2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Mixed Frame-/Event-Driven Fast Pedestrian Detection

Zhuangyi Jiang[1], Pengfei Xia[2], Kai Huang[3], Walter Stechele[2], Guang Chen[4], Zhenshan Bing[1] and Alois Knoll[1]

***Abstract*— Pedestrian detection has attracted enormous research attention in the field of Intelligent Transportation System (ITS) due to that pedestrians are the most vulnerable traffic participants. So far, almost all pedestrian detection solutions are based on the conventional frame-based camera. However, they cannot perform very well in scenarios with bad light condition and high-speed motion. In this work, a Dynamic and Active Pixel Sensor (DAVIS), whose two channels concurrently output conventional gray-scale frames and asynchronous low-latency temporal contrast events of light intensity, was first used to detect pedestrians in a traffic monitoring scenario. Data from two camera channels were fed into Convolutional Neural Networks (CNNs) including three YOLOv3 models and three YOLO-tiny models to gather bounding boxes of pedestrians with respective confidence map. Furthermore, a confidence map fusion method combining the CNN-based detection results from both DAVIS channels was proposed to obtain higher accuracy. The experiments were conducted on a custom dataset collected on TUM campus. Benefiting from the high speed, low latency and wide dynamic range of the event channel, our method achieved higher frame rate and lower latency than those only using a conventional camera. Additionally, it reached higher average precision by using the fusion approach.**

I. INTRODUCTION

With the rapid development of the automobile industry, especially the autonomous driving technologies, road traffic safety issue is becoming increasingly prominent. According to the United Nation Improving Global Road Safety report released in 2017, road traffic accidents lead to more than 1.3 million deaths per year worldwide. In particular, pedestrians, motorcyclists and bicyclists are involved in more than half of road traffic deaths. Hence, pedestrian detecting and tracking approaches for collision avoidance have turned into a research hotspot in Intelligent Transportation System (ITS) in the last years.

Up to now, numerous methods for pedestrian detection have been proposed. Most of them were based on time-of-flight sensors [1] and imaging sensors. Various sensing data gathered by these devices were processed by detection algorithms to estimate the positions of pedestrians. By applying a fast and precise detection method, positions of pedestrians can be obtained and informed to drivers or unmanned vehicles immediately to avoid possible collisions with the pedestrians.

[1]Zhuangyi Jiang, Zhenshan Bing and Alois Knoll are with Department of Informatics, Technical University of Munich, Germany. {jiangz, bing, knoll}@in.tum.de [2]Pengfei Xia and Walter Stechele are with Department of Electrical and Computer Engineering, Technical University of Munich, Germany. xiapengfei3@163.com, Walter.Stechele@tum.de [3]Kai Huang is with Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-sen University), Ministry of Education, and School of Data and Computer Science, Sun Yat-sen University, China. huangk36@mail.sysu.edu.cn [4]Guang Chen is with School of Automotive Studies, Tongji University, China. guangchen@tongji.edu.cn

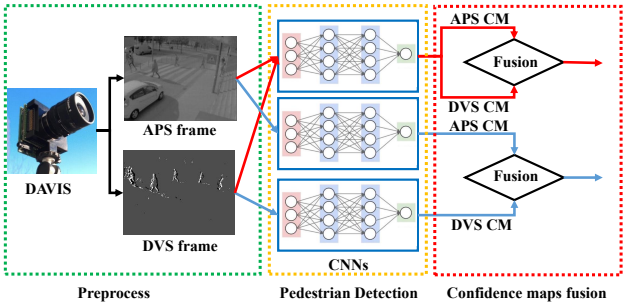


Fig. 1. Framework of mixed frame-/event-driven pedestrian detection.

Nevertheless, all the existing pedestrian detection systems have their own drawbacks. The systems based on the time-of-flight sensor that involves with Lidar are still expensive. Although some methods and algorithms can achieve high accuracy [2][3], the systems based on the imaging sensor still exist inherent problems such as data redundant, high latency and being sensitive to light condition.

In order to achieve high frame rate, low latency and wide dynamic range, the Dynamic and Active Pixel Sensor (DAVIS), a promising vision sensor, has been introduced into ITS for vehicle and pedestrian detection [4]. DAVIS camera is composed by two channels, the Active Pixel Sensor (APS) channel and the Dynamic Vision Sensor (DVS) channel. APS and DVS channel outputs the conventional gray-scale frames and asynchronous events recording the illumination changes, respectively. By mixing both channels in the framework of CNN-based detection, the performance of the pedestrian detection can be improved. The system is able to gain high accuracy, high frame rate and good robustness simultaneously.

In this work, a CNN-based pedestrian detection approach mixing the frame and the event channel of DAVIS camera was first proposed. Two kinds of You Only Look Once (YOLO) networks were trained for detecting pedestrian in two DAVIS channels, respectively. Furthermore, a confidence map fusion method was designed to acquire uniformed and more precise results. We compared the performance of detection algorithm implemented with YOLOv3 and YOLO-tiny, further compared the results with another CNN network trained by the mixed sensor channel. Until now, the DVS channel of DAVIS, a neuromorphic vision sensor, were rarely used in ITS. Hence, our frame and event channel mixed pedestrian detection approach is of great significance to explore the application of neuromorphic vision sensor in detecting pedestrian in ITS.

The rest of this paper is organized as follows. In section II, we listed the state-of-the-art methods of the pedestrian

978-1-5386-6027-0/19/$31.00 ©2019 IEEE

detection. In section III, we introduced the framework and the methodology of the mixed frame-/event-driven pedestrian detection. The experiment results were shown and discussed in section IV. In section V, we draw the conclusion and point out the possible further work.

II. RELATED WORK

In recent years, pedestrian detection remains a popular research topic due to its extensive applications, such as collision avoidance and pedestrian intention prediction. Articles in this field are mainly classified into two categories according to the utilization of vision sensor is conventional or neuromorphic.

A. Pedestrian detection by conventional vision sensors

Features-based model and deep learning model are two widely used models for vision-based pedestrian detection [5]. The difference of them is whether the model extracts features by using specified feature descriptor.

For features-based model, the choice of feature descriptor plays a more important role in improving the quality of pedestrian detection instead of the classifier choice (e.g. SVM or decision forest) [6]. The most effective feature descriptors are Histograms of Oriented Gradients (HOG) [7][2], Haar-like features [8][9], Local Binary Patterns (LBP) [10], texture features [11], Integral Channel Features (ICF) [12][13] as well as its variants [6][14].

Different from the feature-based model, deep learning models extract features at different layers of deep learning architecture automatically instead of manually. The most famous deep learning model is Convolution Neural Networks (CNN). In [15], CNN is employed as classifier to detect pedestrians. Compared with the approach to SVM with Haar features, CNN can achieve higher accuracy and lower false positive rate (FPR). The deep learning frameworks based on CNN usually include single shot detector (SSD) [16], Region CNN (R-CNN) [17] and its variations [18][19], YOLO [20] and its variations [21].

Despite pedestrian detection approaches based on conventional vision sensors have achieved significant improvement, a ten-fold improvement can still be made before reaching human performance [22]. Additionally, they also cannot overcome the image blur and low frame rate in the scenarios with bad light condition and high-speed motion.

B. Pedestrian detection by neuromorphic vision sensors

Neuromorphic vision sensor, also named event-based camera, is a novel bio-inspired silicon retina commonly used in motion detection, object detecting and tracking. In previous work, it was successfully applied to detecting vehicles while no more research regarding on pedestrian detection has been reported, as we know.

In [23], a Spiking Neural Network (SNN) was utilized to count the number of cars only with a simple, fully local Spike-Timing-Dependent Plasticity (STDP) rule. However, the use of spikes restricts the application range of SNNs and so far no obvious ways are available to use SNNs to generate bounding box in object detection. An alternative way to address the outputs of DVS is to convert events to frames and then use frame-based detection methods. In our previous work [4], the neuromorphic vision sensor was firstly introduced into intelligent transportation systems and applied clustering methods to detect and track vehicles from frames which were reconstructed by events.

Due to the similarity between a conventional frame and an accumulating frame of events, CNNs are more and more employed to deal with the object detection problem by event-based cameras. [24] used a DVS to predict steering angle of a self-driving car by adapting a convolutional architecture to the events. It was proved that pre-trained CNN still has good performance on frames that accumulated by events. DVAIS is also performed in object detection with a lack of labeled ground truth. In [25], images from APS channel of DAVIS passed through a normal CNN to get pseudo-labels of vehicles which were later used as targets in a supervised learning process based on YOLO for event-based images from DVS channel.

In a predator/prey scenario, DAVIS (both the APS and DVS channel) and CNN architectures were utilized to detect the prey and steering a predator robot to chase the prey robot [26][27]. In [26], regions of interest (ROI) were generated by clustering the event information. Then, target was detected by applying CNN on output of APS channel in ROI. High accuracy and low computational cost were achieved simultaneously via this method.

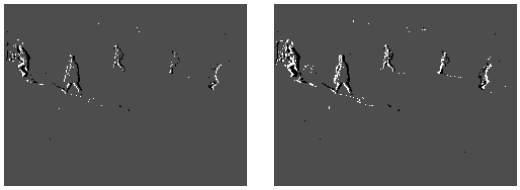
III. METHODOLOGY

A. Event-based Vision Sensor

Dynamic vision sensor (DVS) is the first commercial neuromorphic vision sensor, also named silicon retina [28]. Different from the conventional passive and active pixel sensors, DVS outputs responses to light intensity changes in the order of microseconds as an asynchronous events. Only if the difference between current brightness and last sampled brightness exceeds a certain threshold, a pixel-level spike is transmitted. The spikes are encoded as Address Event Representations (AERs). Each spike can be represented as a tuple (x, y, t, p). (x, y) identifies the position of the triggered pixel, t is the timestamp corresponding to the triggering time and p is the polarity of event, +1 indicates the positive spike while -1 indicates the negative one.

Dynamic and Active Pixel Vision Sensor (DAVIS) is a vision sensor combining an APS and a DVS. Therefore, it has two channels to generate event streams and frame streams concurrently. For each DAVIS pixel, the APS circuit and DVS circuit share a single photodiode, but the APS channel is independent to the DVS channel. The active pixel frames have a uniform exposure because of a synchronous global shutter. A DAVIS240 with 240×180 pixels was applied in this work. It has a wide dynamic range of 130 dB. The DVS channel can generate 12 million events per second and the APS channel can output maximal 35 frames per second [29].

Although the DVS channel can output events with high dynamic range, low latency and sparse output, the lack of absolute brightness information introduces a great challenge for object detection and classification. The frames generated from APS channel can address this shortage. Hence, DAVIS combined with DVS channel and APS channel has a more promising range of application than DVS. The output of APS channel can be used for detection and classification, while the output of DVS channel is suitable for detecting and tracking



(a) 10 ms interval (b) 20 ms interval

Fig. 2. DVS frames reconstruction by fixed time interval method

the moving objects. In this work, outputs of both channels of DAVIS were performed for pedestrian detection to achieve a fast and accurate detection algorithm.

B. Frame reconstruction

As mentioned above, converting events to frames is the prerequisite for using CNN. Thereby, the events deriving from the DVS channel need to be accumulated as frames. Based on the generation mechanism of events, there exist three frame reconstruction methods. They are fixed events number, leaky integrator, and fixed time interval method, respectively.

In this work, the frames were reconstructed at a fixed frame duration, such as 10ms, 20ms, etc. In each time interval, the positive events were plotted as white pixels and the negative events were plotted as black pixels with a gray background. The reconstructed frames with different time intervals were shown in Fig. 2. It is clear that accumulation with longer time results in more intact objects in frames.

In particular, the output frequency of APS channel was 25fps, it means that an APS frame was generated every 40 ms. Therefore, 10 ms and 20 ms intervals were selected to reconstruct DVS frames so that higher frequency of detection could be achieved.

C. Pedestrian detection based on CNN

YOLO is a novel convolutional neural network (CNN), which considers object detection as a regression problem. Instead of adopting sliding windows approach and region proposal methods, YOLO uses the whole images to train detection model which can realize real-time prediction of bounding boxes and their class probabilities simultaneously.

YOLO-tiny is a tiny model with 16 convolutional layers to realize extremely fast object detection in sacrifice of little accuracy. Another YOLO model is YOLOv3, the most advanced version of YOLO by far. Compared with YOLO-tiny, YOLOv3 is a larger network with more convolutional layers to extract features. By applying several optimization approaches, YOLOv3 improves accuracy significantly, while it is more time-consuming and requires more computing resources.

In this work, both YOLOv3 and YOLO-tiny models were used to detect pedestrians in both APS frames and DVS frames.

D. Confidence Map Fusion

YOLO network can generate the confidence maps, in which each pixel value represents the probability of this pixel belonging to an object. All pixel values are normalized in the range of [0,255]. Fig. 3 showed an example of the

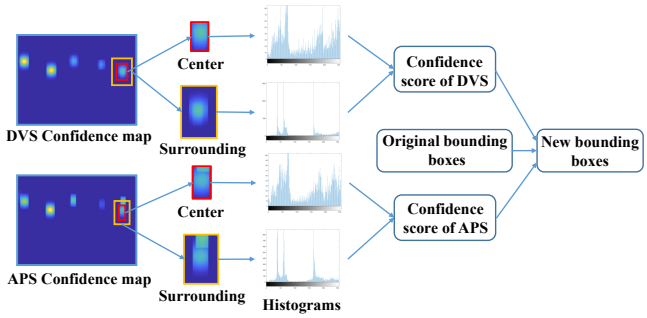


Fig. 3. Framework of fusion process

fusion process. On the left side, confidence maps of two channels generated by YOLO were visualized as heatmaps, where yellow boxes represented large value and blue boxes indicated small value.

The fusion goal is to infer the more accurate information about object from multiple confidence maps. Mathematically, to find out the information variable x (location, size) of an object from the input image I can be formed as $P(x|I)$. Once the identical object appears in different confidence maps, Bayesian inference can be applied to obtain

$$P(x|I) = \int P(x|C)P(C|I)dL \approx \sum_{i=1}^{N} w_i P(x|C_i) \quad (1)$$

where N denotes the number of confidence maps, C_i is the i-th confidence map and $w_i = P(C_i|I)$ represents the confidence score of the i-th confidence map. w_i indicates the weight of each individual confidence map and the sum of w_i over the $i = 1, \cdots, N$. Hence, $P(x|I)$ is approximated by the weighted sum of the confidence maps.

In this case, the value of N is 2, which means two channels of DAVIS. $P(x|C_i)$ represents the location and the size (x, y, w, h) of one bounding box in the image from APS channel or DVS channel. Since $P(x|C_i)$ is available from the output of CNN, the fusion task is converted to derive the optimal confidence score of each bounding box in each channel.

E. Approximation of the confidence scores

However, it is infeasible to obtain $w_{i,opt}$ directly owing to the unknown true probability $P_t(x|I)$ in practice. In order to approximate the value of confidence score, an approach based on the difference between center area and surrounding area of the object defined on confidence maps was proposed. Fig. 3 showed the whole framework of this fusion approach. Both center area and surrounding area were selected in confidence maps from DVS channel and APS channel. The center area, which represented object pixels, was determined by the bounding box generated by CNN. A bigger box representing background pixels was created as surrounding area box at the same center position of the center area box. Fig. 4 showed the relationship between center area box (red box) and surrounding area box (yellow box). Obviously, the area ratio between center area box and surrounding area box was $(1 + \alpha)^2$, where α was the padding ratio of width and height. Through the comparison of characteristics of the

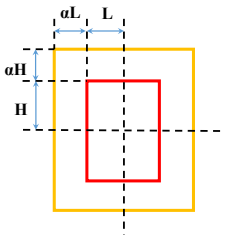


Fig. 4. Relationship between center area and surrounding area

center area and the surrounding area, the difference between object pixels and background pixels could be approximated.

In this work, histograms were adopted to represent characteristics of pixels. The gray-scale histogram of the center area was denoted as $H_c(b)$. Similarly, the gray-scale histogram of the surrounding area was $H_c(b)$. b was the index of the bins of histogram. In order to realize the comparison, both histograms are normalized as normalized center histogram $\{p(b)\}$ and normalized surrounding histogram $\{q(b)\}$. The log likelihood of the normalized histogram can be defined as

$$L(b) = \log \frac{\max\{p(b), \delta\}}{\max\{q(b), \delta\}} \tag{2}$$

where δ is a small value to ensure that log likelihood $L(b)$ makes sense even when $p(b)$ or $q(b)$ is equal to zero.

In order to quantify the distinction of the object pixels and surrounding pixels, the variance of log likelihood $L(b)$ with respect to a normalized histogram $h(b)$ can be defined as

$$\mathrm{var}(L; h) = E\left[L^2(b)\right] - \left(E\left[L(b)\right]\right)^2 \tag{3}$$

Therefore, the ${M_{ii}}^1$ can be approximated by the variance of log likelihood function with respect to the normalized center histogram and normalized surrounding center histogram as follows:

$${M_{ii}}^1 \approx \frac{\mathrm{var}\left(L; (p+q)/2\right)}{\mathrm{var}(L; p) + \mathrm{var}(L; q)} \tag{4}$$

In (4), $\mathrm{var}(L; p)$and $\mathrm{var}(L; q)$ are the center probability variance and surrounding probability variance of the confidence map, respectively. $\mathrm{var}(L; (p+q)/2)$ represents the mixed probability variance of the confidence map. The lower the center probability variance and the surrounding probability variance are, the higher the mixed probability variance is. This results in larger ${M_{ii}}^1$. On the contrary, high center probability and surrounding probability lead to small ${M_{ii}}^1$. Low probability variance indicates that the characteristics of pixels are similar and high probability indicates that the characteristics of pixels are various. Large ${M_{ii}}^1$ means that the current bounding box can distinguish the object and the surrounding area well and it should have higher confidence score in the fusion process.

Substituting ${M_{ii}}^1$ back into $W_{opt} = \frac{{M}^{-1}\bar{\mathbf{1}}}{\bar{\mathbf{1}}^T {M}^{-T}\bar{\mathbf{1}}}$, the i-th element of optimal confidence score is

$$w_{i,opt} \approx \frac{\mathrm{VR}(L_i; p, q)}{\sum_{j=1}^{N} \mathrm{VR}(L_j; p, q)} \tag{5}$$

where

$$\mathrm{VR}(L_i : p, q) = \frac{\mathrm{var}(L_i; (p+q)/2)}{\mathrm{var}(L_i; p) + \mathrm{var}(L_i; q)} \tag{6}$$

Here, $w_{i,opt}$ is directly proportion to between-class probability variance and inversely proportion to within-class probability variance. Hence, with optimal confidence scores of both APS frames and DVS frames, new bounding boxes can be created.

IV. EXPERIMENTS

The experiments of pedestrian detection were conducted on a custom dataset which was collected by a DAVIS240 mounted on a pole nearby traffic signal on TUM campus. The dataset was of length 14.4 seconds with 34.3 million events. For the sake of training and testing, the APS frames and DVS frames were both annotated manually to record the position and the size of bounding boxes as well as its label. Totally, 1313 APS frames, 2630 DVS frames reconstructed by 20 ms interval and 5260 DVS frames reconstructed by 10 ms interval were labeled.

A. Detection by CNN

TABLE I
THE FRAME NUMBER OF EACH DATASET USED IN EXPERIMENTS

Dataset	Train set	Valid set	Test set
APS frames	463	438	440
10ms DVS frames	1776	1748	1738
20ms DVS frames	888	874	869
Mixed APS and 10ms DVS frames	2239	2186	2178
Mixed APS and 20ms DVS frames	1351	1312	1309

In order to be concise, an unified naming method was adopted to shorten the length of testing case names. For example, YOLOv3 driven by APS frames was abbreviated to YOLOv3_APS, YOLO_tiny driven by mixed APS and 10ms DVS frames was abbreviated to YOLO_tiny_M10ms, YOLOv3 which was trained by mixed APS and 10ms DVS frames but tested by APS frames was abbreviated to YOLOv3_M10ms_APS.

Both YOLOv3 and YOLO-tiny were trained with five different datasets separately as shown in Table I. The mixed APS and DVS frames means that frames from two channels are regarded as one set for training a identical network, but the number of frame pairs is the same as the number of DVS frames when fusing both channels. The pedestrian detecting results which were separately obtained by YOLOv3 as well as YOLO-tiny in a road crossing scene are presented in Fig. 5. For the mixed APS and DVS frames, APS frames and DVS frames are tested separately on the same network. In addition, the threshold was set to 0.2, which maximized the F1-score in our case, to display images with bounding boxes whose confidence score was above 0.2. As shown in Fig. 5, The results from YOLOv3 were superior to those from YOLO-tiny on the same dataset. On the other hand, the detection process on 20ms DVS frames performed better than that on 10ms DVS frames using the same network model. Comparing the APS channel with DVS channel, it indicated that the former detected more pedestrians than the latter. Moreover, for YOLO-tiny architecture, the better performance were achieved by using the mixed datasets than using single APS channel or DVS channel.

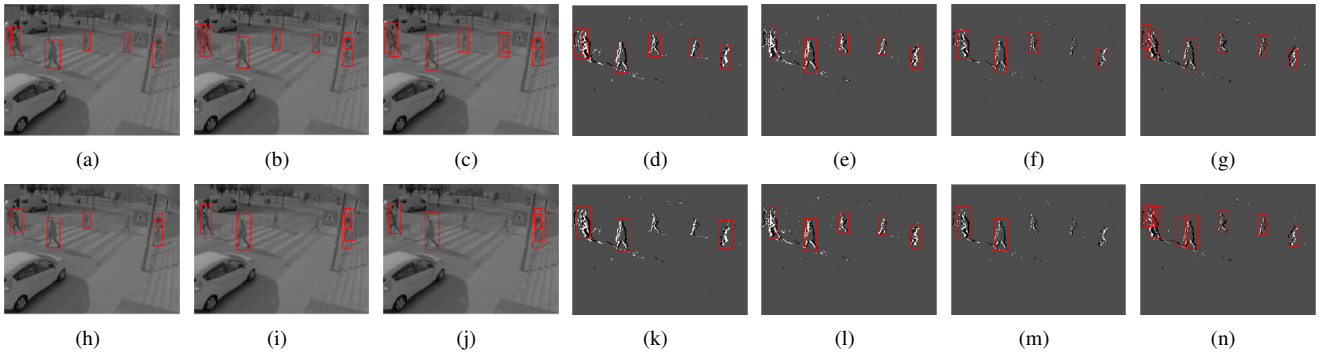
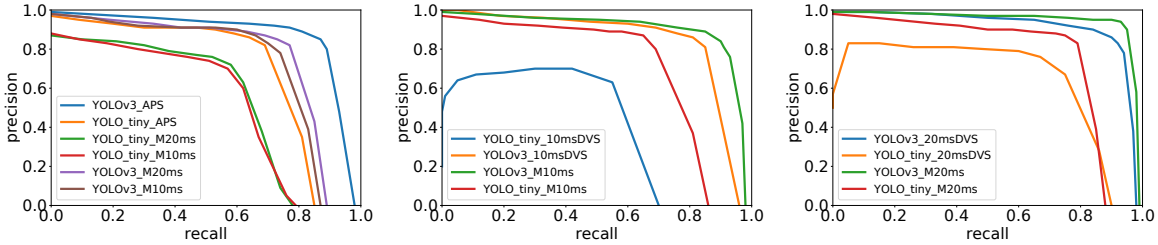


Fig. 5. Predicted results. The outputs from YOLOv3 are shown in the upper row while outputs from YOLO-tiny are shown in the lower row. Meanwhile, the results of the same dataset are shown in the same column. Specifically, (a) YOLOv3 (b) YOLOv3_M20ms_APS (c) YOLOv3_M10ms_APS (d) YOLOv3_20msDVS (e) YOLOv3_M20ms_DVS (f) YOLOv3_10msDVS (g) YOLOv3_M10ms_DVS (h) YOLO_tiny (i) YOLO_tiny_M20ms_APS (j) YOLO_tiny_M10ms_APS (k) YOLO_tiny_20msDVS (l) YOLO_tiny_M20ms_DVS (m) YOLO_tiny_10msDVS (n) YOLO_tiny_M10ms_DVS



(a) APS frames (b) 10msDVS frames (c) 20msDVS frames

Fig. 6. Precision-Recall curves

Meanwhile, the Precision-Recall (P-R) curves of above tests were shown in Fig. 6. In order to reflect the overall performance, the Average Precision (AP) of each model, instead of F1-score, was evaluated on APS and DVS frames, as shown in Table II and Table III, respectively. The AP is approximately equal to the area under the P-R curve.

TABLE II
APS OF APS FRAMES EVALUATED BY DIFFERENT MODELS

	AP (%)
YOLOv3_APS	84.97
YOLO_tiny_APS	70.30
YOLOv3_M20ms_APS	76.79
YOLO_tiny_M20ms_APS	54.04
YOLOv3_M10ms_APS	73.38
YOLO_tiny_M10ms_APS	53.00

TABLE III
APS OF DVS FRAMES EVALUATED BY DIFFERENT MODELS

	AP (%)
YOLOv3_20msDVS	87.30
YOLO_tiny_20msDVS	64.32
YOLOv3_M20ms_DVS	93.08
YOLO_tiny_M20ms_DVS	77.28
YOLOv3_10msDVS	83.30
YOLO_tiny_10msDVS	47.80
YOLOv3_M10ms_DVS	86.68
YOLO_tiny_M10ms_DVS	70.33

In terms of APS frames, YOLOv3 performed better than YOLO-tiny. For YOLOv3 driven only by APS frames, the AP reached 84.97%. In addition, the models which were driven by mixed APS and DVS frames were inferior to that driven only by APS frames. One possible reason is that the number of APS frames used for training was less than DVS frames. For instance, the number of APS frames was half of 20ms DVS frames in mixed APS and 20ms DVS frames dataset. Due to the unbalancedness in mixed datasets, the model tended to learn features from DVS frames with lower APs than APS features.

Regarding on 20ms DVS frames and 10ms DVS frames, the performance of YOLOv3 was better than YOLO-tiny due to the deeper network. The APs from the 20ms DVS frames were higher than that from the 10ms DVS frames. Since the shapes of pedestrians in the 20ms DVS frames are more intact than in the 10ms DVS frames, it is easier for CNN to recognize pedestrians in 20ms DVS frames. As predicted, the networks driven by the mixed APS and DVS frames performed better than that driven by only DVS frames, which indicated that APS frames can be regarded as the supplements of DVS frames.

B. Fusion based on confidence maps

Before fusing, the time synchronization problem of two channels needed to deal with. The frame rate of the APS channel was approximately 25Hz (the interval is 40ms) while the DVS frames were reconstructed in 10 ms and 20 ms interval, respectively. Hence, in the duration of each APS frame, two DVS frames in 20 ms interval or four DVS frames in 10 ms interval were generated. The matched timeline was plotted in Fig. 7. For the pair of APS and 20ms DVS frames, each APS confidence map matched two 20ms DVS confidence maps, including one occurring at the same time and another occurring 20 ms later. For the pair of APS and 10ms DVS frames, each APS confidence map matched four 10ms DVS confidence maps which were generated 10 ms before, at the same time, 10 ms later and 20 ms later,

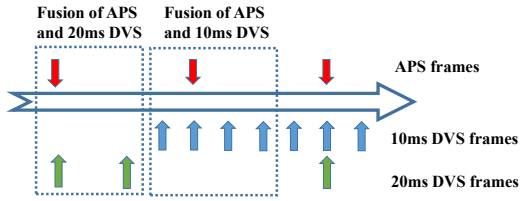


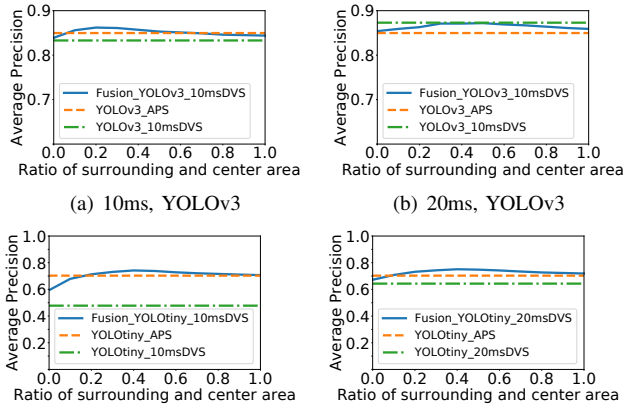
Fig. 7. Timeline of generated APS and DVS frames

respectively. Hence, the maximum time difference between a DVS confidence map and the corresponding APS confidence map was 20 ms. Considering the normal walking speed of an adult is about $1.4m/s$, the maximum position error of the pedestrians from two channels was 0.028 m.

After detection process by CNNs, confidence maps from both APS channel and DVS channel were obtained. To improve detection performance, the fusion method was applied to combine confidence maps from both channels. In this work, eight different fusion pairs were tested, each of them consisted of an APS confidence map and a DVS confidence map. For the non-mixed datasets, APS and DVS confidence map derived from two separated networks which were trained by APS frames and DVS frames, respectively. Whereas, for the mixed datasets, the APS and the DVS confidence map were two subsets of those deriving from an identical network.

Furthermore, the fusion process was performed under different values of $\alpha \in (0, 1)$ to find out the best size ratio of the surrounding box and the center box. As shown in Fig. 8 and Fig. 9, the blue line represented the AP evaluated on DVS frames after fusion. The red and green dotted lines respectively represent the constant AP value on APS frames as well as DVS frames detected by the non-mixed model and the mixed model. By changing the value of α, the maximum AP value occurred between 0.3 and 0.5. Hence, the value of α was determined as 0.4.

Average precisions of different models by using fusion approach were shown in Table IV, where the APS channel was regarded as the conventional frame-based camera.



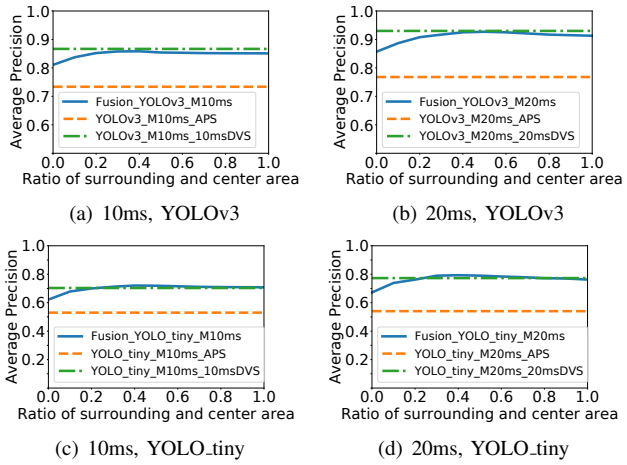
(a) 10ms, YOLOv3 (b) 20ms, YOLOv3

(c) 10ms, YOLO_tiny (d) 20ms, YOLO_tiny

Fig. 8. The relationship between APs and ratio of surrounding area and center area during fusion process in non-mixed models

C. Discussion

As shown in Table IV, the APs of the YOLOv3 were around 12-23% higher than that of the YOLO_tiny for all cases. Pedestrian detection on DVS channel reached



(a) 10ms, YOLOv3 (b) 20ms, YOLOv3

(c) 10ms, YOLO_tiny (d) 20ms, YOLO_tiny

Fig. 9. The relationship between APs and ratio of surrounding area and center area during fusion process in mixed models

TABLE IV
APS IN APS CHANNEL, DVS CHANNEL AND FUSION PROCESS

	APS (%)	DVS (%)	Fusion (%)
10ms, non-mixed, YOLOv3	84.97	83.30	**86.20**
20ms, non-mixed, YOLOv3	84.97	**87.30**	87.20
10ms, non-mixed, YOLO_tiny	70.30	47.80	**74.20**
20ms, non-mixed, YOLO_tiny	70.30	64.32	**75.10**
10ms, mixed, YOLOv3	73.38	**86.68**	85.83
20ms, mixed, YOLOv3	76.79	**93.08**	92.70
10ms, mixed, YOLO_tiny	53.00	70.33	**72.06**
20ms, mixed, YOLO_tiny	54.04	77.28	**79.31**

equivalent performance to the APS channel via YOLOv3, but worse via YOLO_tiny. By contrast, our method fusing both channels significantly improved the AP of YOLO_tiny. Moreover, the AP achieved 72-93% via our method, on average, 3% exceeding to the APS channel on non-mixed datasets and 18% exceeding to APS channel on the mixed datasets. Theoretically, our method could achieve 2-4 times of the frame-rate, that is 50-100 fps, than the APS channel. In experiments, by using the mixed YOLO_tiny architecture, the average frame-rate on APS channel was 141.51 fps and on DVS channel was 145.12 fps, which were higher than inherent frame-rate of APS channel and DVS channel, but it actually achieved about 57 fps by fusing both channels.

V. CONCLUSIONS

Pedestrian detection is a very important and knotty problem in the field of Intelligent Transportation System. Current approaches are usually based on the frame-based camera. Aiming to improve the performance of pedestrian detection at speed and accuracy, a DAVIS was utilized in a scenario of traffic monitoring and a confidence map fusion method was used on both APS and DVS channels. After tested on a custom dataset collected in a cross, our method ultimately achieved 2.28 times higher frame-rate and 3-18% higher average precision than only using the APS camera.

ACKNOWLEDGMENT

This work is supported in part by the scholarship from China Scholarship Council (CSC) under the Grant No. 201606270201 and the funding from National Natural Science Foundation of China (NSFC) under the Grant No. 61872393.

REFERENCES

[1] X. Wei, S. L. Phung, and A. Bouzerdoum, "Pedestrian sensing using time-of-flight range camera," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*. IEEE, 2011, pp. 43–48.

[2] Y.-M. Chan, L.-C. Fu, P.-Y. Hsiao, and M.-F. Lo, "Pedestrian detection using histograms of oriented gradients of granule feature." in *Intelligent Vehicles Symposium*, 2013, pp. 1410–1415.

[3] I. Jegham and A. B. Khalifa, "Pedestrian detection in poor weather conditions using moving camera," in *Computer Systems and Applications (AICCSA), 2017 IEEE/ACS 14th International Conference on*. IEEE, 2017, pp. 358–362.

[4] G. Hinz, G. Chen, M. Aafaque, F. Röhrbein, J. Conradt, Z. Bing, Z. Qu, W. Stechele, and A. Knoll, "Online multi-object tracking-by-clustering for intelligent transportation system with neuromorphic vision sensor," in *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*. Springer, 2017, pp. 142–154.

[5] A. Brunetti, D. Buongiorno, G. F. Trotta, and V. Bevilacqua, "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey," *Neurocomputing*, vol. 300, pp. 17–33, 2018.

[6] S. Zhang, R. Benenson, B. Schiele *et al.*, "Filtered channel features for pedestrian detection." in *CVPR*, vol. 1, no. 2, 2015, p. 4.

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[8] G. Monteiro, P. Peixoto, and U. Nunes, "Vision-based pedestrian detection using haar-like features," *Robotica*, vol. 24, pp. 46–50, 2006.

[9] S. Zhang, C. Bauckhage, and A. B. Cremers, "Informed haar-like features improve pedestrian detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 947–954.

[10] Y. Mu, S. Yan, Y. Liu, T. Huang, and B. Zhou, "Discriminative local binary patterns for human detection in personal album," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[11] C.-H. Zheng, W.-J. Pei, Q. Yan, and Y.-W. Chong, "Pedestrian detection based on gradient and texture feature integration," *Neurocomputing*, vol. 228, pp. 71–78, 2017.

[12] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," 2009.

[13] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[14] S. Paisitkriangkrai, C. Shen, and A. Van Den Hengel, "Strengthening the effectiveness of pedestrian detection with spatially pooled features," in *European Conference on Computer Vision*. Springer, 2014, pp. 546–561.

[15] M. Szarvas, A. Yoshizawa, M. Yamamoto, and J. Ogata, "Pedestrian detection with convolutional neural networks," in *Intelligent vehicles symposium, 2005. Proceedings. IEEE*. IEEE, 2005, pp. 224–229.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

[18] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[21] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint*, 2017.

[22] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "How far are we from solving pedestrian detection?" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1259–1267.

[23] O. Bichler, D. Querlioz, S. J. Thorpe, J.-P. Bourgoin, and C. Gamrat, "Extraction of temporally correlated features from dynamic vision sensors with spike-timing-dependent plasticity," *Neural Networks*, vol. 32, pp. 339–348, 2012.

[24] A. I. Maqueda, A. Loquercio, G. Gallego, N. Garcıa, and D. Scaramuzza, "Event-based vision meets deep learning on steering prediction for self-driving cars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5419–5427.

[25] N. F. Chen, "Pseudo-labels for supervised learning on dynamic vision sensor data, applied to object detection under ego-motion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 644–653.

[26] H. Liu, D. P. Moeys, G. Das, D. Neil, S.-C. Liu, and T. Delbrück, "Combined frame-and event-based detection and tracking," in *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 2511–2514.

[27] D. P. Moeys, F. Corradi, C. Li, S. A. Bamford, L. Longinotti, F. F. Voigt, S. Berry, G. Taverni, F. Helmchen, and T. Delbruck, "A sensitive dynamic and active pixel vision sensor for color or neural imaging applications," *IEEE transactions on biomedical circuits and systems*, vol. 12, no. 1, pp. 123–136, 2018.

[28] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 db 15μ s latency asynchronous temporal contrast vision sensor," *IEEE journal of solid-state circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[29] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.