot occlusion clips, so we set a normal video threshold t_n based on α in the interval of 0.1 down from the maximum value $max(S)$. Its principle is similar to the occlusion threshold t_o, where a smaller threshold is required to be set for a more stable SoU. If the minimum value $min(S)$ of the sequence S is greater than the normal threshold t_n, it can be inferred that no shot occlusion occurs in the video.

Next, we predict the state of each second in the video according to the optimal occlusion threshold t_o. When the minimum value of SoU_i is less than the threshold t_o, our algorithm predicts that shot occlusion occurs in the ith second of the video. Otherwise, the ith second of the video is normal.

$$r_i = \begin{cases} 1, & if\ min(SoU_i \in S) < t_o \\ 0, & if\ min(SoU_i \in S) \geq t_o \end{cases} \tag{7}$$

Finally, our algorithm yields the predicted result R.

$$R = \{r_1, r_2, \ldots r_{m-1}, r_m\} \tag{8}$$

IV. VSOD Dataset

As a hot research topic, there are many related datasets for occlusion detection [29], [49], [50], [51]. Fig. 5 shows examples of occlusion detection datasets in different task scenarios. Due to the different scenarios and definitions of occlusion for different tasks, there are significant differences

Fig. 6. Examples of different videos in the VSOD dataset.

between different occlusion detection datasets, making existing datasets difficult to apply to the video shot occlusion detection task. To this end, we contribute the first large-scale video shot occlusion detection dataset, namely VSOD, which serves as a benchmark for evaluating the performance of shot occlusion detection methods. Fig. 6 shows five frames of the sample videos in the VSOD dataset.

TABLE I
COMPARISON OF DIFFERENT VERSIONS OF THE VSOD DATASET

Dataset version	VSOD v1.0 [7]	VSOD v2.0
Number of videos	1000	200
Average occlusion percentage	79.99%	4.05%
Average occlusion event	1.02	2.58
Average frames	73	11516
Dataset length	58 min	25 hours 37 min

A. VSOD v1.0 Dataset

Version 1.0 of the VSOD dataset was published in our conference publication [7]. Three annotators with basic computer vision knowledge first looked for videos with predefined occlusion taken in the real world from YouTube.[1] Second, the annotators used Corel VideoStudio[2] to clip the shot occlusion clips in the video materials into short videos and reencoded them with the "XviD" video codec to reduce storage space and facilitate the download of the dataset. Finally, the bounding boxes of occlusion in each frame are annotated for the potential of further extension. The VSOD v1.0 dataset includes a total of 1000 videos with shot occlusion, with the dataset being randomly divided into the training set and testing set according to the ratio of 8:2.

B. VSOD v2.0 Dataset

As the first dataset for the video shot occlusion detection task, the VSOD v1.0 dataset contains 1000 videos, but the average length of the videos is only 3.486 seconds. Due to constraints imposed by video length, this dataset comprises only a limited number of normal clips. Therefore, the VSOD v1.0 dataset can only verify the sensitivity of the shot occlusion detection methods. In video shooting, since shot occlusion is a low-probability event, the normal period of videos is usually much longer than the period of shot occlusion. If a high-sensitivity detection method results in high false positives, editors still need to review the predicted occlusion clips, thereby failing to effectively reduce their workload. To better implement the shot occlusion detection algorithm, we also need a large number of normal clips to verify the false positive rate of the detection algorithm.

In this work, we also contributed version 2.0 of the VSOD dataset to overcome the limitations of version 1.0. This dataset consists of 200 videos of large-scale activities in the real world. These video data are selected from YouTube by four annotators with knowledge of computer vision, taking several months. To evaluate the performance of video shot occlusion detection methods, we need to know the temporal annotations, i.e., the start and end time points of the occlusion event in each video. To this end, we invite an editor with seven years of experience to annotate the time extent of occlusion events.

The version differences of the VSOD dataset are shown in Table I. Although the VSOD v2.0 dataset has only 200 videos, the average duration of each video is as long as 7.688 minutes, which is 132 times that of the VSOD v1.0 dataset. Since shot occlusion is an accidental event, the average occlusion percentage of 4.05% in the VSOD v2.0 dataset is closer to the actual situation than the average occlusion percentage of 79.99% in the VSOD v1.0 dataset, i.e., video is usually dominated by normal clips. In addition, Fig. 7 shows the VSOD dataset from the perspectives of video length and occlusion percentage to visually display the differences between versions. The extended VSOD dataset can more

[1] https://www.youtube.com/
[2] https://www.videostudiopro.com/

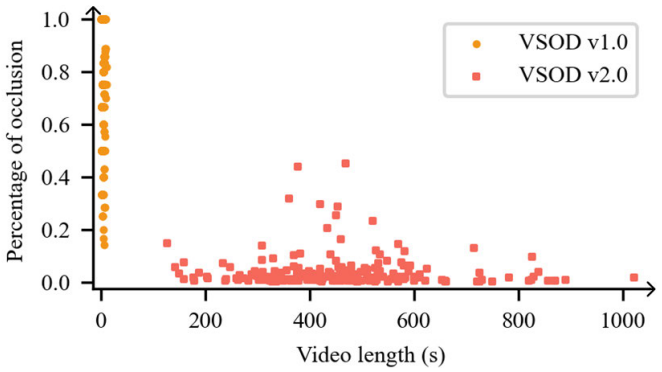

Fig. 7. Length and occlusion percentage of videos in the VSOD dataset. The yellow circles and red squares represent videos in the VSOD v1.0 dataset and the VSOD v2.0 dataset, respectively.

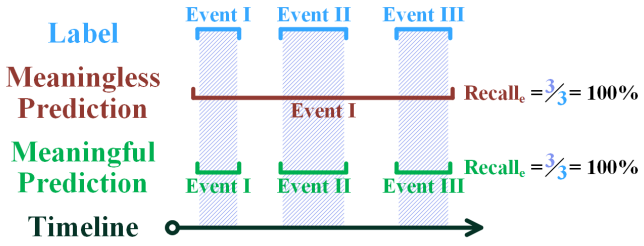

Fig. 8. The abnormal situation of Recall$_e$. Although three occlusion events are detected in both predictions, the meaningless prediction almost regards the entire video as an occlusion event, which causes the editors to spend extra effort checking manually.

comprehensively and realistically evaluate the performance of video shot occlusion detection methods.

V. Experiments

A. Implementation Details

We implement all deep learning models with PyTorch [54] and carry out performance evaluations on a standard Deepin-15.11 OS with an NVIDIA GTX 1080Ti GPU (11GB). The batch size is set to 8, and the maximal training epoch is fixed to 50. We use the stochastic gradient descent (SGD) with a momentum of 0.9 and weight decay of 0.0005 to train models.

B. Evaluation Metric

Editors pay more attention to whether the algorithm can detect all occlusion events in the video, so we choose the event recall (Recall$_e$) as the evaluation metric, where the event represents the process after which the occluder enters the shot and before which the occluder leaves the shot. Nevertheless, reliance on this metric solely may lead to an inaccurate evaluation of shot occlusion detection methods due to the deceptive nature of its value. Fig. 8 shows an abnormal situation where the Recall$_e$ values for both meaningful and meaningless prediction are 100%. While this meaningless prediction yields perfect results in terms of valuation performance, it is useless for editors. Because it almost predicts the entire video to be an occlusion event, the editors need to watch the entire video to complete a manual inspection, which cannot effectively reduce the workload of editors. Although this is an extreme situation, it still exposes the deception of the current evaluation metric when the algorithm detects a long-term occlusion event.

To make up for the deficiency of Recall$_e$ and more accurately evaluate the performance of the shot occlusion detection methods, we introduce the evaluation metrics in seconds: Intersection-over-Union$_s$, Accuracy$_s$, False Positive Rate$_s$, Recall$_s$, Precision$_s$, and F1 Score$_s$. Seconds are chosen as the base unit because second-level predictions are sufficient to assist editors in locating and eliminating shot occlusion clips in the postprocessing process.

C. Comparison With the State-of-the-Art

In this subsection, we compare the proposed video shot occlusion detection algorithm with nine different detection methods, including the state-of-the-art video shot occlusion detection method [7], two classic classification methods [52], [53], and the state-of-the-art occlusion detection methods used in person re-identification [13], facial landmark detection [20], pedestrian detection [9], [11], panoptic segmentation [10], and X-ray security inspection [12].

We first train the comparison methods on the training set of the VSOD v1.0 dataset. Table II and Table III show the quantitative comparison results of shot occlusion detection algorithms on the test set of the VSOD v1.0 dataset and the VSOD v2.0 dataset, respectively. Experimental results demonstrate that our video shot occlusion detection algorithm can also significantly outperform state-of-the-art occlusion detection methods without occlusion data driving. On the test set of the VSOD v1.0 dataset [7], this algorithm ranks first in 5 metrics (Recall$_e$, Accuracy$_s$, False Positive Rate$_s$, Precision$_s$, and F1 Score$_s$) and second in 1 metric (Intersection-over-Union$_s$). Moreover, our algorithm has 5 metrics (Intersection-over-Union$_s$, Accuracy$_s$, False Positive Rate$_s$, Precision$_s$, and F1 Score$_s$) ranking first in the VSOD v2.0 dataset. Our algorithm demonstrates strong generalization ability, achieving an ultra-high event recall ($> 95\%$) and a low false positive rate ($< 3\%$) across the entire VSOD dataset. Fig. 9 presents the shot occlusion clips of sample videos predicted by different methods. Compared with those of other methods, the prediction results of our algorithm closely match the ground truth.

The experimental results show that the classic classification method ResNet-101 [52] exhibits strong competitiveness in terms of Recall$_e$ and Recall$_s$. However, as shown by the prediction results in Fig. 9, ResNet-101 tends to make predictions that treat the entire video as an occlusion event, that is, the meaningless prediction mentioned in Fig. 8. Since the VSOD v1.0 dataset is mainly composed of occlusion clips, even if the method makes meaningless predictions, Accuracy$_s$ is 78.14%. However, on the VSOD v2.0 dataset dominated by normal clips, meaningless predictions significantly raise the False Positive Rate$_s$ of ResNet-101 to 96.6%, thereby causing its Accuracy$_s$ to plummet to a mere 7.36%. This also confirms that the VSOD v1.0 dataset has defects and cannot really reflect the comprehensive performance of the shot occlusion detection algorithms. In addition, the Precison$_s$ of only 4.25% indicates that more than 95% of the shot occlusion predictions made by ResNet-101 are false positives, so such a method cannot help editors eliminate the shot occlusion clips.

TABLE II
COMPARISON OF SHOT OCCLUSION DETECTION PERFORMANCE ON THE TEST SET OF VSOD V1.0 DATASET

Methods	Recall_e	$\text{Intersection-over-Union}_s$	Accuracy_s	$\text{False Positive Rate}_s$	Recall_s	Precision_s	F1 Score_s
ResNet-101 [52]	0.9900	0.7730	0.7814	0.5100	0.9696	0.7918	0.8480
DenseNet-169 [53]	0.9400	0.7680	0.7939	0.3785	0.9112	0.7968	0.8282
STCnet [13]	0.2950	0.1952	0.3526	0.0565	0.2191	0.2694	0.2228
ODN [20]	0.9950	0.8528	0.8613	0.3317	**0.9759**	0.8716	0.9049
PRNet [9]	0.9850	0.8406	0.8453	0.3417	0.9686	0.8562	0.8926
OCFusion [10]	0.9150	0.7398	0.7683	0.3467	0.8742	0.7798	0.7999
PedHunter [11]	0.5550	0.4473	0.5422	0.1540	0.5066	0.4945	0.4833
DOAM [12]	0.9600	0.8411	0.8525	0.1525	0.9011	0.8992	0.8830
Liao et al. [7]	0.9800	**0.8692**	0.8798	0.1624	0.9511	0.8982	0.9067
Our Method	**1.0000**	0.8623	**0.8806**	**0.0214**	0.8761	**0.9833**	**0.9105**
CS-Video(thr=0.9)	0.3900	0.2816	0.4145	0.0321	0.3000	0.3715	0.3139
HS-Video(thr=0.9)	**1.0000**	0.7821	0.7821	0.5700	**1.0000**	0.7821	0.8582
LCC-Video(thr=0.9)	0.7700	0.6123	0.6638	0.1745	0.6839	0.6984	0.6652
SROCC-Video(thr=0.9)	0.7800	0.6200	0.6662	0.2200	0.7068	0.6932	0.6735
Our Method	**1.0000**	**0.8623**	**0.8806**	**0.0214**	0.8761	**0.9833**	**0.9105**
CS-Video(thr=dynamic)	0.8200	0.6823	0.7336	0.0475	0.7046	0.7944	0.7291
HS-Video(thr=dynamic)	**1.0000**	0.8325	0.8416	0.3232	**0.9527**	0.8690	0.8916
LCC-Video(thr=dynamic)	0.9250	0.7537	0.7896	0.0841	0.7903	0.8792	0.8114
SROCC-Video(thr=dynamic)	0.9700	0.8174	0.8395	0.0848	0.8557	0.9265	0.8695
Our Method	**1.0000**	**0.8623**	**0.8806**	**0.0214**	0.8761	**0.9833**	**0.9105**
CS-Frame(thr=0.9)	0.1800	0.1326	0.3010	**0.0050**	0.1351	0.1775	0.1448
HS-Frame(thr=0.9)	0.9900	0.7813	0.7937	0.3946	**0.9339**	0.8285	0.8522
LCC-Frame(thr=0.9)	0.5100	0.3768	0.4862	0.0354	0.3898	0.4958	0.4176
SROCC-Frame(thr=0.9)	0.5200	0.4000	0.5045	0.0317	0.4158	0.5042	0.4370
Our Method	**1.0000**	**0.8623**	**0.8806**	0.0214	0.8761	**0.9833**	**0.9105**
CS-Frame(thr=dynamic)	0.6100	0.5357	0.6251	**0.0175**	0.5437	0.6020	0.5602
HS-Frame(thr=dynamic)	**1.0000**	0.8340	0.8414	0.3992	**0.9870**	0.8453	0.8937
LCC-Frame(thr=dynamic)	0.8300	0.7282	0.7645	0.0514	0.7524	0.8032	0.7631
SROCC-Frame(thr=dynamic)	0.8950	0.8047	0.8266	0.0521	0.8279	0.8706	0.8351
Our Method	**1.0000**	**0.8623**	**0.8806**	0.0214	0.8761	**0.9833**	**0.9105**

TABLE III
COMPARISON OF SHOT OCCLUSION DETECTION PERFORMANCE ON THE VSOD V2.0 DATASET

Methods	Recall_e	$\text{Intersection-over-Union}_s$	Accuracy_s	$\text{False Positive Rate}_s$	Recall_s	Precision_s	F1 Score_s
ResNet-101 [52]	0.9950	0.0424	0.0736	0.9660	**0.9883**	0.0425	0.0748
DenseNet-169 [53]	0.9499	0.1137	0.2672	0.7652	0.9121	0.1365	0.1568
STCnet [13]	0.2444	0.1017	0.9288	0.0350	0.1569	0.2193	0.1343
ODN [20]	0.9779	0.1120	0.3156	0.7169	0.9681	0.1215	0.1641
PRNet [9]	0.9762	0.1545	0.4008	0.6273	0.9336	0.1783	0.2081
OCFusion [10]	0.8964	0.1367	0.4032	0.6169	0.8094	0.2043	0.1865
PedHunter [11]	0.7501	0.1497	0.6177	0.3801	0.6292	0.2134	0.2017
DOAM [12]	0.9695	0.2746	0.6891	0.3228	0.8643	0.3157	0.3520
Liao et al. [7]	**0.9975**	0.2072	0.6613	0.3490	0.9345	0.2309	0.2847
Our Method	0.9588	**0.5128**	**0.9531**	**0.0252**	0.6623	**0.7662**	**0.6190**
CS-Video(thr=0.9)	0.4754	0.2216	0.9365	0.0318	0.3612	0.3816	0.2816
HS-Video(thr=0.9)	**1.0000**	0.0405	0.0405	1.0000	**1.0000**	0.0405	0.0716
LCC-Video(thr=0.9)	0.8511	0.1979	0.7006	0.2909	0.7638	0.2672	0.2647
SROCC-Video(thr=0.9)	0.8682	0.2110	0.6664	0.3291	0.7694	0.2920	0.2806
Our Method	0.9588	**0.5128**	**0.9531**	**0.0252**	0.6623	**0.7662**	**0.6190**
CS-Video(thr=dynamic)	0.4097	0.2239	**0.9539**	**0.0130**	0.3187	0.3754	0.2793
HS-Video(thr=dynamic)	0.7220	0.2070	0.7239	0.2650	0.5704	0.3056	0.2693
LCC-Video(thr=dynamic)	0.7093	0.3311	0.9413	0.0321	0.5375	0.5515	0.4204
SROCC-Video(thr=dynamic)	0.7042	0.3433	0.9482	0.0244	0.5295	0.5796	0.4339
Our Method	**0.9588**	**0.5128**	0.9531	0.0252	**0.6623**	**0.7662**	**0.6190**
CS-Frame(thr=0.9)	0.1318	0.0502	**0.9601**	**0.0004**	0.0535	0.2045	0.0709
HS-Frame(thr=0.9)	**0.9894**	0.0505	0.2575	0.7671	**0.9125**	0.0530	0.0886
LCC-Frame(thr=0.9)	0.4259	0.1669	0.9563	0.0085	0.2439	0.3590	0.2279
SROCC-Frame(thr=0.9)	0.4253	0.1751	0.9536	0.0113	0.2543	0.3552	0.2308
Our Method	0.9588	**0.5128**	0.9531	0.0252	0.6623	**0.7662**	**0.6190**
CS-Frame(thr=dynamic)	0.1512	0.0644	**0.9594**	**0.0019**	0.0756	0.1847	0.0910
HS-Frame(thr=dynamic)	0.9012	0.1051	0.3685	0.6574	**0.8332**	0.1543	0.1565
LCC-Frame(thr=dynamic)	0.3681	0.1451	0.9589	0.0042	0.1844	0.3719	0.2011
SROCC-Frame(thr=dynamic)	0.3746	0.1523	0.9574	0.0061	0.1985	0.3833	0.2103
Our Method	**0.9588**	**0.5128**	0.9531	0.0252	0.6623	**0.7662**	**0.6190**

For the state-of-the-art occlusion detection methods [9], [10], [11], [12], [13], [20] in other tasks, due to the different definitions of occlusion in different tasks and their different design motivations, these methods do not perform well after migrating to the shot occlusion detection task. STCnet [13] uses a neural network to calculate the feature map of each

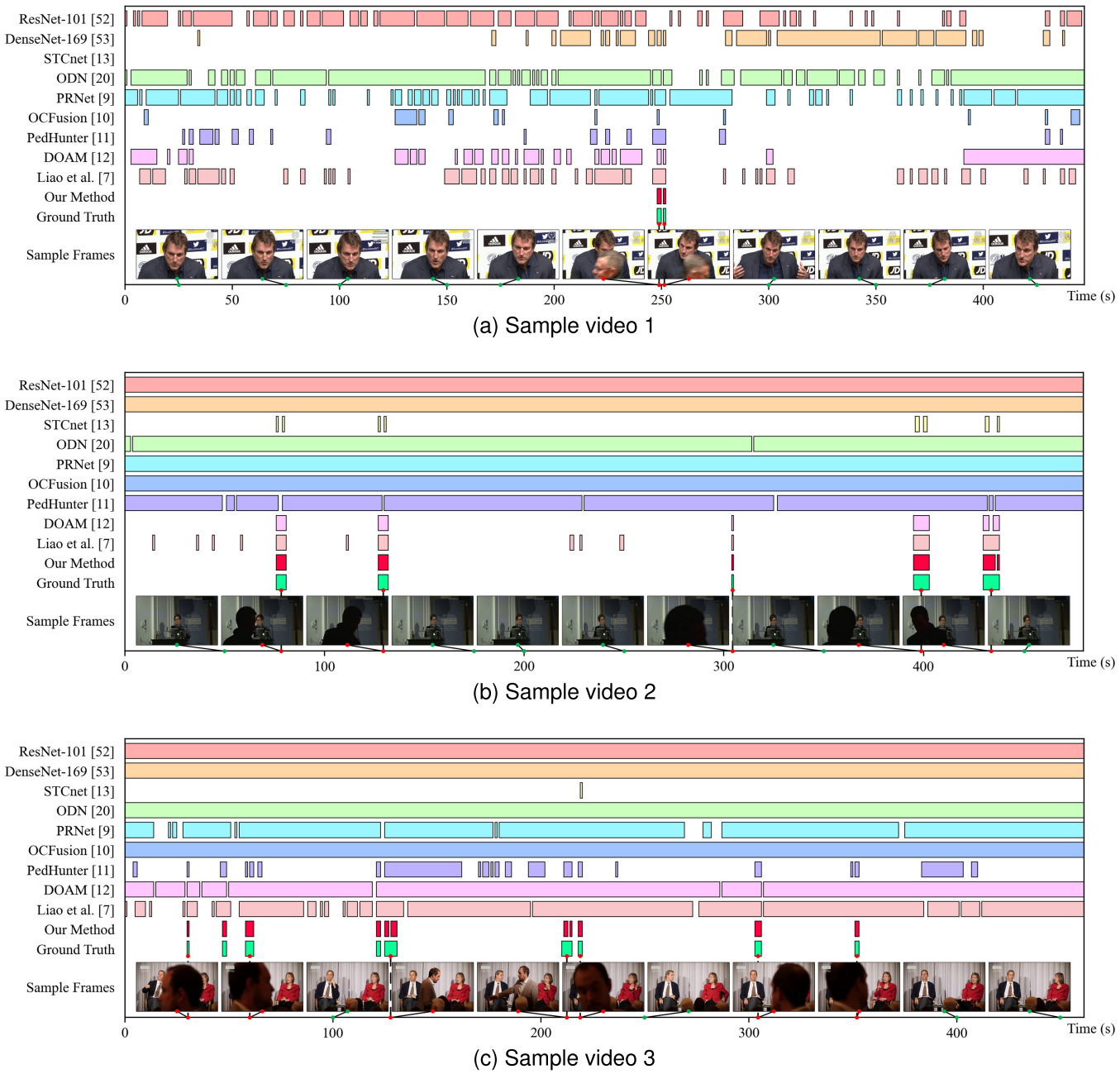

Fig. 9. Visualization results of shot occlusion clips predicted by different methods on sample videos. The different color blocks represent the time extent of shot occlusion predicted by different methods.

frame, and then compares it with the average of the feature map of the entire video for similarity. Frames whose similarity is lower than the threshold are regarded as occluded frames. This design stems from the idea that an occluded picture will be very different from a normal picture, which accounts for most of the video. The False Positive Rate$_s$ of this detection method is very competitive and does not exceed 6% on the entire VSOD dataset, but its Recall$_e$ and Recall$_s$ are the lowest among all comparison methods. This may be because the information in the feature map is too abstract, resulting in the loss of information changes caused by small and medium occluders. The extremely poor Recall$_e$ makes it unable to meet the demands of editors.

As the strongest competitor in the test set of the VSOD v1.0 dataset, the state-of-the-art shot occlusion detection algorithm [7] achieves the highest Recall$_e$ of 99.75% on the VSOD v2.0 dataset, but its False Positive Rate$_s$ is also as high as 34.9%. Since the VSOD v1.0 dataset is dominated by shot occlusion clips, the data-driven neural network is affected by this dataset and is more sensitive to shot occlusion. The Precison$_s$ of this algorithm on the VSOD v2.0 dataset is only 23.09%, which means that nearly 77% of its predicted occlusions are false positives. Although the excellent Recall$_e$ of this algorithm seems to meet the needs of the editors, the high False Positive Rate$_s$ and low Intersection-over-Union$_s$ will make the workload of editors substantial.

TABLE IV
COMPARISON OF SHOT OCCLUSION DETECTION PERFORMANCE BEFORE AND AFTER FINE TUNING ON THE VSOD V2.0 DATASET

Methods	Fine Tuning?	Recall$_e$	Intersection-over-Union$_s$	Accuracy$_s$	False Positive Rate$_s$	Recall$_s$	Precision$_s$	F1 Score$_s$
ResNet-101 [52]	✗	0.9750	0.0413	0.0901	0.9491	0.9696	0.0413	0.0704
DenseNet-169 [53]	✗	0.9458	0.1462	0.3305	0.7033	0.9199	0.1789	0.1961
STCnet [13]	✗	0.2637	0.1223	0.9477	**0.0143**	0.1633	0.2335	0.1596
ODN [20]	✗	0.9917	0.1159	0.3447	0.6849	**0.9855**	0.1214	0.1694
PRNet [9]	✗	0.9833	0.1509	0.4761	0.5456	0.9007	0.1835	0.2105
OCFusion [10]	✗	0.8686	0.1855	0.5865	0.4165	0.7648	0.2662	0.2465
PedHunter [11]	✗	0.7625	0.2437	0.6634	0.3393	0.6818	0.3339	0.3067
DOAM [12]	✗	0.9783	0.3537	0.7180	0.2844	0.8496	0.4121	0.4272
Liao et al. [7]	✗	**1.0000**	0.1964	0.6778	0.3195	0.9374	0.2302	0.2590
Our Method	-	0.9572	**0.6148**	**0.9482**	0.0319	0.7241	**0.8433**	**0.7041**
ResNet-101 [52]	✓	0.0000	0.0000	0.9596	**0.0000**	0.0000	0.0000	0.0000
DenseNet-169 [53]	✓	0.8193	0.1110	0.4817	0.5177	0.6910	0.2488	0.1649
STCnet [13]	✓	0.5381	0.1828	0.8720	0.1023	0.3633	0.3315	0.2508
ODN [20]	✓	0.4342	0.1849	0.9134	0.0551	0.2667	0.4068	0.2470
PRNet [9]	✓	0.6533	0.2925	0.8844	0.0864	0.4478	0.5305	0.3655
OCFusion [10]	✓	0.6255	0.3035	0.8417	0.1425	0.4773	0.5099	0.3698
PedHunter [11]	✓	0.2868	0.0913	0.9491	0.0181	0.1247	0.3638	0.1408
DOAM [12]	✓	0.6404	0.4304	0.9665	0.0006	0.4447	0.7308	0.5063
Liao et al. [7]	✓	0.8726	0.5903	**0.9668**	0.0044	0.6268	**0.8659**	0.6726
Our Method	-	**0.9572**	**0.6148**	0.9482	0.0319	**0.7241**	0.8433	**0.7041**

In addition, we randomly divide the VSOD v2.0 dataset into a training set and a test set with a ratio of 8:2 and fine-tune the different comparison methods on this training set. The performance comparison of the shot occlusion detection algorithms before and after fine-tuning on the VSOD v2.0 dataset is shown in Table IV. According to the experimental results, driven by the training set of the VSOD v1.0 dataset, which is dominated by occluded clips, most of the neural network models have Recall$_e$ and Recall$_s$ values of more than 90%, but their false positive rate is also high. After fine-tuning on the training set of the VSOD v2.0 dataset, which is dominated by normal clips, the false positive rate of the data-driven methods decreases significantly, but their Recall$_e$ is also lower than 90%. This verifies that the performance of neural network-based methods is directly affected by the dataset itself, and the robust model can only be driven by abundant and high-quality data. Due to the complex and diverse video shooting environment, it is difficult to collect sufficient data to drive the neural network to achieve the ideal video shot occlusion detection algorithm with both a high event recall and low false positives.

To this end, we provide a novel idea for constructing an ideal video shot occlusion detection algorithm, that is, to detect shot occlusion by capturing abnormal fluctuations of frame depth information caused by occluders. Experimental results on different versions of the VSOD dataset show that our algorithm has not only a Recall$_e$ of more than 95%, but also a False Positive Rate$_s$ of less than 3%. In addition, the Accuracy$_s$ and Precison$_s$ of our shot occlusion detection algorithm are the best on the entire VSOD dataset, indicating that the shot occlusion predictions made by this algorithm are more reliable than those of other detection algorithms. To more intuitively demonstrate the convenience our algorithm brings to video editors, we show the results of shot occlusion clips predicted by different detection methods. As shown in Fig. 9, our shot occlusion detection algorithm can greatly reduce the workload of video editors more than other detection algorithms by accurately localizing occlusion events in videos.

D. Comparison With the Baseline

In addition to state-of-the-art occlusion detection methods, we also design a large number of baseline methods for performance comparison. Since normal frames usually occupy the majority of the time in the regular video, we use the cosine similarity (CS), histogram similarity (HS), linear correlation coefficient (LCC), and Spearman's rank order correlation coefficient (SROCC) between the frame and the overall video to detect shot occlusion. Furthermore, we design baseline methods that rely on evaluating the similarity and correlation of neighboring frames to identify shot occlusion. As shown in Table II and Table III, the comprehensive performance of these baseline methods on the entire VSOD dataset is far inferior to that of the proposed video shot occlusion detection method, further demonstrating the superiority of our algorithm. Moreover, baseline methods have improved performance compared to the fixed occlusion threshold of 0.9 after using our dynamic occlusion threshold selection algorithm, proving the effectiveness of this algorithm.

E. Ablation Studies

1) Depth Estimation: Table V presents the impact of different depth estimation methods on the performance of the proposed shot occlusion detection algorithm. Although these state-of-the-art depth estimation methods [55], [56], [57] outperform the DPT [8] on some mainstream depth datasets [28], [29], they almost degrade the performance of the proposed algorithm across the board. Due to the variety of scenarios oriented by automatic video editing tasks, depth estimation methods with excellent performance in a particular scenario

TABLE V
THE IMPACT OF DIFFERENT DEPTH ESTIMATION METHODS

Methods	Dataset	$Recall_e$	$Intersection\text{-}over\text{-}Union_s$	$Accuracy_s$	$False\ Positive\ Rate_s$	$Recall_s$	$Precision_s$	$F1\ Score_s$
NeWCRFs [55]	VSOD v1.0 [7]	0.8450	0.6769	0.7320	0.0771	0.7086	0.8078	0.7347
GuideDepth [56]		0.9950	0.8105	0.8282	0.2044	**0.8988**	0.8973	0.8733
Jun et al. [57]		**1.0000**	0.8408	0.8589	0.1091	0.8950	0.9407	0.8956
DPT [8]		**1.0000**	**0.8623**	**0.8806**	**0.0214**	0.8761	**0.9833**	**0.9105**
NeWCRFs [55]	VSOD v2.0	0.8840	0.2509	0.8897	0.0882	0.5684	0.4134	0.3465
GuideDepth [56]		0.9402	0.1941	0.8390	0.1429	0.6588	0.2940	0.2809
Jun et al. [57]		0.9561	0.2720	0.9017	0.0777	0.6275	0.4145	0.3746
DPT [8]		**0.9588**	**0.5128**	**0.9531**	**0.0252**	**0.6623**	**0.7662**	**0.6190**

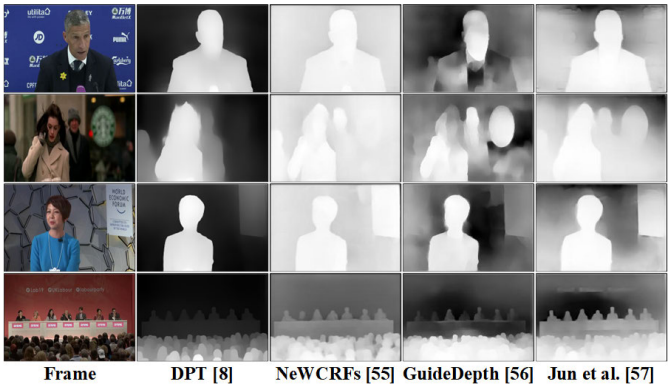

Fig. 10. Examples of depth maps predicted by different depth estimation methods [8], [55], [56], [57]. The brighter area is closer to the shot.

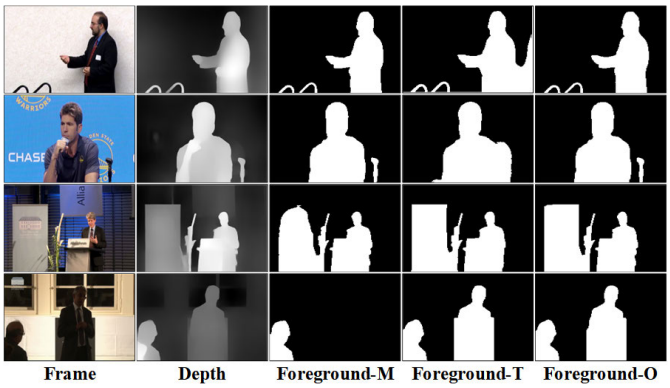

Fig. 11. Examples of depth foreground segmentation by different depth threshold selection methods. Foreground-M, Foreground-T, and Foreground-O represent the depth foreground segmented by the middle value of the depth interval, the depth foreground segmented by the Triangle algorithm, and the depth foreground segmented by the Otsu algorithm, respectively.

may not be competent for this task. The DPT is driven by the largest monocular depth estimation training set (1.4 million images), which is composed of numerous mainstream depth datasets and private datasets. This method has superior robustness and can provide reliable depth maps for the proposed shot occlusion detection algorithm. Fig. 10 shows the prediction results of different depth estimation methods in this task scenario, and the depth maps predicted by DPT are more accurate than those predicted by other state-of-the-art methods.

2) Depth Threshold: We experimentally verify the impact of three different depth threshold selection methods, namely, the middle value of the depth interval, the Triangle algorithm [58], and the Ostu algorithm [48], on the proposed

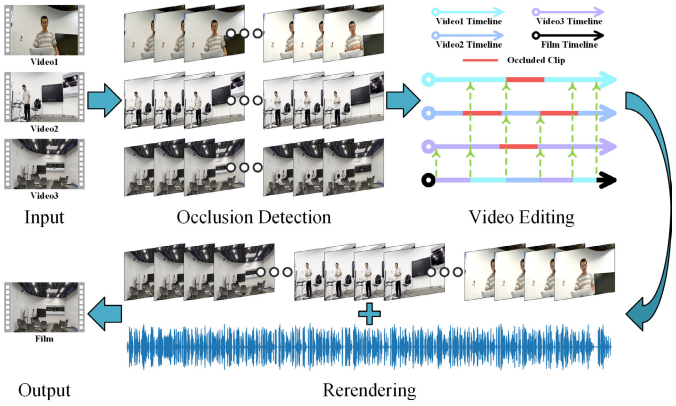

Fig. 12. The architecture of automatic video editing system.

shot occlusion detection algorithm. The Triangle algorithm and Ostu algorithm are both implemented by the interface provided by OpenCV [59]. The experimental results in Table VI indicate that the algorithm based on the middle value performs better in the recall, while the algorithm based on the Otsu has fewer false positives. The reason may be as shown in Fig. 11, where the foreground segmented by the middle value of the depth interval is not as complete as the foreground segmented by the Otsu algorithm in complex scenes, and the incomplete foreground causes the *SoU* to be more volatile. This makes the algorithm based on the middle value more sensitive, but the false positive rate also increases accordingly. In contrast, the algorithm based on the Otsu can more accurately segment the foreground and have a lower false positive rate when having a high recall of occlusion events, which is a better choice.

F. Automatic Video Editing System

Since our algorithm is oriented towards video editing, we construct a simple automatic video editing system for multi-camera shooting scenes to make our work more complete. Fig. 12 shows its architecture. First, the system uses the proposed algorithm to identify the shot occlusion time period in each video. Second, the system performs video editing. When recording content along the timeline of the film, it randomly selects videos that do not have shot occlusion during the corresponding time period. After recording the content of the currently selected video for 10 to 20 seconds, the system will randomly select other normal videos to continue recording. Finally, the system combines video and audio to rerender a high-quality film that does not contain shot occlusion clips.

TABLE VI
THE IMPACT OF DIFFERENT DEPTH THRESHOLD SELECTION METHODS

Methods	Dataset	Recall_e	$\text{Intersection-over-Union}_s$	Accuracy_s	$\text{False Positive Rate}_s$	Recall_s	Precision_s	F1 Score_s
Middle (127.5)	VSOD v1.0 [7]	**1.0000**	**0.9006**	**0.9114**	0.0268	**0.9164**	**0.9842**	**0.9369**
Triangle [58]		0.9900	0.7962	0.8216	0.1363	0.8594	0.9180	0.8633
Otsu [48]		**1.0000**	0.8623	0.8806	**0.0214**	0.8761	0.9833	0.9105
Middle (127.5)	VSOD v2.0	**0.9806**	0.4605	0.9098	0.0736	**0.7427**	0.5821	0.5570
Triangle [58]		0.8883	0.1930	0.7761	0.2091	0.6055	0.3040	0.2696
Otsu [48]		0.9588	**0.5128**	**0.9531**	**0.0252**	0.6623	**0.7662**	**0.6190**

VI. LIMITATION AND FUTURE WORK

Although our shot occlusion detection algorithm performs well on the entire VSOD dataset, it still has limitations. When the occluder moves slowly or even stops after breaking into the shot, SoU will return to a higher value in advance, resulting in missed detection. This explains why the Recall_e of our algorithm is as high as 95%, but the Recall_s is less than 90%. The difference between Recall_e and Recall_s indicates that although the proposed detection algorithm is capable of identifying a majority of occlusion events, the completeness of the detected occlusion events remains a need for improvement. In addition, the violent movement of the protagonist also leads to a significant fluctuation of SoU, resulting in false detection. It may be insufficient to rely solely on depth information for achieving optimal performance in detecting complex occlusions in video shots. Therefore, introducing an additional module to the shot occlusion detection algorithm to identify the protagonist in the video as supplementary information may be helpful for improving the integrity of the detected occlusion events. How to further improve the Recall_s of the shot occlusion detection algorithm while maintaining a low false positive rate will be the focus of our follow-up research. In addition, further investigative work also includes improving the automatic video editing system by enriching the abnormal situations (e.g. blur, jitter, and distortion) during data preprocessing and optimizing the video editing rules.

VII. CONCLUSION

In this work, we conduct an in-depth study on challenging video shot occlusion detection and achieve inspiring results. Due to the diversity and complexity of shot occlusion, training a neural network with limited manually-collected data may not constitute an optimal solution for video shot occlusion detection. Therefore, we propose a non-data-driven video shot occlusion detection algorithm to assist video editors. Unlike the data-driven methods, our algorithm detects occlusion according to the commonality of abnormal fluctuations in frame depth information caused by the occluder, thus possessing good generalization ability, even without occlusion training data. Furthermore, to accurately evaluate the comprehensive performance of detection methods, we expand the VSOD dataset released from our conference publication, which is the first dataset for video shot occlusion detection. Experimental results on the VSOD dataset demonstrate that our algorithm outperforms other state-of-the-art occlusion detection methods, as well as self-designed baseline methods, in terms of comprehensive performance. Even on different types of datasets, our algorithm can maintain more than a recall of occlusion events of 95%, while the false positive rate is less than 3%, demonstrating good generalization ability. We believe that our contributions are beneficial in promoting the development of abnormal detection in automatic video editing.

REFERENCES

[1] A. Pardo, F. C. Heilbron, J. L. Alcázar, A. Thabet, and B. Ghanem, "Learning to cut by watching movies," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6838–6848.

[2] M. Leake, A. Davis, A. Truong, and M. Agrawala, "Computational video editing for dialogue-driven scenes," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–130, 2017.

[3] R. L. Guimarães, P. Cesar, D. C. A. Bulterman, V. Zsombori, and I. Kegel, "Creating personalized memories from social events: Community-based support for multi-camera recordings of school concerts," in *Proc. 19th ACM Int. Conf. Multimedia*, Nov. 2011, pp. 303–312.

[4] A. Truong, F. Berthouzoz, W. Li, and M. Agrawala, "QuickCut: An interactive tool for editing narrated video," in *Proc. 29th Annu. Symp. User Interface Softw. Technol.*, Oct. 2016, pp. 497–507.

[5] M. Wang, G.-W. Yang, S.-M. Hu, S.-T. Yau, and A. Shamir, "Write-a-video: Computational video montage from themed text," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–13, Dec. 2019.

[6] X.-S. Hua, L. Lu, and H.-J. Zhang, "Optimization-based automated home video editing system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 5, pp. 572–583, May 2004.

[7] J. Liao et al., "Occlusion detection for automatic video editing," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 2255–2263.

[8] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12159–12168.

[9] X. Song, K. Zhao, W.-S. Chu, H. Zhang, and J. Guo, "Progressive refinement network for occluded pedestrian detection," in *Proc. Eur. Conf. Comput. Vis.* Glasgow, U.K.: Springer, Aug. 2020, pp. 32–48.

[10] J. Lazarow, K. Lee, K. Shi, and Z. Tu, "Learning instance occlusion for panoptic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10717–10726.

[11] C. Chi, S. Zhang, J. Xing, Z. Lei, S. Z. Li, and X. Zou, "PedHunter: Occlusion robust pedestrian detector in crowded scenes," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 10639–10646.

[12] Y. Wei, R. Tao, Z. Wu, Y. Ma, L. Zhang, and X. Liu, "Occluded prohibited items detection: An X-ray security inspection benchmark and de-occlusion attention module," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 138–146.

[13] R. Hou, B. Ma, H. Chang, X. Gu, S. Shan, and X. Chen, "VRSTC: Occlusion-free video person re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7176–7185.

[14] X. Zhang, Y. Yan, J. Xue, Y. Hua, and H. Wang, "Semantic-aware occlusion-robust network for occluded person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 7, pp. 2764–2778, Jul. 2021.

[15] Q. Xu, J. Pan, and Y. Qian, "Learning an occlusion-aware network for video deblurring," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4312–4323, Jul. 2022.

[16] C. Zhou and J. Yuan, "Occlusion pattern discovery for object detection and occlusion reasoning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 2067–2080, Jul. 2020.

[17] H. Pham, Z. Dai, Q. Xie, and Q. V. Le, "Meta pseudo labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11552–11563.

[18] X. Dai et al., "Dynamic head: Unifying object detection heads with attentions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7369–7378.

[19] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 801–818.

[20] M. Zhu, D. Shi, M. Zheng, and M. Sadiq, "Robust facial landmark detection via occlusion-adaptive deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3481–3491.

[21] J. Lahoud and B. Ghanem, "2D-driven 3D object detection in RGB-D images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4632–4640.

[22] H. Sheng, R. Cong, D. Yang, R. Chen, S. Wang, and Z. Cui, "UrbanLF: A comprehensive light field dataset for semantic segmentation of urban scenes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 11, pp. 7880–7893, Nov. 2022.

[23] R. S. Pahwa, J. Lu, N. Jiang, T. T. Ng, and M. N. Do, "Locating 3D object proposals: A depth-based online approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 3, pp. 626–639, Mar. 2018.

[24] W. Chen, Z. Fu, D. Yang, and J. Deng, "Single-image depth perception in the wild," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 730–738.

[25] X. Meng, C. Fan, Y. Ming, and H. Yu, "CORNet: Context-based ordinal regression network for monocular depth estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4841–4853, Jul. 2022.

[26] R. Li, P. Ji, Y. Xu, and B. Bhanu, "MonoIndoor++: Towards better practice of self-supervised monocular depth estimation for indoor environments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 2, pp. 830–846, Feb. 2023.

[27] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1623–1637, Mar. 2022.

[28] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2012, pp. 746–760.

[29] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[30] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.

[31] K. Xian et al., "Monocular relative depth perception with web stereo data supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 311–320.

[32] C. Wang, S. Lucey, F. Perazzi, and O. Wang, "Web stereo video supervision for depth prediction from dynamic scenes," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2019, pp. 348–357.

[33] W. Chen, S. Qian, and J. Deng, "Learning single-image depth from videos using quality assessment networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5597–5606.

[34] Y. Kim, H. Jung, D. Min, and K. Sohn, "Deep monocular depth estimation via integration of global and local predictions," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 4131–4144, Aug. 2018.

[35] Z. Li and N. Snavely, "MegaDepth: Learning single-view depth prediction from Internet photos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2041–2050.

[36] T. Schöps et al., "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2538–2547.

[37] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2012, pp. 611–625.

[38] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The ApolloScape open dataset for autonomous driving and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2702–2719, Oct. 2020.

[39] Q. Wang, S. Zheng, Q. Yan, F. Deng, K. Zhao, and X. Chu, "IRS: A large synthetic indoor robotics stereo dataset for disparity and surface normal estimation," 2019, *arXiv:1912.09678*.

[40] W. Wang et al., "TartanAir: A dataset to push the limits of visual SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 4909–4916.

[41] K. Xian, J. Zhang, O. Wang, L. Mai, Z. Lin, and Z. Cao, "Structure-guided ranking loss for single image depth prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 608–617.

[42] Y. Yao et al., "BlendedMVS: A large-scale dataset for generalized multi-view stereo networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1787–1796.

[43] C. Wu, J. Wang, M. Hall, U. Neumann, and S. Su, "Toward practical monocular indoor depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3804–3814.

[44] Q. Wang, Z. Li, D. Salesin, N. Snavely, B. Curless, and J. Kontkanen, "3D moments from near-duplicate photos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3896–3905.

[45] J. Peng, Z. Cao, X. Luo, H. Lu, K. Xian, and J. Zhang, "BokehMe: When neural rendering meets classical rendering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16262–16271.

[46] E. Arnold et al., "Map-free visual relocalization: Metric pose relative to a single image," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 690–708.

[47] P. Z. Ramirez, F. Tosi, M. Poggi, S. Salti, S. Mattoccia, and L. D. Stefano, "Open challenges in deep stereo: The booster dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 21136–21146.

[48] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[49] X. P. Burgos-Artizzu, P. Perona, and P. Dollár, "Robust face landmark estimation under occlusion," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1513–1520.

[50] J. Qi et al., "Occluded video instance segmentation: A benchmark," *Int. J. Comput. Vis.*, vol. 130, no. 8, pp. 2022–2039, Aug. 2022.

[51] J. Miao, Y. Wu, P. Liu, Y. Ding, and Y. Yang, "Pose-guided feature alignment for occluded person re-identification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 542–551.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[53] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[54] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.

[55] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan, "Neural window fully-connected CRFs for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3906–3915.

[56] M. Rudolph, Y. Dawoud, R. Güldenring, L. Nalpantidis, and V. Belagiannis, "Lightweight monocular depth estimation through guided decoding," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 2344–2350.

[57] J. Jun, J.-H. Lee, C. Lee, and C.-S. Kim, "Depth map decomposition for monocular depth estimation," in *Proc. Eur. Conf. Comput. Vis.* Tel Aviv, Israel: Springer, Oct. 2022, pp. 18–34.

[58] G. W. Zack, W. E. Rogers, and S. A. Latt, "Automatic measurement of sister chromatid exchange frequency," *J. Histochemistry Cytochemistry*, vol. 25, no. 7, pp. 741–753, Jul. 1977.

[59] G. Bradski et al., "OpenCV," *Dr. Dobb's J. Softw. Tools*, vol. 3, no. 2, pp. 1–81, 2000.

Junhua Liao (Graduate Student Member, IEEE) received the B.Eng. degree in computer science and technology from Sichuan University, Chengdu, China, in 2020, where he is currently pursuing the Ph.D. degree in computer science and technology. His research interests include multimedia, human-centered computing, and medical image analysis.

Haihan Duan (Student Member, IEEE) received the B.Eng. degree in computer science and technology from East China Normal University, Shanghai, China, in 2017, the M.Eng. degree in software engineering from Sichuan University, Chengdu, China, in 2020, and the Ph.D. degree in computer and information engineering from The Chinese University of Hong Kong, Shenzhen, China, in 2023. He is currently a Visiting Student with the Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates. His research interests include multimedia, Web3, metaverse, blockchain, and human-centered computing.

Wanbing Zhao received the B.Eng. degree in Internet of Things engineering from Sichuan University, Chengdu, China, in 2019, where he is currently pursuing the Ph.D. degree in computer science and technology. His research interests include few-shot learning, computer vision, and machine learning.

Kanghui Feng received the B.Eng. degree in software engineering from Sichuan University, Chengdu, China, in 2021, where he is currently pursuing the Ph.D. degree in software engineering. His research interests include medical image analysis and deep learning.

Yanbing Yang (Member, IEEE) received the B.E. and M.E. degrees from the University of Electronic Science and Technology of China, Chengdu, China, and the Ph.D. degree in computer science and engineering from Nanyang Technological University, Singapore. He is currently an Associate Research Professor with the College of Computer Science, Sichuan University, Chengdu. His research interests include the IoT, visible light communication, visible light sensing, and also their applications.

Liangyin Chen (Member, IEEE) received the Ph.D. degree from the College of Computer Science, Sichuan University, China, in 2008. He joined the School of Computer Science, Sichuan University and he has been a Professor since 2014. He is currently the Director of the Sichuan Big Data Analysis and Fusion Application Technology Engineering Laboratory. From 2009 to 2010, he was a Visiting Researcher with the University of Minnesota, under the supervision of Prof. Tian He. His research interests include wireless sensor networks, the Intelligent Internet of Things, the IoT security, industrial internet, blockchains, and big data. He has authored or coauthored more than 100 papers, many of which were published in premier network journals and conferences, such as IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, *Knowledge-Based Systems* (KBS), IEEE International Conference on Computer Communications (INFOCOM), ACM International Conference on Multimedia (MM), IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (Secon), and ACM Conference on Embedded Networked Sensor Systems (SenSys).