Cross-modal Matching CNN for Autonomous Driving Sensor Data Monitoring

Yiqiang Chen
RAMS Reliability Technology Lab
Huawei Technology Co., Ltd.
chenyiqiang@huawei.com

Feng Liu, Ke Pei
TTE-DE RAMS Lab
Huawei Technology Co., Ltd.
{feng.liu1, peike}@huawei.com

Abstract

Multiple sensor types have been increasingly used in modern autonomous driving systems (ADS) to ensure safer perception. Through applications of multiple modalities of perception sensors that differ in their physical properties, obtained data complement to each other and provide a more robust view of surroundings. On the other hand, however, sensor data fault is inevitable thus lead to wrong perception results and consequently endangers the overall safety of the vehicle. In this paper, we present a cross-modal Convolutional Neural Networks (CNN) for autonomous driving sensor data monitoring functions, such as fault detection and online data quality assessment. Assuming the overlapping view of different sensors should be consistent under normal circumstances, we detect anomalies such as missynchronisation through matching camera image and LIDAR point cloud. A masked pixel-wise metric learning loss is proposed to improve exploration of the local structures and to build an alignment-sensitive pixel embedding. In our experiments with a selected KITTI dataset and specially tailored fault data generation methods, the approach shows a promising success for sensor fault detection and point cloud quality assessment (PCQA) results.

1. Introduction

In recent years, researches on Autonomous Driving (AD) technologies have made significant progresses and received increasing attentions from both industry and academia. It is expected that AD systems could drastically reduce the number of traffic accidents caused by human carelessness and negligences. This expectation gives the safety property of AD systems in their operational domains into an unprecedented high importance.

Since the perception system captures the environment information through different types of sensors such as camera, radar, and LIDAR, inevitable sensor data faults are regarded as one of the major potential causes to the failure of the perception algorithms such as object detection, semantic segmentation, scene recognition *etc*. This exposes the intended functional safety and, furthermore, overall safety of the vehicle into great dangers and can be even life-threatening for passengers in the vehicle and other road users.

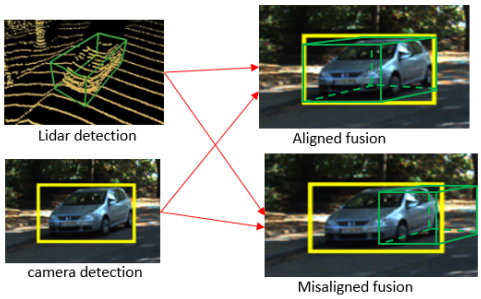


Figure 1. Impact of camera-LiDAR synchronization on object detection. The green box denotes a lidar based 3D detection bounding box projected onto the 2D image and the yellow box denotes a 2D detection bounding box. The misaligned object detections may lead to a wrong fusion result.

Sensor data faults are usually classified into two types: cross-sensor fault and single sensor fault. Single sensor fault patterns are extensively studied in the literature [9]. Single sensor faults could be either caused by internal malfunctions (e.g., broken lens, defect transmitter/receiver) or disturbing external factors (e.g., occlusions).

One of the widely accepted methods to tackle sensor faults of safe-critical systems is sensor redundancy. But this might lead to the cross sensor fault caused by either misalignment from wrong extrinsic calibration [28] or missynchronization [17] (see Fig. 1). When two sensors, for instance a camera and a LIDAR, take two independent measurements of environment, the measurements need to be aligned and happen at the same time in order to fuse the measurements to build an accurate perception of the environment. Without proper synchronization and calibration, multiple sensors could provide inaccurate, ambiguous view of the environment, leading to potentially catastrophic outcomes.

In order to improve safety guarantees and avoid afore-

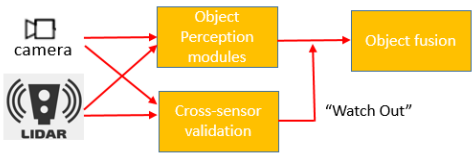


Figure 2. The cross-sensor matching module runs in parallel with the perception pipeline. An alert will be sent to sensor fusion module when an inconsistency is detected between different sensors.

mentioned issues, it is the key to design and implement a sensor monitoring system as an input verification for machine learning based perception tasks. This monitoring system is expected to provide the capability of self-assessment to the perception system. It helps to improve safety and robustness during the deployment through monitoring the performance continuously and allows to take preventive measures when the data quality below an expected level, *e.g.* changing sensor fusion strategy (Fig. 2). This is a crucial prerequisite, the quality of an automated driving functions strongly depends on the reliability of the perception data.

The most of state-of-the-art methods [9] concentrate on detecting a specific single sensor fault, in this work, we propose a novel sensor data monitoring method detecting both unspecific single and cross data fault for AD perception and fusion systems. We match sensor data from different modalities by using a cross-modal Siamese CNN. The CNN is trained by a proposed pixel-wise metric learning loss to learn a common latent space in which both modalities are projected and corresponding LIDAR point and image pixel pairs are closer than mismatched pairs. Different from the cross modal retrieve problem, the fine-grained feature matching is more crucial than the scene level semantic matching for cross-sensor data validation and monitoring, since the misalignment in real scenarios can be very subtle. Thus we propose that the cross modal joint embedding is extracted on the pixel level to explore local features and fine-grained structures in order to enforce the discriminability of alignment.

One of the major challenges of learning-based fault detection methods is the difficulty of collecting the fault-perturbed data. Real faults are comparatively rare events and randomly occur generally at very low frequency. It's hard if not impossible to collect a sufficient amount of real data to perform the detection task. To tackle this problem, we apply a self-supervised training mode in which we randomly generalize misaligned sensor data. We experimentally show that the detection can work not only for the generalized misaligned fault types but also for unseen single sensor faults.

Moreover, we propose the point cloud-image distance to be considered as reduced-reference point cloud quality assessment (RR-PCQA) under the premise of the alignment of sensors and the quality of RGB image. RR-PCQA aims to evaluate the quality of a distorted point cloud through partial information of the corresponding reference. We consider the RGB image of the overlapping field of view as such partial information. With further experiments, we show this distance could have a proper correlation with the performance degradation of the LIDAR based 3D object detection model. This proves further that our proposed measurement could be an appropriate monitoring method for autonomous driving perception and fusion systems.

To summarize, the main contributions of our paper are followings: 1) A cross-modal matching CNN is first applied for autonomous driving sensor data fault detection and monitoring. And a masked pixel-wise contrastive loss is designed in order to better explore local features and structures for matching task. 2) Unlike most fault detection methods working only for specific fault types and requiring fault data, our method could work for unseen fault types and train without collecting fault data thanks to the self-supervised learning procedure. 3) Under guarantee of alignment of the sensors and image quality, we show that the proposed point cloud-image distance can be employed as traffic scene RR-PCQA method which is barely studied in the state of the art.

2. Related work

Sensor fault analysis and detection. Sensor data monitoring is one of the key factors of the perception system monitoring mechanism. To fulfil requirements on ensuring the intended functionality of sensors, many single sensor fault analysis and detection methods have been proposed in the literature. For LIDAR sensor, contamination is one of the common sensor fault. Contaminations or damages on the sensor in the front reduce the amount of transmitted light both in the sender and the receiver path. Rivero *et al*. [19] examined the effect of dirt on the performance of a LIDAR and analyzed the sensor's uncertainty in raw data position measurements. James *et al*. [14] classified and detected different types of contaminations of LIDAR using a deep learning approach. They artificially contaminated LIDAR sensor for data generation and train a deep neural network following a multi-view concept. Some other work investigate noise of LiDAR point cloud. Segata *et al*. [23] analyzed a set of estimated distance traces obtained with a LiDAR sensor and develops a stochastic error model. Xie *et al*. [27] proposed a suitable and practical method of calculating the LIDAR signal-to-noise ratio. Diehm *et al*. [4] studied crosstalk noise in the acquired data and proposed data-based spatio-temporal filtering.

The online quality monitoring for camera images can be performed by no-reference image quality assessment (NR-IQA) which is an extensively studied research topic. Quality assessment is applied to ensure and improve the qual-

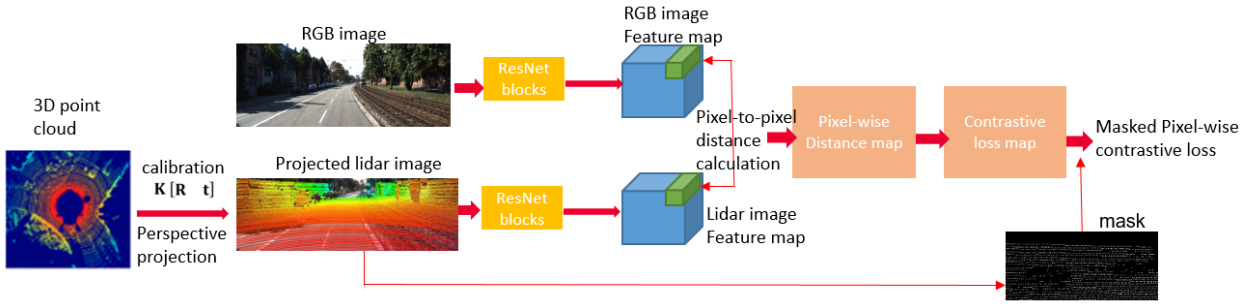


Figure 3. Overview of the cross-modal matching CNN architecture. CNN feature extractions are performed prospectively on the RGB image and the projected point cloud image. The distance map is calculated on each pixels on the two result feature maps. The loss is calculated based on each point of the distance map over a sparsity mask.

ity of visual contents delivered to the end-users. Saad *et al*. [20] leveraged the statistical features of discrete cosine transform (DCT) to estimate the NR-IQA values. In [1], Bianco *et al*. pre-trained a deep model on the large-scale database for image classification task and then fine-tuned it for NR-IQA task. For especially camera fault in autonomous driving context, [5] showed that on moving systems most artifacts can be detected by analyzing the frames in a sequence of images from a camera for static image parts. [2] [3] used CNN to detect soiling degradation of vehicle fisheye camera. Qiu *et al*. [18] studied blurry artifacts of cameras caused by defocus, motion and haze. Generated synthetic data are used for training the CNN.

Compared to single sensor fault analysis, cross sensor fault analysis and detection is seldom studied in the literature. RegNet [21] perfomed online calibration of camera and LIDAR by a CNN which can also be used as a sensor monitoring module. Zhu *et al*. [15] uses the deviation of object detection of two sensors for cross-validation.

Point cloud data assessment. Refer to the consensus in IQA, there are three different types of PCQA metrics, namely, full-reference (FR), reduced-reference (RR) and no-reference (NR) metrics. The state-of-the-art methods mainly focus on the FR-PCQA and apply to single object point cloud data, for example point-to-plane distance [25]. With absence of no distortion data, FR-PCQA is not applicable for online monitoring task. NR-PCQA and RR-PCQA are barely proposed in the state-of-the-art. To the best of our knowledge, our proposed method is the first to tackle the traffic scene PCQA.

Deep metric learning and Cross modal matching. The goal of metric learning is to quantify the similarity among samples using an optimal distance metric. Contrastive loss [10] and triplet loss [22] are two basic types of loss functions for deep metric learning. With a similar spirit of increasing and decreasing the distance between similar and dissimilar data samples, respectively, the former one takes pairs of sample as input while the latter is composed of triplets. Deep metric learning has proven effective in a wide variety of computer vision tasks, such as person re-identification [29], image retrieval [26] and face recognition [22].

However, most of existing metric learning methods are designed for unimodal matching, which cannot effectively model the relationship of features captured from different modalities. Cross modal matching is mostly applied to multimodal data for vision and language matching. For example, Liong *et al*. [16] introduced a deep coupled metric learning that designs two nonlinear transformations to reduce the modality map. Frome *et al*. [7] proposed a deep visual semantic embedding model mapping visual features and semantic features into a shared embedding space, using a hinge rank loss as the objective function.

3. Method

Our approach is based on a consistency assumption: when all sensors work well, their contents should be consistent in some way. Once inconsistency occurs, it is caused by either single-sensor or cross-sensor fault. From a security perspective, a further possibility could be an attack that disrupts the proper behavior of the sensor. Additionally we assume a very low probabilistic event that both sensors fail simultaneously and still show consistency between their data. Similarly, attacking two different sensors and keeping consistent should be extremely hard and therefore as unlikely occurrences.

The overall procedure of our CNN is shown in Fig. 3. Our method can generally be applied to different sensor modalities and combinations, in this paper, we focus on the matching between LIDAR point cloud and camera image due to popular applications in automated vehicles. In the following sections, we introduce the network architecture, the proposed pixel-wise contrastive loss, and network training details.

3.1. Siamese network and Contrastive loss

First we describe the classic contrastive loss which is the most commonly used loss function in Siamese architecture [10]. The objective is to minimize the distance between a positive pair and separate any two negative pairs with a distance margin. The contrastive loss is defined as:

$$E_{contrastive} = \frac{1}{N}\sum^{N} y \left\| f(x_i) - f(x_j) \right\|^2 + (1-y) \cdot max(m - \left\| f(x_i) - f(x_j) \right\|, 0)^2$$

Where x_i, x_j are input pair. m is a constant margin. y is the label of the input pair: $y = 1$ for positive pairs and $y = 0$ for negative pairs, f is the projection of neural networks.

The parameters are shared between the sub-networks in a Siamese architecture, that means the weights of the two sub-networks are the same and updated in the same way. Weight sharing guarantees that the learned distance metric is symmetric, *i.e.* $d(a, b) = d(b, a)$. The gradient is additive across the sub-networks due to the weight sharing.

3.2. Network architecture

The goal of this work is to detect and monitor the input data for the perception systems. To this end, we leverage the strength of CNN for feature extraction. We design the architecture inspired by classic Siamese neural networks [10] and the cross-modal calibration deep networks [21], [13].

The network takes as input an RGB image, the corresponding LiDAR point cloud, and the camera calibration matrix. The point cloud is first converted into a sparse depth map as a pre-processing step. This is done by projecting the LiDAR point cloud onto the image plane using the perspective transformation.

Then we normalize both the RGB image and the sparse depth map as input of the CNN. The network primarily consists of two asymmetric branches, each performing a series of convolutions (see Fig. 3). The weight of the two branches is not shared, since our cross-modal metric is not symmetric. For the RGB branch we use the first three blocks of the ResNet-18 network [11] (the first convolution and two following ResNet blocks). For the depth branch, we use a same architecture as the RGB stream, but the first convolution has single input channel. In ResNet-18, high-level semantic information is encoded with consecutive downsampling operations for image classification task. However, this procedure weakens the spatial capacity. The reason to use only a few low-level layers of ResNet-18 for network backbones is two-fold. First, this helps to saving the computation cost, since the resource used for monitoring function should be limited. Secondly, to preserve sufficient spatial details which are crucial for our matching task.

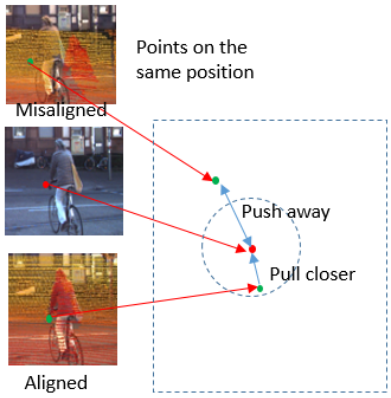


Figure 4. Cross-modal matching learns a shared embedding space where RGB image features and LIDAR point cloud features can be compared. Points with the same color are from the same modality.

3.3. Masked pixel-wise contrastive loss

Joint embedding learning aims to find a shared latent space under which the embeddings of images and LIDAR point clouds can be directly compared (see Fig. 4). Contrastive loss is utilized to encourage the distance of matched point cloud-image pairs to be smaller than mismatched pairs on this common latent space.

Unlike the classic Siamese networks that calculates the contrastive loss for pair of image instances, our loss calculation is performed at pixel level (see Fig. 3). It means that feature embedding is extracted for each point and the distance is measured between pixels on the latent feature space. This motivates the network to explore the fine-grained spatial information and restrict attention to the local structures which is important for subtle matching task. To this end, a distance layer is applied to calculate the pixel-to-pixel distance of each RGB image feature map and projected point cloud feature map. Contrastive loss is then calculated on each point on this distance map.

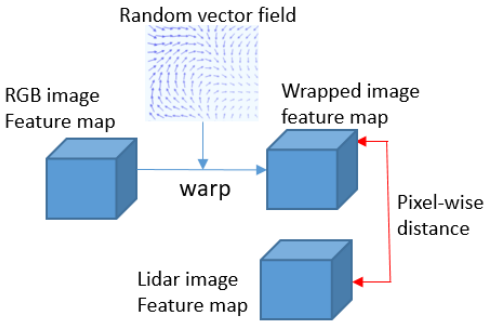
Since only part of the pixels on RGB image plane get a corresponding LIDAR point projection, the projected point cloud image is sparse. It is meaningless that compare a RGB pixel to a blank position. Thus, we extract a sparsity mask encoding which position is projected by a LIDAR point. The contrastive loss map is spatially averaged over the mask. The introduced masked pixel-wise contrastive loss is finally formulated as following:

$$E = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\|M\| \, HW} \sum_{h=0}^{H-1}\sum_{w=0}^{W-1} M_{hw} \Big[y \left\| I_{ihw} - L_{ihw} \right\|^2 + (1-y) \cdot max(m - \left\| I_{ihw} - L_{ihw} \right\|, 0)^2 \Big]$$

Where N is the batch size, M is the sparsity mask, H and W are height and weight of the feature map. I and L

(a)

(b)



(c)

Figure 5. Self-supervised negative pair generation. (a) Synthetic mis-calibrated negative pair example by applying random rotation and translation to extrinsic calibration. (b) Synthetic mis-synchronised negative pair example by timely shifting image and point cloud. (c) Wrapping feature maps by a random vector field to generate pixel-wise misaligned negative pairs.

are image and LIDAR feature embedding. y is the matching label.

For inference, the point cloud-image distance metric is obtained by spatially averaging the distance map over the sparsity mask.

3.4. Self-supervised training procedure

In order to learn a proper distance metric, both positive and negative pairs should be provided. Positive pairs are the perfectly matched pixels of projected point cloud and image. Negative pairs are mismatched pixels. Matched point cloud can be collected easily without human annotations. Real mismatched cases are extremely rare and it is impossible to collect enough representative samples. Hence, to overcome that obstacle, we propose four different approaches to generate synthetically mismatched pairs (see Fig. 5).

1. Replace the corresponding RGB image by one of the adjacent frames in the same video sequence to make mis-synchronization effect.

2. Add an extra translation and rotation to the extrinsic calibration to make mis-calibration effect, like [21], [13].

3. Use a projected point cloud and another random but not corresponding RGB image in training set to make a mismatched pair.

4. Generate a random vector field map. Each vector points to another position on the map. The feature map is wrapped following this vector field, that means we replace the point embedding on current position by the embedding of other position on feature map in order to make mismatched pixel pairs.

With the first three ways, we make negative pairs on manipulating on input data at instance level. In order to diversify further training examples, we build negative pairs at pixel level with the last method.

The random generation is performed online so that no duplicate negative pairs appear during training. This helps to avoid overfitting. And the data collection and labeling work is reduced by this self-supervised training procedure.

3.5. Training details

Our training of the proposed DNN is organized into 50 epochs with the Adam optimizer that adaptively estimates the moments. We set the parameters of the optimizer to the suggested default values. The learning rate is fixed at 10^{-3} initially. After 30 epochs, the learning rate is reduced to 10^{-4}. The input image is resized to 621×188. Random crop to 576×176 and random flip are performed.

The batch size is set to 64. To balance the number of positive and negative pairs, each batch consists of half positive and half negative pairs. Negative pairs are generated randomly by one of the four ways mentioned before. The extent is also uniformly random. The mis-synchronization range is 2˜10 frames. The decalibration translation range is 0˜0.5m and rotation range is 0˜5 degrees. The pixel displacement range is 2˜20 pixels on feature map.

It is common that the training convergence of deep metric learning can be easily compromised by the fact that the vast majority of the training samples will produce gradients with magnitudes that are zero. Hard example mining is a common solution for this problem. In order to converge to a better result, we perform a harder example generation by reducing the upper limit of the negative example ranges mentioned above after 30 epochs. Mismatched pairs formed by a point cloud and another random image are not used any more after 30 epochs, since they are relative easy examples.

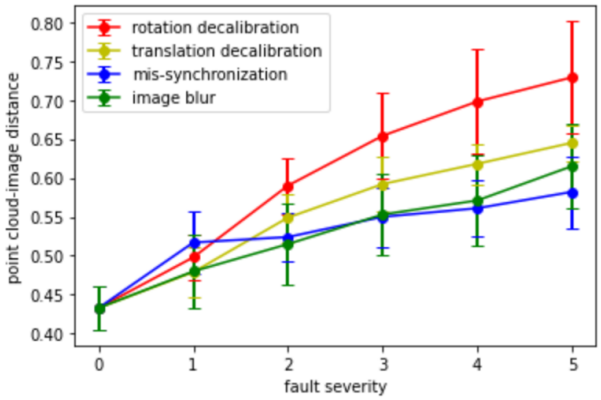
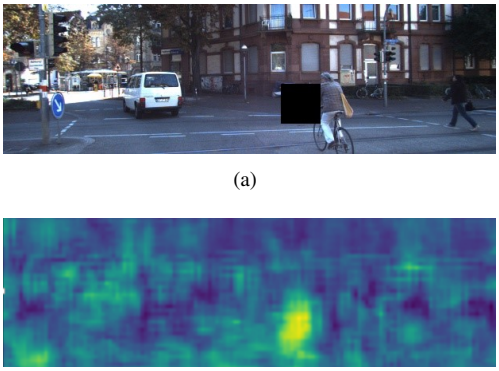


Figure 6. The evolution of point cloud-image distance for different severities of rotation/translation decalibration, mis-synchronisation and image blur. The severity levels 1˜5 correspond to 1˜5 degrees rotation decalibration, 0.1˜0.5m translation decalibration, displacement of (2,4,6,8,10) frames, and Gaussian blur of kernel size (7, 9, 11, 13,15) and sigma (2, 2.5, 3, 3.5, 4).



(a)

(b)

Figure 7. Occlusion case examples. (a) The input image with synthetic occlusion. (b) The corresponding feature distance map.

4. Experiments

In this section, we provide systematic analysis of the proposed approach with empirical experiments based on open datasets and synthetically generated mis-alignment between selected combination of sensor data. We conclude with qualitative as well as quantitative results on LiDAR-camera data combination specifically from the KITTI [8] autonomous driving benchmark. We evaluate our proposed matching method from two different aspects: sensor fault detection and PCQA. The experimental details and obtained results are presented in the following sections.

4.1. Dataset

We evaluate our approach on the KITTI dataset [8], which contains both RGB images and Velodyne LIDAR point cloud sequences. The calibration matrix are also provided. Since we need consecutive frames to generate mis-synchronisation samples, the raw recordings of the KITTI dataset are taken. We employ the 26/09 driving sequences, since they consist of a high number of sequences with good scene variation. We randomly choose 40 sequences for training, 2 sequences for validation and 3 sequences for test.

4.2. Sensor fault detection

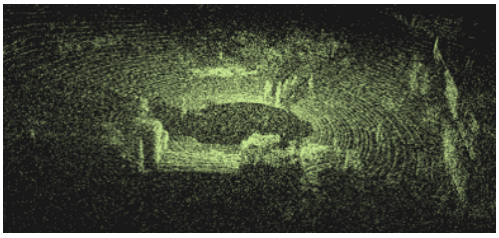
4.2.1 Qualitative point cloud-image distance analysis under misalignment and noise

First we investigate the case with cross sensor fault. Different levels of mis-synchronisation and mis-calibrations are introduced to the whole test set. We then investigate the single sensor fault, which is not seen in the training procedure. For camera fault, we add to image different levels of Gaussian blur and black regions which simulate lens occlusion. The corresponding mean and standard deviation of the point cloud-image distance on the test set suffered from different levels of cross sensor fault and camera fault are shown in Fig 6. And an example of distance map with occulted image is shown in Fig 7. To simulate LIDAR fault, we introduce two types of point cloud noises. The first one is adding n% noisy points to the point cloud. The position of the noisy point follow the Gaussian distribution of the original point cloud. The second one is perturbing n% points. The points are randomly moved 0˜0.5m along a random direction (see Fig 8). In the similar way, we calculate the mean and standard deviation of the point cloud-image distance on the test set suffered from different levels of point cloud noise. The noise level and the distance relations are shown in the Fig 9. From all these curves, the proposed point cloud-image distance shows an obvious increase while adding either cross or single sensor fault. For the occlusion case, we can find a corresponding highlighted region on the distance map which reveals a severe mismatch. Overall, we can conclude the output distance of our CNN is an reasonable metric to measure the matching levels.

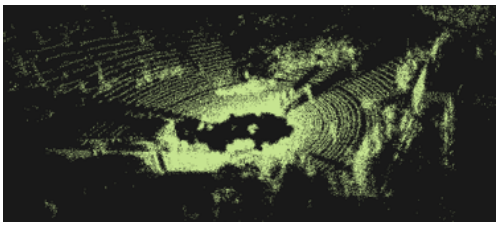
4.2.2 Sensor fault detection evaluation

In the next, we evaluate quantitatively the fault detection performance of our method. We form a fault detection test set which includes half positive examples from the raw Kitti test sequences and half negative examples from the test sequences with fault injection of random type and level. The fault types include mis-synchronization, mis-calibration, image blur, point cloud noise and perturbation.

Since state-of-the-art methods only handle specific type of fault. There is no approach close to our general fault detection dealing with both unspecific single and cross sensor fault. Therefore, we set up different variants of loss and features of our method for comparison, which is shown in Tab 1. We introduced the Area under precision-recall



(a)



(b)

Figure 8. Point cloud noise examples. (a) Adding noisy points. (b) Perturbing position of part of LIDAR points.

Methods	P-R AuC
R18-C1 feat map + instance-level loss	87.1
R18-C3 feat map + masked pixel-wise loss	86.8
R18-C1 feat map + pixel-wise loss w/o mask	93.9
R18-C1 feat map + masked pixel-wise loss	**98.1**

Table 1. Sensor fault detection results with variants of feature map and loss functions

Curve (AuC) for evaluation. Compared to using deeper ResNet-18 feature layers (R18-C3), classic instance level contrastive loss and the proposed loss function without the mask, our proposed full method shows a superior performance. This shows the pixel-wise loss and lower level features are effective to extract spatial details for cross-modal matching. And the sparsity mask is useful to filter valid regions for matching. This proves the effectiveness of our model architecture and demonstrates that our approach is suitable for online sensor data monitoring task.

There are some points to notice. First, the single sensor faults we add are all synthetic, but we can consider that they have similar effects as real faults, since they can degrade the detector's performance. These faults can be detected without being seen in the training set, we can say that there is no over-fitting to a specific fault type and the model could generalize well to other real sensor faults. Secondly, slight misalignment or noise will probably not degrade significantly the perception performance. The distance threshold for monitoring alert should be defined in function of robustness of the sensor fusion module by some offline validation tests.

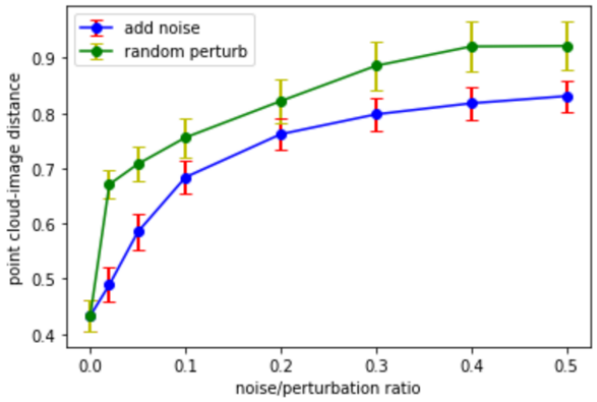


Figure 9. The evolution of point cloud-image distance over different levels of point cloud noise.

4.3. Reduced-reference Point cloud Assessment

The introduced sensor fault detection is sensitive to both single and cross data fault and to even unseen fault. But the downside is that it's difficult to localize the source of fault. In fact, image data fault can be generally detected by various IQA approaches and sensor data mis-alignment can be revealed by online calibration approaches [12]. After excluding these two cases, point could data fault analysis lacks necessary means. Hence, another possible application of the point cloud-image distance is the PCQA task. In this section, we evaluate the proposed approach as PCQA metric.

4.3.1 Full referenced point cloud distance

Since there is no NR-PCQA or RR-PCQA for point cloud of outdoors street scene in the state-of-the-art. We refer to the common used point cloud distance for comparison. The Chamfer Distance [6] between two point clouds is defined as the sum of squared distances of the nearest points between the two clouds.

The Earth Mover's Distance (EMD) [6] is originally a measure of dissimilarity between two multidimensional distributions. A bijection between two point clouds is solved for each point. The distance is defined as the sum of squared distances of the corresponding points.

4.3.2 Evaluation metrics

Following most IQA works, two evaluation criteria are adopted in our paper: the Spearman's Rank Order Correlation Coefficient (SROCC) and the Linear Correlation Coefficient (LCC). SROCC is a measure of the monotonic relationship between the ground-truth and model prediction. LCC is a measure of the linear correlation between the ground-truth and model prediction.

4.3.3 Relation to Object detector performance drop

For IQA task, the assessment standard based on human subjective perception. So the IQA datasets are normally annotated by human. But it is difficult for human to perceive point cloud quality. Unlike the color images whose viewer is human, the end user of point cloud is perception algorithms, for example, 3D object detection. So in this work, we propose a new method to evaluate PCQA. We use the performance drop of an object detector over test set as assessment ground-truth. The larger performance drops, the lower quality of point cloud.

We employ PointRCNN [24] as our reference object detection method. PointRCNN is two-stage object detection approach directly from raw point cloud: The first stage for proposal generation and the second for proposal refinement. We add different levels of noise to validation set and evaluate the PointRCNN to get the performance drops. The obtained noise level and detector performance drop relation is shown in Fig 10. Note that here the training and validation data used for PointRCNN is KITTI object detection benchmark, since there is no object detection ground-truth for raw data sequences. It is of minor importance, since our objective is to explore the noise level and performance drop relationship.

4.3.4 Evaluation result

From the Fig 8 of the previous section, we got already the relation between our distance metric and the noise levels. Based on that, we build the relation between our distance metric and the performance drop, shown in Fig 11. We can see a qualitatively linear relation between these. Furthermore, we use SROCC and LCC to evaluate quantitatively this correlation. We repeat the same experimental procedure to the chamfer distance and EMD to make a comparison. To save computation time, we use only 30% points of a point cloud to calculate chamfer distance and EMD distance. The results are shown in Tab 2. We can conclude that our point cloud-image distance has a more considerable correlation with the performance drop and faster execution than chamfer distance and EMD. This proves that our approach is a fair RR-PCQA method for online monitoring.

distance metric	LCC	SROCC	GPU time
Chamfer distance [6]	0.536	0.620	16.0s
EMD [6]	0.528	0.429	21.3s
Our pc-image distance	**0.843**	**0.920**	**3ms**

Table 2. The point cloud data assessment results and execution time for distance measure of one single example. Proposed method compared to other point cloud distances.

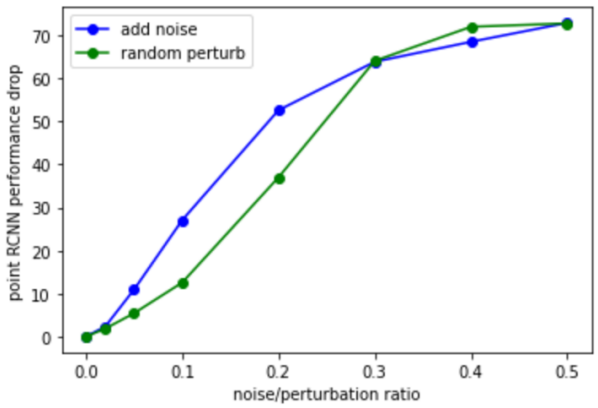


Figure 10. The evolution of mAP score drop of point RCNN over different levels of point cloud noise.

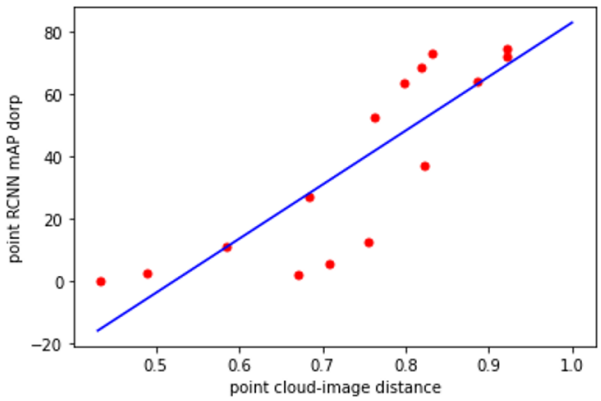


Figure 11. The point cloud-image distance and point RCNN mAP drop relation curve.

5. Conclusion and future direction

In this paper, we present a novel approach for autonomous driving sensor data monitoring. The novelty of our approach is a cross-model matching CNN trained with a masked pixel-wise contrastive loss. This loss function directs the model to optimize according to both local features and structures, it guides pixel embedding towards cross-modal alignment-sensitive representation. To address the scarcity of data containing faults, we also proposed viable methods to generate plausible faulty sensor data in order to make the training procedure fully self-supervised. The learned point cloud-image metric shows promising fault detection results with cross-sensor fault and unseen single sensor fault. Furthermore, we empirically show that our model can work as a RR-PCQA for autonomous driving scene point cloud. As one of the future directions, it is interesting to explore and incorporate contrastive learning methods such as the normalized temperature-scaled cross entropy loss to this task.

References

[1] Simone Bianco, Luigi Celona, Paolo Napoletano, and Raimondo Schettini. On the use of deep learning for blind image quality assessment. *Signal, Image and Video Processing*, 12(2):355–362, 2018.

[2] Arindam Das. Soildnet: Soiling degradation detection in autonomous driving. *arXiv preprint arXiv:1911.01054*, 2019.

[3] Arindam Das, Pavel Křížek, Ganesh Sistu, Fabian Bürger, Sankaralingam Madasamy, Michal Uřičář, Varun Ravi Kumar, and Senthil Yogamani. Tiledsoilingnet: Tile-level soiling detection on automotive surround-view cameras using coverage metric. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2020.

[4] Axel L Diehm, Marcus Hammer, Marcus Hebel, and Michael Arens. Mitigation of crosstalk effects in multi-lidar configurations. In *Electro-Optical Remote Sensing XII*, volume 10796, page 1079604. International Society for Optics and Photonics, 2018.

[5] Nils Einecke, Harsh Gandhi, and Jörg Deigmöller. Detection of camera artifacts from camera images. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 603–610. IEEE, 2014.

[6] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[7] Andrea Frome, Greg Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. 2013.

[8] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[9] Thomas Goelles, Birgit Schlager, and Stefan Muckenhuber. Fault detection, isolation, identification and recovery (fdiir) methods for automotive perception sensors including a detailed literature survey for lidar. *Sensors*, 20(13):3662, 2020.

[10] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Stephanie Hold, Steffen Gormer, Anton Kummert, Mirko Meuter, and Stefan Muller-Schneiders. A novel approach for the online initial calibration of extrinsic parameters for a car-mounted camera. In *2009 12th International IEEE Conference on Intelligent Transportation Systems*, pages 1–6. IEEE, 2009.

[13] Ganesh Iyer, R Karnik Ram, J Krishna Murthy, and K Madhava Krishna. Calibnet: Geometrically supervised extrinsic calibration using 3d spatial transformer networks. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1110–1117. IEEE, 2018.

[14] Jyothish K James, Georg Puhlfürst, Vladislav Golyanik, and Didier Stricker. Classification of lidar sensor contaminations with deep neural networks. In *Proceedings of the Computer Science in Cars Symposium (CSCS)*, page 8, 2018.

[15] Zhu Jiajun, DolgovChristopher Dmitri, and Urmson Paul. Cross-validating sensors of an autonomous vehicle.

[16] Venice Erin Liong, Jiwen Lu, Yap-Peng Tan, and Jie Zhou. Deep coupled metric learning for cross-modal matching. *IEEE Transactions on Multimedia*, 19(6):1234–1244, 2016.

[17] Shaoshan Liu, Bo Yu, Yahui Liu, Kunai Zhang, Yisong Qiao, Thomas Yuang Li, Jie Tang, and Yuhao Zhu. The matter of time–a general and efficient system for precise sensor synchronization in robotic computing. *arXiv preprint arXiv:2103.16045*, 2021.

[18] Jiaxiong Qiu, Xinyuan Yu, Guoqiang Yang, and Shuaicheng Liu. Deepblindness: Fast blindness map estimation and blindness type classification for outdoor scene from single color image. *arXiv preprint arXiv:1911.00652*, 2019.

[19] Jose Roberto Vargas Rivero, Ilir Tahiraj, Olaf Schubert, Christoph Glassl, Boris Buschardt, Mario Berk, and Jia Chen. Characterization and simulation of the effect of road dirt on the performance of a laser scanner. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2017.

[20] Michele A Saad, Alan C Bovik, and Christophe Charrier. Blind image quality assessment: A natural scene statistics approach in the dct domain. *IEEE transactions on Image Processing*, 21(8):3339–3352, 2012.

[21] Nick Schneider, Florian Piewak, Christoph Stiller, and Uwe Franke. Regnet: Multimodal sensor registration using deep neural networks. In *2017 IEEE intelligent vehicles symposium (IV)*, pages 1803–1810. IEEE, 2017.

[22] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[23] Michele Segata, Renato Lo Cigno, Rahul Kumar Bhadani, Matthew Bunting, and Jonathan Sprinkle. A lidar error model for cooperative driving simulations. In *2018 IEEE Vehicular Networking Conference (VNC)*, pages 1–8. IEEE, 2018.

[24] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019.

[25] Dong Tian, Hideaki Ochimizu, Chen Feng, Robert Cohen, and Anthony Vetro. Geometric distortion metrics for point cloud compression. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3460–3464. IEEE, 2017.

[26] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1386–1393, 2014.

[27] Chenbo Xie and Jun Zhou. Method and analysis of calculating signal-to-noise ratio in lidar sensing. In *Optical Technologies for Atmospheric, Ocean, and Environmental Studies*, volume 5832, pages 738–746. International Society for Optics and Photonics, 2005.

[28] Yi Yang. Automatic online calibration between lidar and camera, 2019.

[29] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Deep metric learning for person re-identification. In *2014 22nd International Conference on Pattern Recognition*, pages 34–39. IEEE, 2014.