


Received July 17, 2019, accepted July 28, 2019, date of publication August 19, 2019, date of current version September 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2936126

# Joint Object Detection and Depth Estimation in Multiplexed Image

CHANGXIN ZHOU<sup>1</sup>, YAZHOU LIU<sup>1</sup> , QUANSEN SUN<sup>1</sup>,  
AND PONGSAK LASANG<sup>2</sup> , (Member, IEEE)

<sup>1</sup>School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

<sup>2</sup>Core Technology Group, Panasonic R&D Center Singapore, Singapore 469332

Corresponding author: Yazhou Liu (yazhouliu@njust.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61672286.

**ABSTRACT** This paper presents an object detection method that can simultaneously estimate the positions and depth of the objects from multiplexed images. Multiplexed image is produced by a new type of imaging device that collects the light from different fields of view using a single image sensor, which is originally designed for stereo, 3D reconstruction and broad view generation using computational imaging. Intuitively, multiplexed image is a blended result of the images of multiple views and both of the appearance and disparities of objects are encoded in a single image implicitly, which provides the possibility for reliable object detection and depth/disparity estimation. Motivated by the recent success of CNN based detector, a multi-anchor detector method is proposed, which detects all the views of the same object as a clique and uses the disparity of different views to estimate the depth of the object. The proposed method is interesting in the following aspects: firstly, both locations and depth of the objects can be simultaneously estimated from a single multiplexed image; secondly, there is almost no computation load increase comparing with the popular object detectors; thirdly, even in the blended multiplexed images, the detection and depth estimation results are very competitive. There is no public multiplexed image dataset yet, therefore the evaluation is based on the simulated multiplexed image using the stereo images from KITTI, and very encouraging results have been obtained.

**INDEX TERMS** Object detection, depth estimation, multiplexed image.

## I. INTRODUCTION

Benefiting from the development of deep convolution neural networks (CNN) [2]–[4], object detection [5]–[7] has made great progress in recent years. Given an image, the purpose of object detection is to obtain the location and category information of each object instance in it. As an important part of computer vision, object detection has a broad range of applications in many areas such as autonomous driving [8], [9], robot vision [10], [11], and surveillance system [12]. However, some applications (e.g. autonomous driving) not only need the positions of objects in the image but also require these detected objects' actual depth.

The task can be completed by 3D object detection that predicts the 3D location, dimensions (height, width, and length) and orientation of objects. Benefiting from the accurate depth measurement, methods [13]–[15] based on LiDAR (Light

Detection And Ranging) data achieve state-of-the-art performance in 3D object detection. But, LiDAR has the disadvantage of high cost, relatively short perception range and sparse information. On the other hand, methods [16]–[18] whose input is a monocular image cannot predict the accurate depth of objects, especially for unseen scenes. Stereo R-CNN [19] is a 3D object detection method that utilizes the sparse and dense, semantic and geometry information in the stereo images. But, its inference speed (0.28s per image) is far from the real-time demanded for autonomous driving because of extracting features from two images (a stereo image) and post-processing.

To this end, we propose a simple and fast detector capable of depth estimation based on the multiplexed image. Optically multiplexed imaging is a developing field in the area of computational imaging. Shepard and Rachlin [1] proposed a new imaging device that collects multiple channels of light simultaneously by a single sensor, as illustrated in Fig. 1. They also proposed methods to disambiguate a captured

The associate editor coordinating the review of this article and approving it for publication was Sudipta Roy.

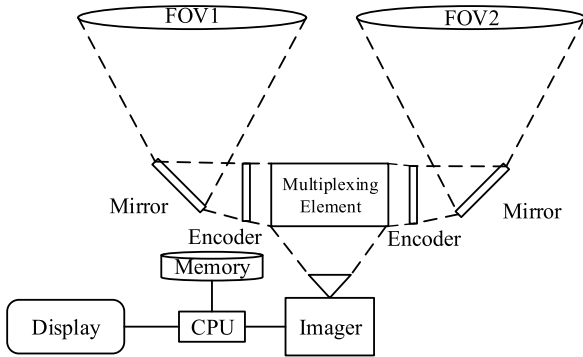


FIGURE 1. The architecture of the device for multiplexed imaging [1].

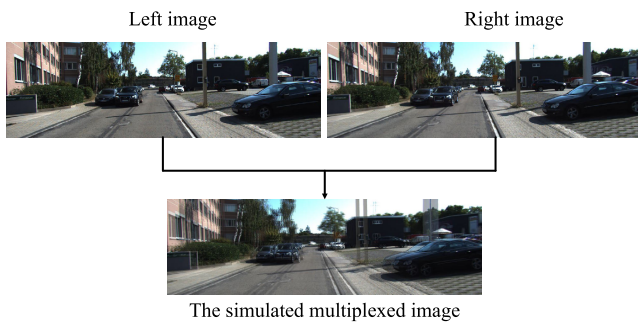


FIGURE 2. A stereo image (the image  $I_l$  from the left camera and the image  $I_r$  from the right camera) and the multiplexed image. In this paper, we use overlapped image  $I = (I_l + I_r)/2$  to simulate the multiplexed image.

multiplexed image to create images for each of the plurality of image channels that can produce stereo images. Comparing with multiplexed imaging, the method that uses two or more cameras to create a stereo image suffers from the added cost, power, volume and complexity of using multiple cameras. What’s more, our method directly uses multiplexed images as input instead of stereo images recovered from them. This makes our method as fast as a common detector.

Notice that the purpose of our work is to estimate depth information of detected objects. The multiplexed imaging in this paper has two horizontal camera lens like stereo imaging but only uses a single imaging sensor. This makes the multiplexed image equivalent to the overlapping of a stereo image pair, as shown in Fig. 2.

Our work is based on the observation that both the appearance and the disparity of every object are encoded implicitly in the multiplexed image. Our method, named Disparity Detector, firstly detects all the views of the same object as a clique by the strategy we proposed, then uses the disparity of different views to estimate the depth of each object. There is no public multiplexed image dataset yet; therefore, the experiments are conducted on the simulated multiplexed image using the stereo images from KITTI [20]. The proposed method is developed based on the VGG16 [2] backbone and SSD detector framework [5], but it can be easily incorporated with other anchor-based CNN detectors (e.g. DSSD [21]) and backbones (e.g. ResNet [3]) for better performance. The whole pipeline of our work is shown in Fig. 3.

The works in this paper are interesting in the following aspects: 1) The proposed method can simultaneously estimate the 2D positions and the actual depth of objects in the multiplexed image; 2) Comparing with popular object detectors, the proposed method has almost no extra computation load or latency time; 3) Even in blended multiplexed images, the proposed method achieves competitive detection and depth estimation results.

II. RELATED WORK

In this section, we are going to briefly review the advances of the related works from three aspects: multiplexed imaging, object detection, and disparity estimation framework.

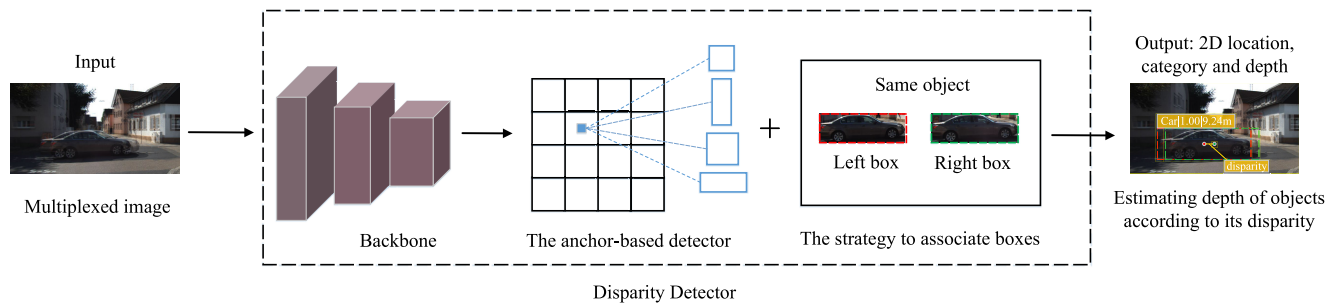
A. MULTIPLEXED IMAGING

Optically multiplexed image acquisition techniques [1], [22] have become increasingly popular for encoding different exposures, color channels, light fields, and other properties of light onto image sensors. Wetzstein et al. [23] presented a joint optical light modulation and computational reconstruction approach to boost the dynamic range of multiplexed photographs. Shepard and Rachlin [1] proposed new multiplexed imaging devices and methods that disambiguate a captured multiplexed image to create image channels. Uttam et al. [24] proposed a class of task-specific multiplexed imagers to collect encoded data in a lower-dimensional measurement space named superposition space and developed a decoding algorithm that tracks targets directly in this superposition space.

B. OBJECT DETECTION

The goal of object detection is to obtain the location and category information of each object instance in a given image. Classic detectors extract features of each sliding window by hand-engineered descriptors (e.g. HOG [25], SIFT [26] and Edge Box [27]) and then apply classifiers to find objects. In the recent years, deep convolution neural networks (CNNs) are widely used for vision tasks. Different from classic detectors, CNN-based object detectors use image features extracted by a base network (e.g. VGG16 [2]) to find objects. Due to the outstanding performance, CNN-based object detectors become the main force in the detection field. Usually, CNN-based detectors can be roughly divided into two categories, i.e., the two-stage approach and the one-stage approach. The two-stage approach (e.g. R-CNN [28], Fast R-CNN [29], and Faster R-CNN [30]) has two steps, where the first one produces a fixed number of potential object proposals, and the second one predicts the offsets of the spatial location and category labels. The two-stage methods have been achieving top results on several benchmarks, including PASCAL VOC [31] and MS COCO [32]. Recently plenty of novel techniques are used for better performance, such as iterative bounding box [7] regression, training strategy [33] and new loss [34] for bounding box regression.

The two-stage approach could be computationally expensive for real applications, which have limited storage and



**FIGURE 3.** The whole pipeline of our Disparity Detector framework. Disparity Detector takes the multiplexed image as input and consists of three flexible modules (the backbone, the anchor-based detector, and the strategy to associate boxes of the same object). Disparity Detector estimates the depth of the detected objects by their disparities of left and right views.

computational capability. The one-stage approach directly predicts class probabilities and bounding box offsets with a single forward convolutional neural network. Therefore, the one-stage approach has a better trade-off between speed and accuracy. SSD [5] and YOLO [35]–[37] are the representative object detectors of the one-stage approach. YOLO [35] directly predicted the object category and the offsets of spatial location with a single convolution network with fast inference speed. Based on YOLO, YOLO9000 [36] used batch normalization after each convolution layers for better results and used convolution layers in place of fully connected layers for classification and regression of location offsets. Liu *et al.* [5] proposed a single shot object detector, named SSD, which predicts objects using feature maps with different receptive fields. DSSD [21] applied deconvolution operation to SSD for additional context and used a more complex prediction module for better accuracy. RetinaNet [38] investigated the extreme class imbalance problem in the current one-stage approach and solved it by re-designing the loss function. Although the one-stage methods achieved faster speed than that of the two-stage, their performance is still inferior to the two-stage approach.

### C. DISPARITY ESTIMATION

The goal of the depth estimation task is to predict the disparity of every pixel in the input image. Several depth estimation methods have made great progress benefiting from the rapid development of neural networks. Zbontar and LeCun [39] calculated patch similarities of a stereo image pair with a Siamese convolutional network. Their method inspired several studies on depth estimation using convolution networks. DispNet [40] formulated the depth estimation as a supervised learning problem and predicted disparities directly with a convolutional network. PSMNet [41] used spatial pyramid pooling to take advantage of the capacity of global context information and achieved state-of-the-art performance. Methods mentioned above can generate an accurate disparity map but they are slow and require extensive computation. To achieve a better trade-off between accuracy and speed, AnyNet [42] estimated the depth in several stages, during which the model can be queried at any time to output its current best estimation. Monodepth [43] proposed the

unsupervised method that attempted to generate a dense disparity map by training the network with an image reconstruction loss. It only required the stereo image pair for training and enabled the network to learn to perform single image depth estimation at a faster speed.

### III. DISPARITY DETECTOR FRAMEWORK

In this section, we propose a strategy that can be cooperated with any anchor-based object detector to form our Disparity Detector. Disparity Detector can simultaneously detect objects and estimate the depth of the detected objects in the multiplexed image. Firstly, we analyze the characteristic of the multiplexed image and explain the reason why current detectors are not suitable for the multiplexed images. Then, we introduce the Disparity Detector which composes of a backbone network, an anchor-based object detector, and our proposed strategy.

#### A. THE CHARACTERISTIC OF MULTIPLEXED IMAGE

Since the multiplexed image used in this paper is a mixture of images from two horizontal views, each object in it has two parts that can be located with a pair of horizontal bounding boxes (dashed boxes in Fig. 5). The disparity of an object can be estimated by the horizontal pixel distance between the centers of its two boxes. In stereo vision, the formula of disparity and depth is:

$$depth = \frac{b \times f}{disparity} \quad (1)$$

where  $b$  is the stereo baseline distance and  $f$  is the focal length of the camera. Benefiting from this characteristic, the multiplexed image provides the possibility of joint object detection and depth estimation.

However, the current CNN-based object detectors have limitations when directly used to detect objects in the multiplexed images: The pair information of each object's two boxes can't be fed into the network for training; Left or right box of the same object may be filtered during non-maximum suppression (NMS) due to their large overlapping area; From the output of a detector, a set of predicted bounding boxes, the disparity of each object is unavailable because the detector cannot associate left and right boxes of the same object.

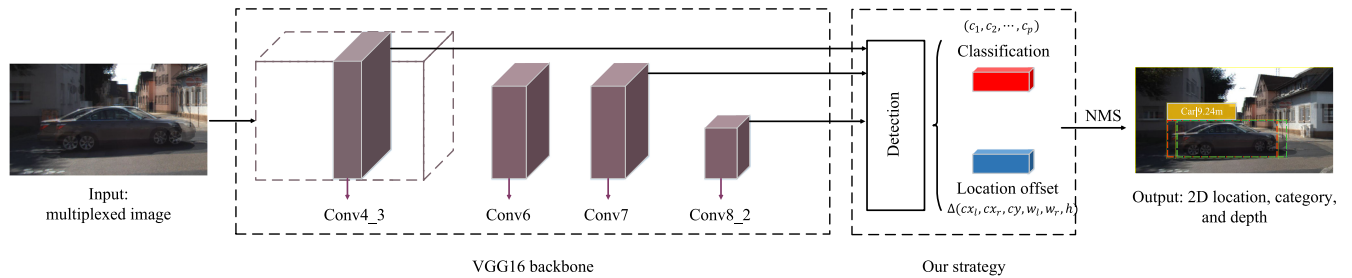


FIGURE 4. The detailed network architecture of the proposed Disparity Detector.

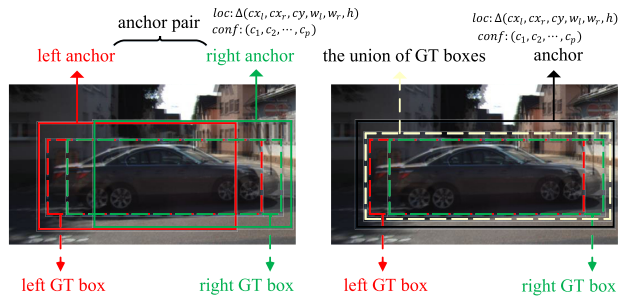


FIGURE 5. The strategy to associate boxes of the same object. Left: Our proposed strategy *anchor pair*; Right: The strategy proposed by [19].

**B. DISPARITY DETECTOR**

Considering the speed-vs-accuracy trade-off, our Disparity Detector is based on SSD [5], which is a representative one-stage detector. SSD is built on top of a backbone network (VGG16 [2]) that ends (or is truncated to end) with some convolutional layers. To detect objects with multiple sizes, SSD utilizes feature maps with different receptive fields to predict scores and offsets for the predefined anchors. These predictions are performed by  $3 \times 3 \times \#channels$  dimensional filters, one filter for classification score and one for location offsets of the anchors. Finally, non-maximum suppression (NMS) is used to reduce redundancy and obtain the detection results. More details can be found in [5].

To enable the detector to detect and associate the left and right boxes from the same object in the multiplexed image, we propose a strategy, named *anchor pair*, and cooperate it with VGG16 [2] backbone and SSD [5] detector to form our Disparity Detector. The detailed network architecture is shown in Fig. 4.

**1) ANCHOR PAIR**

Inspired by each object having a pair of horizontal left and right GT boxes, we propose *anchor pair* which is an extension of the *anchor*. Each *anchor pair* consists of a pair of horizontal left and right anchors, as shown in Fig. 5. For each anchor pair, we calculate its left anchor’s IoU ( $IoU_l$ ) with the left GT box and its right anchor’s IoU ( $IoU_r$ ) with the corresponding right GT box. If its  $IoU_l$  and  $IoU_r$  are both above 0.5, a positive label is assigned to the anchor pair. A negative label is assigned if  $IoU_l$  and  $IoU_r$  are both below 0.5. Each anchor pair predicts a classification score

so that its left and right anchors share the classification score. We let the positive anchor pair predict location offsets  $[\Delta cx_l, \Delta cx_r, \Delta cy, \Delta w_l, \Delta w_r, \Delta h]$  respecting to the left and right GT boxes, where we use  $cx, cy$  to denote the horizontal and vertical coordinates of the box center in image space,  $w, h$  for width and height of the box, and the superscript  $(\cdot)_l, (\cdot)_r$  for corresponding terms in the left and right box. Note that we use the same  $cy, h$  offsets  $\Delta cy, \Delta h$  for the left and right boxes because we use rectified stereo images to simulate multiplexed images. Therefore, we have six offsets for each anchor pair instead of four in the original SSD. Since each predicted object’s left and right boxes are generated by the same anchor pair and shared the classification score, they are associated as a clique naturally. We use NMS on predicted objects’ left and right boxes separately to reduce redundancy and get final detection results. A predicted object will be kept if its left and right box are both kept after NMS.

**2) THE DIFFERENCE WITH THE CURRENT STRATEGY**

Stereo R-CNN [19] proposed a simple but rough strategy (referred to as *strategy stereo*) to associate boxes of the same object. As shown in Fig. 5, *strategy stereo* assigned the union of left and right ground-truth boxes (referred to as union GT box) as the target for object classification. And an anchor is assigned a positive label if its IoU with one of the *union GT boxes* is above a threshold  $T_H$ , or a negative label if the IoU is below  $T_L$ . Each positive anchor predicts offsets respecting to the left and right GT boxes contained in the target union GT box. However, the positive anchor having an IoU above  $T_H$  with the *union box* cannot guarantee that it also has a high IoU for each box inside the union box. In other words, the anchor with the positive label may have an IoU below  $T_H$  (even below  $T_L$ ) with left or right box.

**IV. IMPLEMENTATION DETAILS**

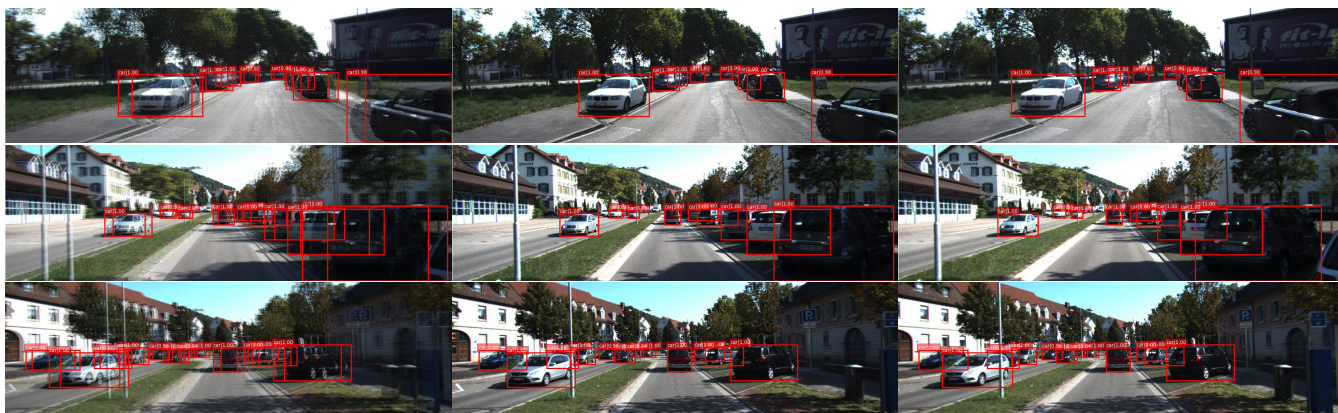
**A. ANCHOR SHAPE**

Different from the implementation in [5], the shape of the anchor is determined by the k-means algorithm proposed by YOLO9000 [36]. We first run Cluster IoU [36] on the training set to automatically choose  $n_1$  ( $n_1 = 6$  in this paper) of different anchor shapes ( $w, h$ ). And these  $n_1$  anchor shapes are used by *strategy stereo*. Cluster IoU is a k-mean algorithm with the distance metric:

$$d(anchor, centroid) = 1 - IoU(anchor, centroid) \quad (2)$$

**TABLE 1.** Average precision (in %) of detection, evaluated on the KITTI evaluation set.

Method	Setting	$AP_{left}$			$AP_{stereo}$			Runtime(s)
		Easy	Mode	Hard	Easy	Mode	Hard	
Faster R-CNN [30]	-	99.23	98.40	90.88	-	-	-	0.082
MFFD [44]		91.16	84.01	72.43	-	-	-	0.005
YOLOv3 [37]		95.96	95.51	88.26	-	-	-	0.031
SSD [5]		98.78	96.06	88.49	-	-	-	0.027
Disparity Detector	<i>strategy stereo</i> [19]	98.37	95.54	85.93	93.17	90.37	81.15	0.027
	<i>anchor pair</i> (ours)	<b>98.55</b>	<b>96.00</b>	<b>88.62</b>	<b>93.69</b>	<b>91.87</b>	<b>83.14</b>	0.027

**FIGURE 6.** Examples of the detection results on KITTI evaluation set using the proposed method. Left: Detection results on the multiplexed images; Middle: Mapping the detection results to left images  $I_l$ ; Right: Mapping the detection results to right images  $I_r$ .

Then, in each of the  $n_1$  cluster, we use the standard k-means with Euclidean distance to get  $n_2$  ( $n_2 = 4$  in this paper) of different distances  $d$ . Therefore, there are  $n_1 \times n_2$  shapes of the anchor pair for the proposed strategy *anchor pair*.

## B. NETWORK

We have made minor changes when re-implementing SSD [5]: (1) The size of network input is  $576 \times 320$ . (2) We remove the layers after Conv\_8 in the original SSD implementation and use three layers feature map (Con4\_3, Conv7, and Conv8\_2) for prediction. Other settings, such as data augmentation and hard example mining, are the same as the original SSD. We train the network using SGD with a weight decay of 0.0001. We train 100K iterations (the batch size is 16) in total on an RTX2080 Ti GPU. The learning rate is initially set to 0.001 and reduced by a factor of 0.1 at the 60K and 80K iterations.

## V. EXPERIMENTS

In this section, we evaluate the proposed Disparity Detector on the KITTI detection dataset [20]. Firstly, we introduce the preparation of the dataset. Then, we compare our proposed strategy with the strategy from Stereo R-CNN [19] on the performance of object detection and depth estimation, respectively.

### A. DATASET PREPARATION

The KITTI detection dataset [20] provides 7481 training stereo image pairs and 3D bounding box label (The 2D box label of the left or right image can be calculated by projecting the 3D box to the corresponding image). We simulate

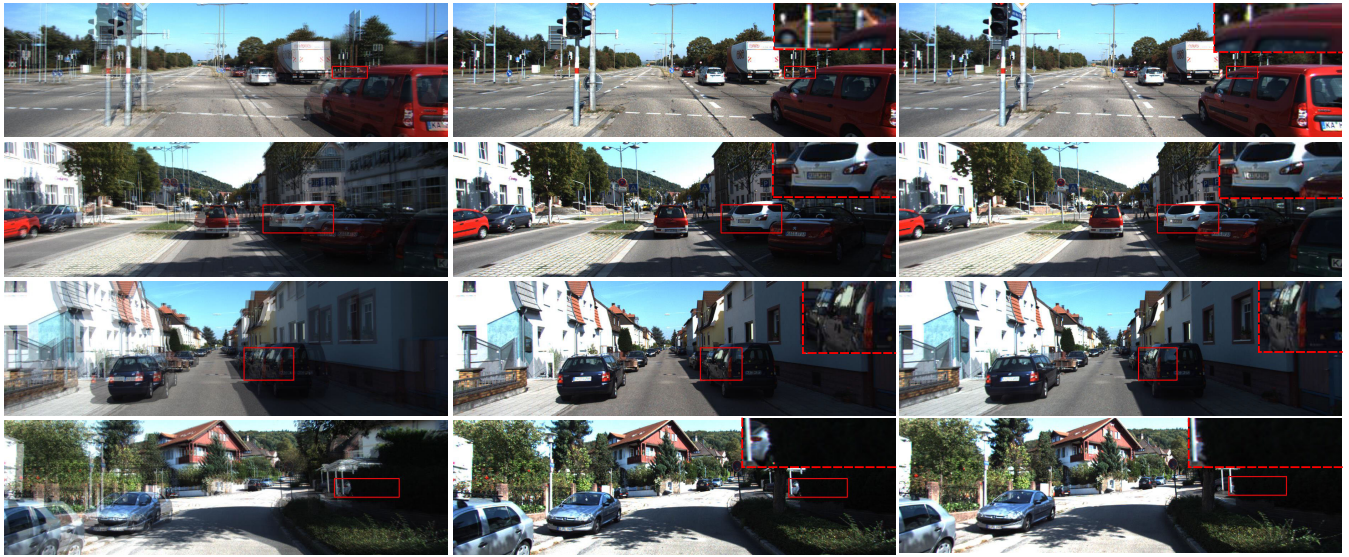
the multiplexed images using the stereo image pairs by  $I = (I_l + I_r)/2$ . Following Stereo R-CNN [19], this paper only uses *car* category labels for training and evaluation, and uses 50% of the images for training (*training set*), the rest images are used for evaluation (*evaluation set*). The evaluation has three difficulty levels: *easy*, *moderate*, and *hard*, which are defined in terms of the occlusion, size and truncation levels of objects. Checking [20] for a detailed definition of the difficulty levels.

We also use KITTI stereo 2015 [45] to train the depth estimation methods [39], [41], [43]. It contains 200 training stereo image pairs with sparse ground-truth disparities obtained using LiDAR.

### B. PERFORMANCE OF OBJECT DETECTION

In this section, we evaluate the proposed Disparity Detector's performance of object detection. We train the Disparity Detector with two strategies on the multiplexed image *training set* and evaluate them on the multiplexed image *evaluation set*. We also train the base detector, SSD and other common detectors (Faster R-CNN [30], MFFD [44], and YOLOv3 [37]) on the left image *training set* and evaluate it on the left image *evaluation set*. For Faster R-CNN, the original image is resized to 600 pixels in the shorter side. For SSD, MFFD, and YOLOv3, the input image is resized to  $576 \times 320$ . All detectors share the same anchor shape that introduced in Section. IV.

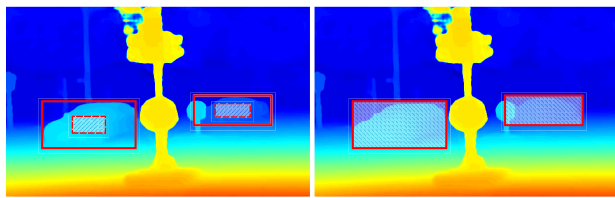
Our Disparity Detector aims to simultaneously detect and associate boxes of the same object in the multiplexed image. Besides evaluating the Average Precision (AP) on the left image (mapping the detection results to the



**FIGURE 7.** Imbalanced visual information of object in different views. Left: The objects in the multiplexed image; Middle: The objects in the left image; Right: The objects in the right image.

**TABLE 2.** The proposed Disparity Detector’s performance of object-level depth estimation.

Method	Setting	Occlusion=0				Occlusion=1				Occlusion=2			
		EPE	>1px	>3px	>5px	EPE	>1px	>3px	>5px	EPE	>1px	>3px	>5px
Disparity Detector	<i>strategy stereo</i> [19]	1.23	41.78	7.46	2.71	1.37	48.80	9.12	2.76	1.46	54.38	10.59	<b>3.14</b>
	<i>anchor pair</i> (ours)	<b>1.11</b>	<b>35.44</b>	<b>6.58</b>	<b>2.16</b>	<b>1.26</b>	<b>42.77</b>	<b>8.00</b>	<b>2.67</b>	<b>1.38</b>	<b>48.15</b>	<b>9.54</b>	3.28



**FIGURE 8.** Combining object detection and depth estimation methods to predict object-level depth. The bounding box (red) of the object is predicted by SSD [5], and the depth estimation methods output the dense disparity map. Left: Using the average disparity of central pixels as the object’s disparity. Right: Using the median disparity of pixels inside the box as the object’s disparity.

left image), we also use the stereo AP metric which defined in Stereo R-CNN [19] to evaluate the association performance. In stereo AP, a left-right box pair is considered as the True Positive (TP) if the following conditions are met:

1. The maximum IoU between the left box and left GT boxes is above the threshold;
2. The maximum IoU between the right box and right GT boxes is above the threshold;
3. The selected left and right GT boxes belong to the same object.

We mark the best method in bold-red. As reported in Table 1, the proposed *anchor pair* outperforms *strategy stereo* [19] by large margins. Specifically, the proposed *anchor pair* outperforms *strategy stereo* over 2.69% and 1.99% for  $AP_{left}$  (Hard level) and  $AP_{stereo}$  (Hard level), respectively. We attribute it to our strategy’s accurate match of

anchor and GT boxes. Some detection examples of Disparity Detector are shown in Fig. 6.

We also observe that our Disparity Detector whose input is the multiplexed image can get comparable performance on the  $AP_{left}$  compared with the common detectors that take the left image as input. Our task (detecting objects from two views) is more challenging and difficult than detecting objects on the left image using the common detector. The visual information of an object in the left and right views is imbalanced. To be specific, some objects that are visible in the left view could be occluded completely (even invisible) in the right view, as shown in Fig. 7. Due to the lack of visual information in the right view, our method cannot detect these objects. This visual information imbalance problem will be the main focus of our future work.

In this section, we evaluate the proposed Disparity Detector’s performance of object-level depth estimation. We report the results of Disparity Detector with two strategies on the *evaluation set*. For evaluation, we use the end-point-error (EPE), which is calculated as the average Euclidean distance between the estimated disparity and the ground-truth. We also use the percentage of disparities with EPE larger than  $t$  pixels ( $>tpx$ ). Here an object is considered as correct if its disparity EPE is less than  $t$  pixels. And the disparity of an object is estimated by the horizontal pixel distance between its centers of the left box and the right boxes.

Table 2 shows the comparison results of objects with different occlusion levels. We mark the best method in bold-red. In the KITTI label, the objects with  $occlusion = 0$  are fully



**FIGURE 9.** Examples of object-level depth estimation results on KITTI *evaluation set* using Disparity Detector (with *anchor pair*). The number in bold-red is the ground-truth depth, and the number in bold-yellow is the predicted depth from our method.

**TABLE 3.** The performance of object-level depth estimation by combining object detection and depth estimation.

Method	Setting	Occlusion=0			Occlusion=1			Occlusion=2			Runtime(s)
		EPE	>3px	>5px	EPE	>3px	>5px	EPE	>3px	>5px	
Disparity Detector	<i>anchor pair</i>	1.11	<b>6.58</b>	<b>2.16</b>	<b>1.26</b>	<b>8.00</b>	<b>2.67</b>	<b>1.38</b>	<b>9.54</b>	<b>3.28</b>	<b>0.027</b>
SSD [5]+MC-CNN [39]	<i>median</i>	2.11	22.11	11.72	3.44	34.83	19.42	7.62	77.54	55.46	0.704
	<i>mean</i>	1.62	14.75	7.91	3.55	33.88	20.67	7.78	80.44	57.08	
SSD [5]+Monodepth [43]	<i>median</i>	1.69	16.44	5.45	2.87	32.71	14.80	6.21	69.46	44.91	0.072
	<i>mean</i>	1.70	17.34	5.61	3.33	36.67	18.55	6.80	74.11	49.94	
SSD [5]+PSMNet [41]	<i>median</i>	0.98	7.53	3.27	1.72	15.12	7.79	5.98	62.74	43.74	0.612
	<i>mean</i>	<b>0.93</b>	7.10	3.39	2.34	23.03	12.61	6.61	73.19	47.07	

visible, *occlusion* = 1 means the objects are partly occluded, and the objects with *occlusion* = 2 are largely occluded. Our *anchor pair* outperforms *strategy stereo* in all object occlusion levels except >5px when *occlusion* = 2. This demonstrates that the detector with our proposed strategy has a better ability to locate objects. We show some depth estimation results of Disparity Detector with *anchor pair* in Fig. 9.

**C. PERFORMANCE OF DEPTH ESTIMATION**

We also conduct an interesting experiment that combines SSD with depth estimation methods [39], [41], [43] to predict object-level depth. These depth estimation methods are trained with the KITTI stereo dataset. We utilize the bounding boxes from SSD to locate objects and get the disparity of each object from the disparity map predicted by depth estimation methods. As shown in Fig. 8, the disparity of an object can be estimated by the average disparity of its central pixels or the median disparity of pixels inside its bounding box:

$$disp_{mean}^{object} = \frac{1}{0.4w \times 0.4h} \times \sum_{cx-0.2w}^{cx+0.2w} \sum_{cy-0.2h}^{cy+0.2h} disp(i, j) \quad (3)$$

and

$$disp_{median}^{object} = median\{disp(i, j) | (i, j) \in BB\} \quad (4)$$

where *w*, *h*, (*cx*, *cy*) are the width, height and center of an object’s bounding box (BB), respectively. *disp*(*i*, *j*) is the disparity map predicted by a depth estimation method such as MC-CNN [39], PSMNet [41] and Monodepth [43].

Table 3 shows the evaluation results on the KITTI detection *evaluation set*. We mark the best method in bold-red. It can be observed that this combination scheme (with the state-of-the-art depth estimation method PSMNet [41]) achieves a comparable performance when the objects are not occluded (*occlusion* = 0). However, their performance drops severely when the object is occluded (*occlusion* = 1 and *occlusion* = 2) because most pixels inside the bounding box do not belong to the object. By contrast, our proposed Disparity Detector performs well and steadily in all occlusion levels. What’s more, our method consumes much less time (3× to 20× faster).

**VI. CONCLUSION AND FUTURE WORK**

In this paper, we have presented a new method for simultaneous object detection and depth estimation from a single

multiplexed image. The multiplexed image can encode the appearance and the disparity of every object by blending multiple views. The object detection task is formulated as a clique detection task that can detect and associate all the views of the same object in the image. Then, the actual position on any single view and the disparity/depth of the object can be recovered. The evaluation results showed that the proposed method can yield very competitive results compared with the state-of-the-art. And we find that the visual information of an object in different views could be imbalanced, this problem will be the main focus of our future work.

## REFERENCES

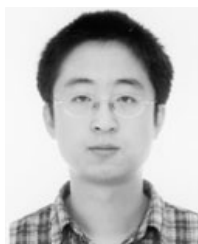
- [1] R. H. Shepard and Y. Rachlin, "Devices and methods for optically multiplexed imaging," U.S. Patent 14 668 214, Mar. 25, 2018.
- [2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
- [4] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4700–4708.
- [5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.
- [6] C. Yao, P. Sun, R. Zhi, and Y. Shen, "Learning coexistence discriminative features for multi-class object detection," *IEEE Access*, vol. 6, pp. 37676–37684, 2018.
- [7] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," 2017, *arXiv:1712.00726*. [Online]. Available: <https://arxiv.org/abs/1712.00726>
- [8] J. Wei, J. He, Y. Zhou, K. Chen, Z. Tang, and Z. Xiong, "Enhanced object detection with deep convolutional neural networks for advanced driving assistance," *IEEE Trans. Intell. Transp. Syst.*, to be published.
- [9] X. Hu, X. Xu, Y. Xiao, H. Chen, S. He, J. Qin, and P.-A. Heng, "SINet: A scale-insensitive convolutional neural network for fast vehicle detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 1010–1019, Mar. 2019.
- [10] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1817–1824.
- [11] D. Lee, G. Kim, D. Kim, H. Myung, and H.-T. Choi, "Vision-based object detection and tracking for autonomous navigation of underwater robots," *Ocean Eng.*, vol. 48, pp. 59–68, Jul. 2012.
- [12] A. Basharat, A. Gritai, and M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
- [13] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1907–1915.
- [14] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3D object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 641–656.
- [15] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 652–660.
- [16] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2147–2156.
- [17] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3D bounding box estimation using deep learning and geometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7074–7082.
- [18] B. Xu and Z. Chen, "Multi-level fusion based 3D object detection from monocular images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2345–2353.
- [19] P. Li, X. Chen, and S. Shen, "Stereo R-CNN based 3D object detection for autonomous driving," in *Proc. CVPR*, 2019, pp. 7644–7652.
- [20] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 3354–3361.
- [21] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," 2017, *arXiv:1701.06659*. [Online]. Available: <https://arxiv.org/abs/1701.06659>
- [22] C. Brown, "Multiplex imaging with multiple-pinhole cameras," *J. Appl. Phys.*, vol. 45, no. 4, pp. 1806–1811, 1974.
- [23] G. Wetzstein, I. Ihrke, and W. Heidrich, "Sensor saturation in Fourier multiplexed imaging," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 545–552.
- [24] S. Uttam, N. A. Goodman, M. A. Neifeld, C. Kim, R. John, J. Kim, and D. Brady, "Optically multiplexed imaging with superposition space tracking," *Opt. Express*, vol. 17, no. 3, pp. 1691–1713, 2009.
- [25] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, vol. 1, no. 1, pp. 886–893.
- [26] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- [27] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 391–405.
- [28] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
- [29] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.
- [30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
- [31] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.
- [32] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.
- [33] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 761–769.
- [34] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," 2019, *arXiv:1902.09630*. [Online]. Available: <https://arxiv.org/abs/1902.09630>
- [35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.
- [36] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7263–7271.
- [37] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: <https://arxiv.org/abs/1804.02767>
- [38] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2980–2988.
- [39] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *J. Mach. Learn. Res.*, vol. 17, nos. 1–32, p. 2, 2016.
- [40] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4040–4048.
- [41] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5410–5418.
- [42] Y. Wang, Z. Lai, G. Huang, B. H. Wang, L. van der Maaten, M. Campbell, and K. Q. Weinberger, "Anytime stereo image depth estimation on mobile devices," 2018, *arXiv:1810.11408*. [Online]. Available: <https://arxiv.org/abs/1810.11408>
- [43] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 270–279.
- [44] S. Cao, Y. Liu, P. Lasang, and S. Shen, "Detecting the objects on the road using modular lightweight network," 2018, *arXiv:1811.06641*. [Online]. Available: <https://arxiv.org/abs/1811.06641>



[45] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3061–3070.



**CHANGXIN ZHOU** received the B.S. degree in electrical engineering from Soochow University, Suzhou, China, in 2017. He is currently pursuing the M.S degree with the Department of Computer Science and Engineering, Nanjing University of Science and Technology (NUST). His research interests mainly include object detection and image denoising.



**YAZHOU LIU** received the B.S. degree in mechanical engineering from Harbin Engineering University, Harbin, China, in 2002, and the M.E. and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, in 2004 and 2009, respectively. From 2007 to 2009, he was an Engineer with Panasonic Research and Development Center Singapore. From 2009 to 2011, he was a Postdoctoral Research Fellow with the Machine Vision Group, Oulu University, Finland. Since 2011, he has been a Faculty Member with the Department of Computer Science and Engineering, Nanjing University of Science and Technology (NUST).



**QUANSEN SUN** received the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology (NUST), in 2006, China. He visited the Department of Computer Science and Engineering, The Chinese University of Hong Kong, from 2004 to 2005. He is currently a Professor with the Department of Computer Science, NUST. His current research interests include pattern recognition, image processing, remote sensing information system, and medicine image analysis.



**PONGSAK LASANG** (M'10) received the B.E. degree (Hons.) in electronics and telecommunication engineering, the M.E. degree in electrical engineering, and the Ph.D. degree in electrical and computer engineering from the King Mongkuts University of Technology Thonburi (KMUTT), Bangkok, Thailand, in 2005, 2006, and 2016, respectively.

From 2005 to 2006, he was a Research Assistant with the Thailand's National Electronics and Computer Technology Center (NECTEC). Since December 2006, he has been with the Panasonic Research and Development Center Singapore (PRD-CSG), Singapore, and he is currently a Senior Research and Development Manager. Since then, he has been working on camera processing and 3D related algorithms design. He is the author of more than 60 inventions and holds ten patents. His research interests include multi-view image/video processing, depth map estimation and 3D reconstruction, SLAM, 3D point cloud compression, digital camera image processing pipeline, computational photography, and light-weight deep learning for edge devices. Dr. Lasang is also a member of the ACM. He was a co-recipient of the IEEE Consumer Electronics Society Best Paper Award in ICCE 2010 and the 18th International Symposium on Communications and Information Technologies (ISCIT 2018) Best Paper Award, in 2018.

...