## Bionotes

**Felix Anhalt**
Chair of Automatic Control, Department of Mechanical Engineering, Technical University of Munich, Boltzmannstr. 15, 85748 Garching/Munich, Germany
**felix.anhalt@tum.de**

Felix Anhalt received the Master's degree in mechanical engineering from Technical University of Munich in 2016. Since 2017 he has been working as scientific staff at the Chair of Automatic Control at TU Munich. His current research interests are in cloud-supported preview control of active suspension systems.

**Boris Lohmann**
Chair of Automatic Control, Department of Mechanical Engineering, Technical University of Munich, Boltzmannstr. 15, 85748 Garching/Munich, Germany
**lohmann@tum.de**

Boris Lohmann received the Dipl.-Ing. degree in electrical engineering and the Ph. D. (Dr.-Ing.) degree in electrical engineering from the University of Karlsruhe, Karlsruhe, Germany, in 1987 and 1991, respectively. He is a Full Professor and the Head of the Chair of Automatic Control with the Department of Mechanical Engineering, Technical University of Munich, Germany. His research interests include linear and nonlinear control systems design, modeling and model reduction, autonomous driving as well as applications in mechatronics and automotive.

DE GRUYTER OLDENBOURG at – Automatisierungstechnik 2021; 69(6): 499–510

**Methods**

Felix Berens*, Stefan Elser and Markus Reischl

# Evaluation of four point cloud similarity measures for the use in autonomous driving

## Bewertung von vier Punktwolkenähnlichkeitsmaßen für die Anwendung im Autonomen Fahren

https://doi.org/10.1515/auto-2020-0140
Received August 28, 2020; accepted January 19, 2021

**Abstract:** Measuring the similarity between point clouds is required in many areas. In autonomous driving, point clouds for 3D perception are estimated from camera images but these estimations are error-prone. Furthermore, there is a lack of measures for quality quantification using ground truth. In this paper, we derive conditions point cloud comparisons need to fulfill and accordingly evaluate the Chamfer distance, a lower bound of the Gromov Wasserstein metric, and the ratio measure. We show that the ratio measure is not affected by erroneous points and therefore introduce the new measure "average ratio". All measures are evaluated and compared using exemplary point clouds. We discuss characteristics, advantages and drawbacks with respect to interpretability, noise resistance, environmental representation, and computation.

**Keywords:** 3D image processing, point clouds, similarity measures, depth map evaluation

**Zusammenfassung:** In vielen Bereichen ist es erforderlich, die Ähnlichkeit zwischen Punktwolken zu messen. Für das autonomen Fahren können Punktwolke zur 3D Umgebungswahrnehmung aus dem Kamerabild vorhergesagt werden, jedoch sind diese Vorhersagen fehleranfällig. Darüber hinaus fehlen Maße zur Qualitätsquantifizierung anhand der Ground Truth. In diesem Paper leiten wir Bedingungen ab, die ein Maß zum Punktwolkenvergleichen erfüllen muss und bewerten entsprechend die folgenden Maße: die Chamfer Distanz, eine untere Schranke der Gromov Wasserstein Metrik und das Ratio Maß. Wir zeigen, dass das Ratio Maß nicht durch fehlerhafte Punkte beeinflusst wird und stellen deswegen das neue Maß "average ratio" vor. Alle Maße werden anhand beispielhafter Punktwolken bewertet und verglichen. Wir diskutieren Charakteristiken, Vor- und Nachteile in Bezug auf Interpretierbarkeit, Geräuschbeständigkeit, Umgebungsdarstellung und Berechnung.

**Schlagwörter:** 3D Bildverarbeitung, Punktwolken, Ähnlichkeitsmaße, Tiefenkarten Bewertung

***Corresponding author: Felix Berens,** Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany; and Institute for Artificial Intelligence, Ravensburg-Weingarten University of Applied Sciences, Weingarten, Germany, e-mail: felix.berens@rwu.de, ORCID: https://orcid.org/0000-0003-0028-830X
**Stefan Elser,** Institute for Artificial Intelligence, Ravensburg-Weingarten University of Applied Sciences, Weingarten, Germany, e-mail: stefan.elser@rwu.de
**Markus Reischl,** Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany, e-mail: markus.reischl@kit.edu

## 1 Introduction

For many technologies, like robotics, augmented reality or virtual reality it is important to have 3D information of the surrounding area. LiDAR sensors are a way to obtain very accurate 3D information, for most applications LiDAR sensors are too expensive, though. Furthermore, most state-of-the-art LiDAR sensors that are used for autonomous driving datasets have moving parts [1, 2, 3, 4, 5].

If cheaper and more robust state-of-the-art sensor systems were used to generate a precise 3D point cloud of the environment, LiDAR sensors were replaced especially for price-sensitive applications. Sensors which are commonly used in cars include camera, RADAR, and ultrasonic sensors. However, the resolution from implemented RADAR sensors is too sparse to detect objects properly [4, 6, 7, 8] and ultrasonic sensors only work in short range [9, 10, 11]. Consequently, only camera sensors remain, from which we need to predict point clouds that represent the depth information like from a LiDAR sensor.

The information of a LiDAR sensor can be represented by a point cloud or a depth map. A depth map is an image that contains information about the distance from the surface of an object to a viewpoint. Formally we can define a depth map as a set of triplets $\{(u_0, v_0, d_0), \ldots, (u_n, v_n, d_n)\}$, where $(u_i, v_i)$ describe the position of the point within the

image frame and $d_i$ the distance between the surface and the viewpoint.

Depth estimation from stereo vision, can be reduced to find image point correspondences [12]. Estimating depth maps from a single camera image is more challenging. The problem is technically ill-posed, as a 2D scene can be projected from an infinite number of world scenes. Humans use local cues such as texture, lightning or relative scale of known objects, but also global context [13]. In recent years, several methods for estimating a depth map from a single camera image were suggested. Examples include the approach by Eigen et al. [12], which proposed a two-network stack, one that makes coarse global predictions and another that redefines these predictions. The approach suggested by Lee et al. [14] is based on an encoder-decoder scheme, with local planar guidance layers in the decoding phase. Fu et al. [15] discretize depth by a spacing-increasing discretization strategy and recast depth network learning as an ordinal regression problem. Further, there are unsupervised approaches like [16, 17, 18, 19] which base on general adversarial networks, [20, 21] that base on recurrent neural networks or Yusiong et al. [22] that based on siamese neural network.

The generated point clouds are error-prone due to missing (or estimated) 3D information, though. Safety is paramount for autonomous driving, hence these methods have to be validated. To validate the depth prediction, there is a need for measures quantifying the quality of the depth maps.

Predicted depth maps are typically evaluated by applying the absolute/squared relative difference or by the root mean square error, linear/log or scale-invariant mean squared error [12]. These measures compare the pixel values of the depth map, hence they compare a 2D representation of a 3D scene and geometric features are not taken into account. Cadena et al. [23] discuss various limitations of these measures for the evaluation of predicted depth maps. Eigen et al. [12] predict the depth maps with a lower resolution than the input image, one has to upsample the predicted depth map or downsample the ground truth depth map. During upsampling, new information is created by interpolation which has to be checked against the ground truth (if available). Furthermore, a problem in predicting a depth map for outdoor scenes is that the ground truth may contain spaces not having any depth information (i. e., the sky or objects being too far away).[1] Because of these problems, common evaluation measures

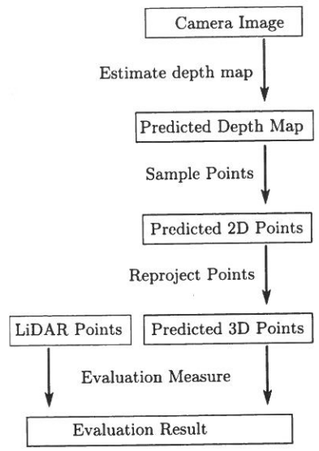



**Figure 1:** According to [23] the workflow to evaluate an image to depth map estimation method.

only evaluate pixels that have a depth value in both the ground truth and the prediction. Thus, these common evaluation methods prefer sparse estimations over dense estimations.

To this end, Cadena et al. [23] proposed to project the depth map points back to the 3D space and compare the 3D point clouds instead. Figure 1 shows the steps of evaluating a depth map. Starting with a camera image, first the depth maps which will be used for evaluation are estimated. From every depth map a set of points $(u_i, v_i, \hat{d}_i)$ is sampled. Ideally, all pixels of the depth map are used, however the calculation of the measures takes too long and sampling interpolations are needed. Sample points can be chosen either equally distributed with a given distance between the points, or according to the ground truth LiDAR points.

The problem of comparing 3D point clouds does not only occur when evaluating depth maps estimated from images, but also in the evaluation of methods that generate new 3D point clouds [24, 25, 26], including the comparison of proteins by their atomic structures [27, 28, 29]. Comparing point clouds for depth maps evaluation is challenging because there is no natural order of the points in the point cloud, which can be used for the comparison and hence a lot of points have to be compared, leading to increased computational costs.[2]

For the evaluation of depth map estimators Cadena et al. [23] proposed the ratio between ground truth points

1 Also, in scenes from the KITTI dataset, the LiDAR point cloud only reaches up to a certain height of the image (Fig. 3).

2 In common LiDAR point cloud benchmarks, e. g., the KITTI dataset, a point cloud consists of ~ 100.000 points [3].



that can be explained and those that cannot from the predicted points (See sec. 2.2.1). In this paper, we propose three other possible measures that have not yet been used to evaluate depth estimations. The measures are the Chamfer distance, a lower bound of the Gromov Wasserstein metric, and a measure that we derive from the measure of Cadena et al. [23], which we call average ratio in the following. These measures will be compared with the measure of Cadena et al. [23]. We define several conditions that a measure must fulfil and test these conditions with case studies. For this we will use the KITTI dataset [3] and the ShapeNet dataset [30]. With these case studies we will give an overview when to use which measure and which measure is best suited for the evaluation of depth map estimators.

This paper is organized as following. In Section 2 we explain problems of comparing point clouds, discuss conditions a measure needs to fulfill, introduce case studies to analyse these conditions, and define measures for point cloud comparisons. In Section 3, we evaluate the quality of the measures. These results are used to check which measures can be used in which scenario. Section 4 gives a discussion about advantages and drawbacks of all the measures.

## 2 Methods

In the following we will denote a point cloud as a finite set of points $X = \{x_0, \ldots, x_n\} \subset \mathbb{R}^3$, where every point $x_i$ is a vector of dimension 3. The well known euclidean distance between two points is denoted as $d(x_i, x_j)$.

### 2.1 Conditions on the measures

The evaluation measure for estimating point clouds must not favour two extreme cases. The first extreme case is that the estimator only predicts one point and this point very accurate. The second extreme case is that the estimator predicts a dense point cloud such that all ground truth points are near a predicted point. Hence, conditions on the evaluation measures have to prevent these two extreme cases.

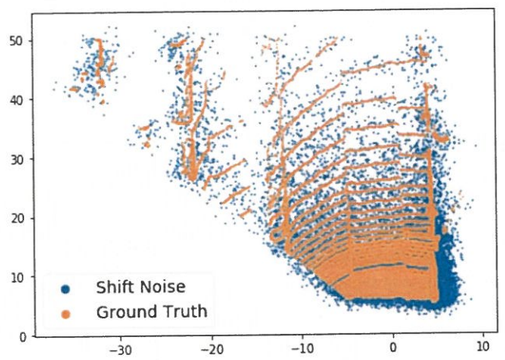
A measure that is used to compare two point clouds for depth estimation must:

1. measure how good the ground truth points are represented in the predicted point cloud,
2. punish predicted points that are far away from ground truth points,
3. give two geometric similar point clouds a smaller measure value than two geometric dissimilar point clouds and
4. be fast to calculate regarding the number of points within the point clouds, because the number of points can increases rapidly.

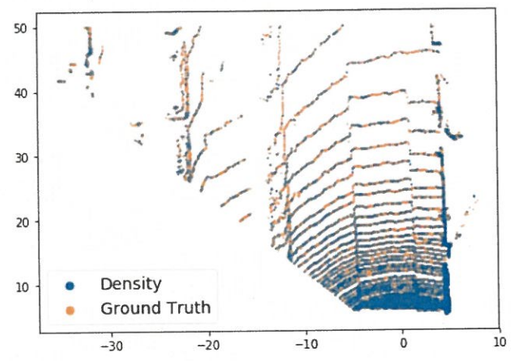
To compare and quantify advantages of the measures related to given situations, three case studies are analysed. In these cases we can control the errors of the two point clouds and we can examine how good the measures fulfil the properties we demanded. These cases are:

1. **Shift Noise:** On every coordinate point of a point cloud we add some noise (normally distributed with mean 0 and standard deviation $\sigma$ between 0 and 2). This noisy point cloud is compared with the original point cloud. With the varying $\sigma$, it will be checked how the measures react to small and large noise to the position of the points, but not totally random points. Because the LiDAR points In Figure 2a, an example of a point cloud with noise that is normally distributed $\sim N(0, 1)$ and the corresponding ground truth point cloud is shown.
2. **Density:** We randomly choose $p$ percent points of the full point cloud. This new subset point cloud is compared with the full point cloud. Thus, we compare the measures according to the density of points. This shows how well the measure can indicate how good the ground truth is represented from the point clouds, which was the first property from before. In Figure 2b, an example of a sparse point cloud and the corresponding full point cloud is shown.
3. **Random noise:** We sample 5000 points of the ground truth and add an increasing number $M$ of random points, which we uniformly sample from the 3D space, within the same range as the ground truth point cloud. The more points are added, the more outlier points are included in the point cloud. The number of inlier points may also increase very slowly, as some points are generated in accordance to the original point cloud by chance. This will show how the measure punishes points that have no near corresponding point in the ground truth. In Figure 2c, an example for this case with $M = 5000$ is shown.

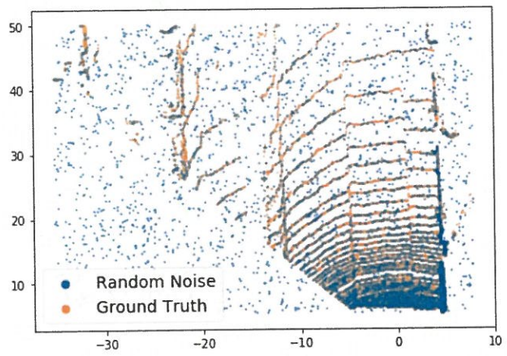
Further, we want to evaluate the outcome on 3D-object-recognition. To see whether a measure returns systematic errors with respect to geometrical similarity, we will test whether the measures can distinguish between point clouds of different categories. Objects of the same category are more geometrically similar than two objects of different




(a) Point cloud with shift noise in blue and the corresponding ground truth in orange. The added noise has a standard deviation of $\sigma = 1$. The shifted points are at random position near the original points.




(b) Sparse point cloud (5581 points) in blue and the corresponding dense ground truth point cloud in orange (18605 points). Note that the sparse point cloud still represents some parts of the original point cloud, i.e., some orange points have been covered by blue points.




(c) Point cloud with random noise in blue and the corresponding ground truth in orange without random noise. Here $M = 2000$ random points are added.

**Figure 2:** Example point clouds for the three case studies that are analysed.

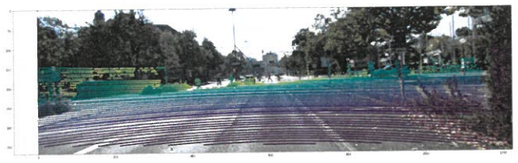


**Figure 3:** A KITTI scene with the projected LiDAR point cloud into the field of view of the camera. The color of the points indicate the depth. The LiDAR point cloud covers only the bottom part of the image.

category. To check whether similar point clouds are close to each other according to the measure, we use 1-nearest neighbor accuracy (1-NNA) on a pair of categories, based on the suggested two sample test of Lopez-Paz and Oquab [31]. As in [24, 32] for 1-NNA we calculate the nearest neighbor for every object. Then it is checked if the nearest object is from the same category. The value of 1-NNA describes the percentage of objects that nearest neighbor is from the same category. If 1-NNA is equal to 1, the next nearest point cloud is always from the same category. The closer 1-NNA is to 0.5, the more similar are the two categories according to the measure.

For analysis we need datasets of point clouds. With the datasets we want:

1. understand how the measures behave under specific errors
2. evaluating the outcome on 3D-object-recognition

Here we will use LiDAR point clouds from the KITTI dataset [3]. The KITTI dataset was published 2012, it contains 15.000 LiDAR point clouds, that consists of ~ 100.000 points. The point clouds were recorded with a Velodyne-HDL-64E in Karlsruhe, Germany. For the three case studies (Shift Noise, Density, and Random Noise) the LiDAR point clouds from KITTI are used, because they are similar to the point clouds from depth maps. We will use these LiDAR point clouds instead of predicted depth maps, because they are less error-prone and we can add the errors for the three cases studies as shown in Figure 2, without having additional unknown errors.

Further, we will use the ShapeNet dataset [30] which contains several 3D objects of different categories. The shapes of the objects are represented by 3D CAD models in the ShapeNet dataset. For the evaluation of the outcome on 3D-object-recognition we will use point clouds that we get from the shapes. We will use the categories *Airplane*, *Bottle*, *Car*, *Chair*, *Table*, and *Vessel* of the ShapeNet dataset. As we can see in Figure 4 these objects are all not



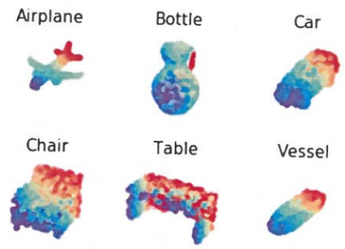



**Figure 4:** Examples of point clouds from the ShapeNet dataset [30].

true to scale and they are all aligned along the same axis, so only geometric features play a role in the comparison.

## 2.2 Point cloud measures

### 2.2.1 Ratio

The measure of Cadena et al. [23] is the percentage of ground truth points that have a predicted point whose distance is smaller than a threshold $d$:

$$\mathbf{R}_d(X, Y) := \frac{|\mathcal{S}_{d,X,Y}|}{|X|},$$

where $\mathcal{S}_{d,X,Y}$ is the set of all points in $X$ whose distance to any point of $Y$ is lower than the threshold $d$. $\mathbf{R}_d$ is bounded in $[0, 1]$. A value of $\mathbf{R}_d = 0$ means that no point of $X$ has a distance lower than the threshold $d$ to any point of $Y$, a value of $\mathbf{R}_d = 1$ means that all points of $X$ have a distance lower than the threshold $d$ to any point of $Y$. $\mathbf{R}_d$ can be calculated in $\mathcal{O}(n^2)$.

$\mathbf{R}_d$ has some drawbacks. A typical situation is shown in Figure 5a, the red crosses are predicted points for the grey surface. Points are equally sampled from the grey surface to evaluate how good these predicted points are. In blue, points are shown having a smaller distance than the threshold $d$ chosen for $\mathbf{R}_d$. Every ground truth point has a distance greater than the threshold to the top red cross. If this red cross moves along the arrow and gets further away from the ground truth points, the value of $\mathbf{R}_d$ does not change, it is just ignored. Figure 5b illustrates the choice of the threshold $d$ of $\mathbf{R}_d$ can lead to the wrong conclusion, such that the point cloud with the best value of $\mathbf{R}_d$ does not have to be the most similar point cloud to the ground truth. If we decrease the threshold in the example of Figure 5b $\mathbf{R}_d$ prefers the left prediction, which looks for a human less similar than the ground truth. Another problem of $\mathbf{R}_d$ is that we do not know if we have chosen the best parameter $d$ for the threshold.

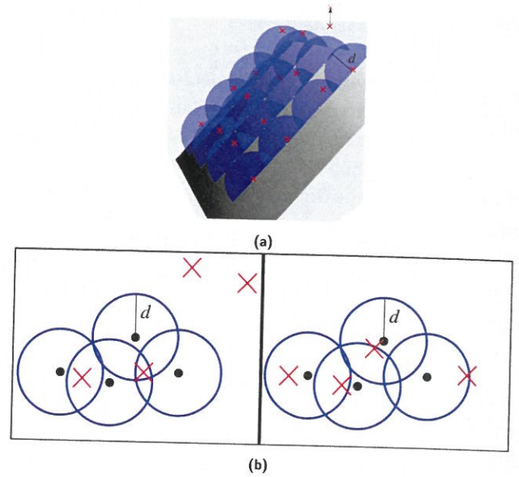



**Figure 5:** Examples for the weaknesses of $\mathbf{R}_d$. The blue balls have a radius equal to the threshold $d$ of $\mathbf{R}_d$ around the ground truth points, red cross estimated points. In (a) points are equally sampled from the gray surface. $\mathbf{R}_d$ ignores in the calculation the position of the upper cross. As this cross is in no blue ball. This cross can translate away from the ground truth, without any change in the value of $\mathbf{R}_d$. (b) Two examples, both with 4 black ground truth points and 4 estimations. In both examples it is $\mathbf{R}_1$. Intuitively, the right estimations represent the ground truth better. If the threshold is decreased $\mathbf{R}_d$ decrease for the right example first.

### 2.2.2 Chamfer Distance

The **C**hamfer **D**istance (**CD**) describes the mean minimal distance of all points in $X$ to a point in $Y$ plus the mean minimal distance of all points in $Y$ to a point in $X$ and is defined as follows:

$$\mathbf{CD}(X, Y) := \frac{\sum_y \min_{x \in X} d(x, y)^2}{|Y|} + \frac{\sum_x \min_{y \in Y} d(x, y)^2}{|X|}.$$

**CD** can be calculated in $\mathcal{O}(n^2)$. The most time consuming part is the calculation of the distance matrix between the points in $X$ and in $Y$. An advantage of **CD** is that the calculation is simple and only depends on the distance matrix between the points in $X$ and in $Y$. In the example of Figure 5b the value of **CD** is smaller for the right prediction than for the left, because of the minimal distance of the two outliers in the left prediction. A drawback of **CD** is that a single outlier can have a huge influence on the measure, especially if the number of points is small.[3]

3 **CD** is not a metric, but a semimetric.

### 2.2.3 Gromov Wasserstein

The Gromov Wasserstein metric measures the similarity between two metric measure spaces and is *np*-hard. Mémoli [33] discusses its properties and states several lower bounds of the Gromov Wasserstein metric, which are faster to calculate, than the Gromov Wasserstein metric and still give a good approximation. We will use the **l**ower bound of the **G**romov **W**asserstein distance (**LGW**), which computes the distances between distributions of eccentricities. The eccentricity of one point $x \in X$ is defined as $s_X(x) := \frac{\sum_{i=1}^{n} d(x,x_i)}{n}$, it is the mean distance of the point $x$ to every other point in the point cloud $X$. The distribution function of the eccentricity $u$ is $S_X(u) := \frac{|\{x \in X | s_X(x) \leq u\}|}{n}$. These distributions are calculated for all $L$ unique occurring eccentricities of the point clouds $X$ and $Y$, where $u_1 < \cdots < u_L$. For a LiDAR point cloud, those points that are near the sensor have a lower eccentricity than points that are far away, because near the sensor the number of the points is much higher (Fig. 6). To calculate **LGW** we use this formula:

$$\mathbf{LGW}(X,Y) := \frac{1}{2}\sum_{i=1}^{L-1} |u_{i+1} - u_i||S_X(u_i) - S_Y(u_i)|.$$

A characteristic of **LGW** is that it is invariant regarding translation, rotation,and symmetry. As a evaluation measure for depth estimation methods this can be a disadvantage, for example if the estimation method has a constant shift of all points, this is not be detectable with **LGW**. In other applications, where only the shape of objects will be compared, these invariants are an advantage. Another disadvantage of **LGW** is the same as for **CD** that a single outlier can have a huge influence. In the example of Figure 5b the distribution of the eccentricity for the left predictions is different to the ground truth points, while the right prediction has a similar eccentricity distribution. The computational costs of calculating **LGW** is in $O(n^2)$.[4]

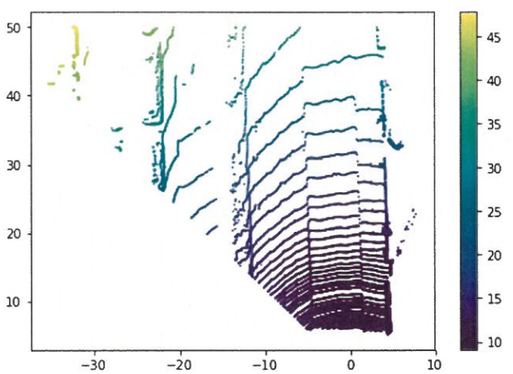



**Figure 6:** Example for the eccentricities of points from a LiDAR point cloud. Points near the LiDAR sensor have a lower eccentricity value.

### 2.2.4 Average ratio

Based on the measure $\mathbf{R}_d$ proposed in [23], we define a newly measure, which considers more than one threshold and also takes both point clouds equally into account. We denote this measure as **AR**, since it can be seen as an **A**verage of $\mathbf{R}_d$:

$$\mathbf{AR}(X,Y) := \frac{\sum_{i=1}^{N} i\frac{|\mathcal{S}_{D_i,X,Y}|}{|X|} + \sum_{i=1}^{N} i\frac{|\mathcal{S}_{D_i,Y,X}|}{|Y|}}{N^2+N},$$

where $\mathcal{S}_{D_i,X,Y}$ is the set of all points of $X$ that have a minimal distance to one point in $Y$ smaller than $D_i$. **AR** is a weighted average percentage of points that can be explained from the other point cloud according to a threshold. The first term of **AR** is a weighted average of $\mathbf{R}_d$. $N$ determines the number of thresholds $D_i$, that are taken into account, where $D_1 < \cdots < D_N$. The weight for the $i$th summand is the index $i$, thus a small threshold has less influence on **AR** than a larger threshold. The sum of all weights from both sums is $N^2 + N$. To guarantee that the values of **AR** are included in $[0,1]$ the sum is divided by $N^2 + N$. A value of 1 means that all points have a distance smaller than the smallest threshold to the other point cloud. A value of 0 means that even for the largest threshold the distance of every point to the other point cloud is larger.

In the following for the comparisons of point clouds the thresholds are defined as $D_i = \frac{2^i}{1000}$, for $i = 1,\ldots,16$. With the quadratic growth, we take smaller errors and also greater errors in the calculation. Furthermore, the biggest threshold $D_{16} = 65.536$ is so large that it is very likely that all points have a distance smaller than this threshold to the other point cloud. We will use the same set of thresholds for the comparisons of point clouds as for the comparisons of objects from the ShapeNet dataset. The computational costs are in $\mathcal{O}(n^2)$ and **AR** can be similarly calculated like **CD**, as for both measures we need the distance matrix between the points in $X$ and in $Y$.[5]

The measures **CD** and **LGW** are small for two point clouds that are similar and increase if the point clouds get more unlike, so they are negatively correlated to the similarity between two point clouds, unlike $\mathbf{R}_d$ and **AR**, that are positively correlated. In order to get the two similarity measures **CD** and **LGW** positively correlated to the similarity of two point clouds, we will have the following transformation:

$$v' = \frac{1}{1+v}, \tag{1}$$

where $v$ is the value of the measure. The transformed similarity measure values are positively correlated and also bounded between 0 and 1. In the following we will use the transformed similarity measures instead of the original values.

4 **LGW** is not a metric, but a pseudometric.

5 **AR** is not a metric.



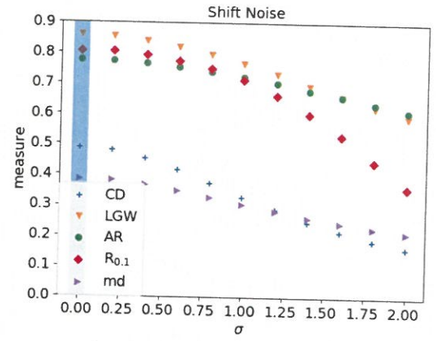
## 3 Experiments

### 3.1 Overview

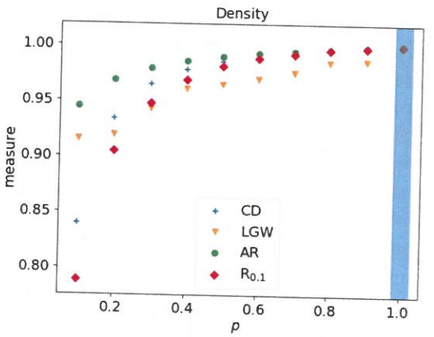
In this section we compare the measures. First we analyse the measures regarding of the case studies Shift Noise, Density, and Random Noise which are defined in Section 2.[6] After this we analyse how well the measures can distinguish point clouds from the ShapeNet dataset [30]. Last it is shown how much time the measures take for comparing point clouds of different size.

### 3.2 Comparison of the measures

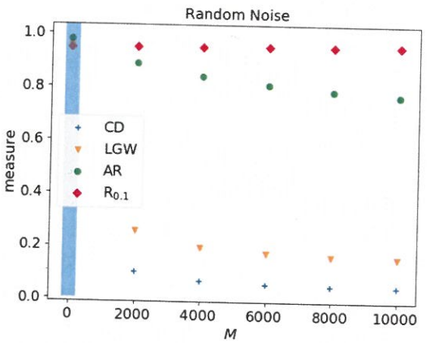
In Figure 7a, the results for the first case Shift Noise can be seen in which we add a shift to every coordinate point. **md** is the mean distance of the shifted points to the corresponding ground truth point.[7] Figure 7a shows that **md** is negatively linear correlated to $\sigma$. The values of the measures must also decrease maximally like **md**, because **md** describes the maximum divergence between the original point cloud and the point cloud with shift noise. We can see that all four measures are negatively correlated to $\sigma$. The values of $\mathbf{R}_{0.1}$ decrease much faster than **md**, but with greater thresholds it decrease slower than **md**. The reason for this is that with larger thresholds small shifts are still




**(a)** Evaluation of different measures with LiDAR point clouds with noise $\mathcal{N}(0,\sigma)$. **md** is the mean distance between a ground truth point and the corresponding point with noise.




**(b)** Evaluation of different measures with LiDAR point clouds with randomly sampled subsets of points.




**(c)** Evaluation of different measures with LiDAR point clouds with adding random points.

**Figure 7:** Results for the different cases. Blue boxes show the starting point, where no error is on the point cloud.

6 For $\mathbf{R}_d$ we have tested several thresholds $d$, but in Figure 7 we only show results for one threshold 0.1, such that the Figure remains clear. $\mathbf{R}_{0.1}$ is chosen as this threshold shows an interesting behaviour in all three cases and is in connection with depth maps for autonomous driving an important threshold, as the value of $\mathbf{R}_{0.1}$ means that this percentage of ground truth points have a distance to a predicted point smaller than 0.1 m.

7 To be more comparable to the other measures we also transformed the values $v$ of **md**, like the other three similarity measures $v = 1/(1+v)$ (1).

treated as correct and in the Shift Noise case the shifted points are still near a ground truth point and not totally random in the space. **CD** and **LGW** decrease faster than **md**, while **AR** has nearly the same decreasing rate as **md**. So **AR** and $\mathbf{R}_{0.1}$ are less effected by $\sigma$ compared to the other measures. With bigger thresholds $d$ for $\mathbf{R}_d$ the effect gets even smaller. For smaller thresholds $d$ for $\mathbf{R}_d$ or smaller $N$ for **AR** the effect on the measure grows. Hence, $\mathbf{R}_d$ and **AR** can be more robust against shift noise than **CD** and **LGW**, if a suitable parameter $d$ or $N$ is chosen.

Figure 7b shows the effects of the case Density, in which we sample $p$ percent of points from the full ground truth and measure the similarity to the full point cloud. It can be seen that all four measures become bigger with an increasing number of points. Thus, all four measures are able to evaluate how good the ground truth points are represented by the predicted points. An interesting behaviour can be seen by **LGW**, as the value of **LGW** looks like a step function. The values of **LGW** are between 0.1 and 0.2, 0.4 and 0.5 and between 0.8 and 0.9 nearly constant. Hence, the eccentricity does not change and thus geometric properties of the point cloud do not change according to **LGW**. A similar behaviour does not occur for any other measure.

In Figure 7c we analyse the case Random Noise, how well the measures punish random points. **LGW**, **CD** and **AR** are decreasing with the number of points $M$ we randomly add. $\mathbf{R}_{0.1}$ is increasing, this indicates that the point cloud with more random points is better for the ground truth point cloud, than the point cloud with fewer random points. But in fact only some points were randomly sampled near enough to a ground truth point, the majority of new points are too far away and are ignored by $\mathbf{R}_{0.1}$. The same behaviour can also be seen for other thresholds $d$ for $\mathbf{R}_d$, especially by larger thresholds. This underlines one problem of $\mathbf{R}_d$, that it ignores the position of points that are far away from the ground truth point cloud. The first term of **AR** which averages the $\mathbf{R}_d$ has a similar behaviour as $\mathbf{R}_d$. The second term in the equation of **AR**, that compares how many points of the prediction have a distance to a ground truth point smaller than a threshold, prevents that **AR** ignores points that are at the wrong position. So the three measures **LGW**, **CD** and **AR** are suitable to detect the error of these wrong points, while $\mathbf{R}_d$ cannot.

## 3.3 Differentiate point clouds of different categories

To evaluate the usefulness in practical applications, we analyse how well the measures can distinguish between objects from different categories. For this, we randomly sample $n$ points from objects from the ShapeNet dataset of the categories *Airplane*, *Bottle*, *Car*, *Chair*, *Table* and *Vessel*. This randomly sampling is similar to a LiDAR sensor, which also samples points from the real objects, instead of generating a complete 3D model of the area around the car.

$\mathbf{R}_d$ is not symmetric, in contrast to the other measures. We obtain two results for the comparison for $\mathbf{R}_d$, depending on the order of the point clouds. For $\mathbf{R}_d$ we tested several thresholds between $10^{-1}$ and $10^{-6}$ as we do not know which one is the best. For **AR** we choose the same parameter $N = 16$ as before. For Table 1 we calculated the mean of all 1-NNA results for every measure, except the 1-NNA results of a category with itself. The 1-NNA results for the test between two sets of objects from the same category has to be small, near 0.5, and not near 1 for the other tests between two different categories.

The correlation matrices (Fig. 8) show the 1-NNA results for $n = 1024$ points for each pair of categories. On the diagonal line of every correlation matrix the test between the same category is shown. As expected, the 1-NNA value is in this case near 0.5, which means that the measures cannot differ between the two sets of objects of the same category. Only for the category *Airplane* the 1-NNA values can be higher, probably some types of *Airplanes* only occur in one of the two sets. **CD** and **AR** differentiate well between the categories the mean of all 1-NNA values is 0.977 for **CD** and 0.975 for **AR**. The mean 1-NNA of **LGW** is 0.813 for $n = 1024$. **CD**, **AR** and **LGW** are in particular good at distinguishing between *Airplane* and another category, except for *Vessels*. Because except for the wings, which are close to the fuselage of a fighter plane, an *Airplane* and a *Vessel* are geometrically similar, both long and narrow. For **LGW**, *Airplane* and *Vessel* are peculiarly difficult to distinguish, since the eccentricities are similarly distributed for both. For a *Vessel* the high eccentricity values are at the bow and stern and for an *Airplane* at bow, stern and wing. Other categories are more compact and thus the eccentricities have a different distribution than the eccentricities for an *Airplane*.

$\mathbf{R}_{5\cdot10^{-4}}$ has the highest mean 1-NNA value for $\mathbf{R}_d$, which is 0.917 and hence worse than **CD** and **AR**, but better than **LGW**. As the correlation matrix in Figure 8 shows, $\mathbf{R}_d$ can differentiate for example *Bottle* from other categories well. This depends on the fact that the range of the points on the vertical axis is for all categories very narrow, except for the *Bottle* category, we have a wide range of the vertical axis, see Figure 4. Thus, the top points of the *Bottle* are not covered by points from the other point cloud. On the other hand $\mathbf{R}_d$ has big issues differentiating a *Car* or a *Vessel* from an *Airplane*, because the geometric shape



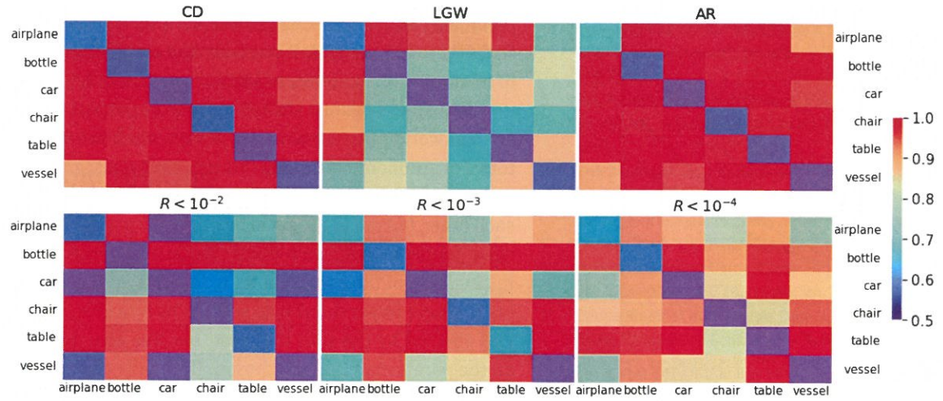



**Figure 8:** Correlation matrices of the paired application of the 1-NNA with the respective measures. The number of points was $n = 1024$, for all objects. A 1-NNA value near 0.5 indicates that the two categories are more similar according to the measure.

**Table 1:** Mean 1-NNA results for the measures, by comparing the objects from the categories *Airplane*, *Bottle*, *Car*, *Chair*, *Table* and *Vessel* from the ShapeNet dataset [30]. The 1-NNA of a category with itself is ignored, because this value has to be near 0.5 and not near 1. Bold are the best values for all measures.

| $n$ | CD | LGW | AR | $\mathbf{R}_{10^{-6}}$ | $\mathbf{R}_{10^{-5}}$ | $\mathbf{R}_{10^{-4}}$ | $\mathbf{R}_{5\cdot10^{-4}}$ | $\mathbf{R}_{10^{-3}}$ | $\mathbf{R}_{5\cdot10^{-3}}$ | $\mathbf{R}_{10^{-2}}$ | $\mathbf{R}_{5\cdot10^{-5}}$ | $\mathbf{R}_{10^{-1}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 128 | 0.977 | 0.813 | 0.975 | 0.511 | 0.616 | 0.774 | 0.883 | 0.892 | **0.892** | 0.850 | 0.616 | 0.505 |
| 512 | 0.979 | **0.848** | 0.979 | 0.619 | 0.725 | 0.878 | 0.910 | **0.912** | 0.875 | **0.838** | 0.618 | **0.507** |
| 1024 | **0.981** | 0.825 | **0.980** | **0.644** | **0.819** | **0.897** | **0.917** | 0.908 | 0.870 | 0.833 | **0.623** | **0.507** |

of a *Car* and a *Vessel* differs from an *Airplane* particularly in the wings, which are not included in the calculation of the value. We can see that the overall ability to differentiate between objects of $\mathbf{R}_d$ becomes worse for very small thresholds or large thresholds (Tab. 1). Because with a large threshold all points are counted as correct and with a very small threshold no point is counted as correct. For $\mathbf{R}_{10^{-1}}$ with $N = 1024$ the mean 1-NNA value is 0.507, which is only slightly better than a random guess. While the overall ability for $\mathbf{R}_{10^{-4}}$ is worse than for $\mathbf{R}_{10^{-3}}$, for some categories it is better with the smaller threshold $d = 10^{-4}$, for example $\mathbf{R}_{10^{-4}}$ can differ between a *Car* and an *Airplane* better than $\mathbf{R}_{10^{-3}}$ (Fig. 8). One reason for this is probably that for $\mathbf{R}_{10^{-3}}$ the surroundings around the points of the *Airplane* are large enough to enclose a point of the *Car* and therefore it is difficult to distinguish between them, while $\mathbf{R}_{10^{-4}}$ the surroundings are no longer so large that they always reach a point of the *Car*.

Table 1 shows that the quality how good categories can be differentiated gets slightly worse for **CD** and **AR** with decreasing the number of points $n$.

Surprisingly, **LGW** and $\mathbf{R}_d$ for the given thresholds $10^{-2}$, $5 \cdot 10^{-3}$ and $10^{-3}$ the highest 1-NNA value, hence the best differentiation, was not achieved with the largest tested number of points, but with fewer points. Especially, $\mathbf{R}_{5\cdot10^{-3}}$ had the best result with only $n = 128$. This stands in contrast to the differentiation with other thresholds for $\mathbf{R}_d$, **AR** and **CD**. With fewer points and sparse distribution, there is more space between the points and therefore, with a lower probability a point of the wrong category in the surrounding area of the points.

## 3.4 Empirically time measure

We have also measured the time of calculating the four measures.[8] Theoretically, they all have a time complexity of $\mathcal{O}(n^2)$, but Figure 9 shows **LGW** and **AR** take more time for calculation than **CD** and $\mathbf{R}_d$.

8 CPU @ 2.60 GHz.

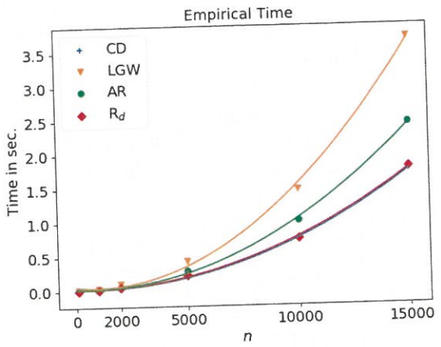



**Figure 9:** Empirically time duration for calculating the measures, for different number of points $n$. A quadratic model was used to fit the data. Note that **CD** and $\mathbf{R}_d$ are almost equal.

## 4 Discussion

A summary of all discussed properties of the measures can be found in Table 2. We will discuss the measures regarding their interpretable, handle of shift noise, false points detection, coverage of ground truth, differentiate objects, dependence on parameters and computational costs.

**Table 2:** Overview of the properties of the different point cloud measures.

| | CD | LGW | AR | $R_d$ |
|---|---|---|---|---|
| Interpretable | + | – | o | ++ |
| Shift Noise | – | – | + | o |
| False Points detection | ++ | ++ | + | – |
| Coverage of the ground truth | + | + | + | + |
| Differentiate objects (aligned) | ++ | o | ++ | + |
| Depends on parameters | ++ | ++ | o | – |
| Calculation time | ++ | – | + | ++ |

Interpretable means that we can infer from the measure value more information than just how similar the two point clouds regarding the measure are. The value of $\mathbf{R}_d$ is the easiest to interpret, since the value is the percentage of ground truth points whose distance to an estimated point is lower than the threshold. For example, a value of $\mathbf{R}_{0.1} = 0.7$ for a depth map estimator means that 70 % of all ground truth points have a predicted point that is closer than 0.1 m. **CD** is the mean of the minimal distance between the points of a point cloud to the other point cloud and can also be interpreted well. For example, a value of **CD** = 0.7 for a depth map estimator means that in average the minimal distance between ground truth points and predicted points and vice versa is 0.7 m. The values of **LGW** and **AR** are not simply interpretable and must be regarded in relation to other results of **LGW**.

Robustness against shifted noise is important, as every real world measurement has always small random errors. For autonomous driving this means that a scan of the same area can lead to point clouds that points are near to each other, but not at the same position. **CD** and **LGW** have some problems with such point clouds and overestimate the increase of the noise. For $\mathbf{R}_d$ it depends on the threshold $d$, where larger thresholds seem to be less affected. **AR** was effected by the shift the least, but this also depends on the parameters that are chosen. If we only chose small thresholds for **AR** it will lead to a greater influence of shift noise.

In False Points detection we look at the results of random noise, this is e. g., important for the evaluation of predicted depth maps. A depth map predictor with many false points will lead to an unsafe behaviour of the car. For coverage of the ground truth we look how good the measures check whether the ground truth point cloud is covered by the estimated point cloud or not. This is also important e. g., for the evaluation of depth map predictors as the target of the predictor is to generate a point cloud that points represent the ground truth. All four measures considered can check whether the ground truth point cloud is covered by the estimated point cloud, but incorrectly placed points can only be assessed with **CD**, **LGW** and **AR**.

To differentiate between objects regarding categories, is import for classification tasks, but also for the evaluation of generators point cloud objects. The best results were achieved by **CD** and **AR**, $\mathbf{R}_d$ performed acceptable, and had some surprising behaviour as for some thresholds the best differentiation was possible with fewer points. This behaviour of $\mathbf{R}_d$ needs further investigation. **LGW** has the most problems to differentiate between objects. **LGW** has one advantage as it is rotation invariant, it can differentiate rotated objects equally good as non rotated objects.

An advantage of **CD** and **LGW** is, that we do not need to chose any parameters, so **CD** and **LGW** can be used for point clouds of different scales. While $\mathbf{R}_d$ and **AR** are dependent on parameters, which are chosen depending on the range of the point clouds. So values of $\mathbf{R}_d$ and **AR** can only be compared with the same parameters or with the same point clouds. Because we can chose a set of parameters in **AR** the value is not as depending on one parameter like $\mathbf{R}_d$, so we have used the same parameter set for **AR** for the comparison of LiDAR point clouds from KITTI, which have a range of ~ 50 and the objects of ShapeNet, which have a range of ~ 1.



The most time consuming measure is **LGW**, while still faster than Gromov Wasserstein, it takes much more time for calculation than the other three measures.

## 5 Conclusions

There are several mathematical functions which can compare point clouds, but only some of them are practically useful. This paper has presented four measures $\mathbf{R}_d$, **CD**, **LGW** and **AR** to evaluate the performance of depth map estimators. We have shown that $\mathbf{R}_d$ is not suitable for this task, because it only checks how well the ground truth point cloud is covered by the predicted point cloud and depends heavily on the choice of a parameter. For the comparison of the measures we show several case studies (Shifted Noise, Density, Random Noise) and also how good these measure can be used to distinguish categories of point clouds.

In total, one can say that **CD** and **AR** are at least equally good as $\mathbf{R}_d$ and can also punish false points, that $\mathbf{R}_d$ cannot. So **CD** and **AR** are more suitable for the evaluation of depth maps than $\mathbf{R}_d$. Due to the good interpretability of $\mathbf{R}_d$ it can be used to describe how well the ground truth is represented by the prediction. **LGW** is not as good as the other three measures, but due to the rotation invariance it has its eligibility. This rotation invariant case occurs in the generation of point cloud objects, if during training the objects are rotated for data augmentation. The new generated objects are rotated and hence for evaluation a measure that is rotation invariance is needed, of those considered, only **LGW** can do this.

In addition to evaluating depth maps from cameras, these measuring methods can also be used for the evaluation of other high resolution 3D perception sensors for autonomous driving, like the high resolution RADAR of Astyx [34] or how well a set of low resolution sensors are suited for autonomous driving [5].

## References

1. Jingyun Liu et al. "TOF Lidar Development in Autonomous Vehicle". In: *2018 IEEE 3rd Optoelectronics Global Conference (OGC)*. 2018, pp. 185–190. doi: 10.1109/OGC.2018.8529992.
2. Sean Campbell et al. "Sensor technology in autonomous vehicles: A review". In: *2018 29th Irish Signals and Systems Conference (ISSC)*. IEEE. 2018.
3. Andreas Geiger et al. "Vision meets robotics: The kitti dataset". In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237.
4. Holger Caesar et al. "nuscenes: A multimodal dataset for autonomous driving". In: *arXiv preprint arXiv:1903.11027* (2019).
5. Jakob Geyer et al. "A2D2: Audi Autonomous Driving Dataset". In: (2020). arXiv:2004.06320 [cs.CV]. url: https://www.a2d2.audi.
6. Erik Ward and John Folkesson, "Vehicle localization with low cost radar sensors". In: (2016), pp. 864–870. doi: 10.1109/IVS.2016.7535489.
7. Simon Chadwick, Will Maddern and Paul Newman, "Distant Vehicle Detection Using Radar and Vision". In: (2019), pp. 8311–8317. doi: 10.1109/ICRA.2019.8794312.
8. Bence Major et al. "Vehicle Detection With Automotive Radar Using Deep Learning on Range-Azimuth-Doppler Tensors". In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 2019, pp. 924–932. doi: 10.1109/ICCVW.2019.00121.
9. Yair Wiseman, "Ancillary ultrasonic rangefinder for autonomous vehicles". In: *International Journal of Security and Its Applications* 10.5 (2018), pp. 49–58.
10. Wenyuan Xu et al. "Analyzing and Enhancing the Security of Ultrasonic Sensors for Autonomous Vehicles". In: *IEEE Internet of Things Journal* 5.6 (2018), pp. 5015–5029. doi: 10.1109/JIOT.2018.2867917.
11. Marco Claudio De Simone, Zandra Betzabe Rivera and Domenico Guida, "Obstacle avoidance system for unmanned ground vehicles by using ultrasonic sensors". In: *Machines* 6.2 (2018), p. 18.
12. David Eigen, Christian Puhrsch and Rob Fergus, "Depth map prediction from a single image using a multi-scale deep network". In: *Advances in neural information processing systems*. 2014, pp. 2366–2374.
13. Ian P Howard, *Perceiving in depth, volume 1: Basic mechanisms*. Oxford University Press, 2012.
14. Jin Han Lee et al. "From big to small: Multi-scale local planar guidance for monocular depth estimation". In: *arXiv preprint arXiv:1907.10326* (2019).
15. Huan Fu et al. "Deep ordinal regression network for monocular depth estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2002–2011.
16. Arun CS Kumar, Suchendra M Bhandarkar and Mukta Prasad, "Monocular depth prediction using generative adversarial networks". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. 2018, pp. 413–4138. doi: 10.1109/CVPRW.2018.00068.
17. Yasin Almalioglu et al. "Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 5474–5480.
18. Tuo Feng and Dongbing Gu, "Sganvo: Unsupervised deep visual odometry and depth estimation with stacked generative adversarial networks". In: *IEEE Robotics and Automation Letters* 4.4 (2019), 4431–4437. doi: 10.1109/LRA.2019.2925555.
19. Praful Hambarde et al. "Depth Estimation From Single Image And Semantic Prior". In: *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2020, pp. 1441–1445.
20. Rui Wang, Stephen M Pizer and Jan-Michael Frahm, "Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth". In: *Proceedings of the IEEE*

*Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5555–5564.
21. Arun CS Kumar, Suchendra M Bhandarkar and Mukta Prasad, "Depthnet: A recurrent neural network architecture for monocular depth prediction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 283–291.
22. John Paul Tan Yusiong and Prospero Clara Naval Jr, "DFRNets: Unsupervised Monocular Depth Estimation Using a Siamese Architecture for Disparity Refinement". In: *Pertanika Journal of Science & Technology* 28.1 (2020).
23. Cesar Cadena, Yasir Latif and Ian D. Reid, "Measuring the performance of single image depth estimation methods". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, pp. 4150–4157.
24. Guandao Yang et al. "PointFlow: 3D Point Cloud Generation With Continuous Normalizing Flows". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 4540–4549.
25. H. Fan, H. Su and L. Guibas, "A Point Set Generation Network for 3D Object Reconstruction from a Single Image". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2463–2471.
26. Zhuotun Zhu et al. "Deep learning representation using autoencoder for 3D shape retrieval". In: *Neurocomputing* 204 (2016), pp. 41–50.
27. Liisa Holm and Chris Sander, "Dali: a network tool for protein structure comparison". In: *Trends in biochemical sciences* 20.11 (1995), pp. 478–480.
28. Carsten Berndt, Jens-Dirk Schwenn and Christopher Horst Lillig, "The specificity of thioredoxins and glutaredoxins is determined by electrostatic and geometric complementarity". In: *Chemical Science* 6.12 (2015), pp. 7049–7058.
29. Manuela Gellert et al. "Substrate specificity of thioredoxins and glutaredoxins – towards a functional classification". In: *Heliyon* 5.12 (2019), e02943. issn: 2405-8440. doi: https://doi.org/10.1016/j.heliyon.2019.e02943.
30. Angel X. Chang et al. "ShapeNet: An Information-Rich 3D Model Repository". In: arXiv:1512.03012 [cs.GR] (2015).
31. David Lopez-Paz and Maxime Oquab, "Revisiting classifier two-sample tests". In: *International Conference on Learning Representations*. 2017.
32. Qiantong Xu et al. "An empirical study on evaluation metrics of generative adversarial networks". In: *arXiv preprint arXiv:1806.07755* (2018).
33. Facundo Mémoli, "Gromov–Wasserstein distances and the metric approach to object matching". In: *Foundations of computational mathematics* 11.4 (2011), pp. 417–487.
34. Michael Meyer and Georg Kuschk, "Automotive radar dataset for deep learning based 3d object detection". In: *2019 16th European Radar Conference (EuRAD)*. IEEE. 2019, pp. 129–132.

## Bionotes

**Felix Berens**
Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany
Institute for Artificial Intelligence, Ravensburg-Weingarten University of Applied Sciences, Weingarten, Germany
**felix.berens@rwu.de**

M. Sc. Felix Berens is doctoral candidate in the field of sensor fusion for autonomous driving at the Institue for Automation and Applied Computer Science at the Karlsruhe Institute of Technology, and the Institute for Artificial Intelligence at the Ravensburg-Weingarten University of Applied Sciences. Research Interests: Sensor fusion, sensor placement, machine learning, object detection.

**Stefan Elser**
Institute for Artificial Intelligence, Ravensburg-Weingarten University of Applied Sciences, Weingarten, Germany
**stefan.elser@rwu.de**

Prof. Dr. rer. nat. Stefan Elser works as professor for autonomous driving at the Ravensburg-Weingarten University of Applied Sciences. Research Interests: Machine learning, object detection, sensor fusion and their applications in autonomous driving.

**Markus Reischl**
Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany
**markus.reischl@kit.edu**

apl. Prof. Dr.-Ing. Markus Reischl is head of the research group "Machine Learning for High-Through put and Mechatronics" of the Institute for Automation and Applied Computer Science at the Karlsruhe Institute of Technology. Research Interests: Man-machine interfaces, image processing, machine learning, data analytics.



**Methods**

Henrietta Lengyel*, Viktor Remeli and Zsolt Szalay

# A collection of easily deployable adversarial traffic sign stickers

## Eine Sammlung leicht nutzbarer adversarischer Verkehrszeichenaufkleber

https://doi.org/10.1515/auto-2020-0115
Received July 9, 2020; accepted March 24, 2021

**Abstract:** The emergence of new autonomous driving systems and functions – in particular, systems that base their decisions on the output of machine learning subsystems responsible for environment perception – brings a significant change in the risks to the safety and security of transportation. These kinds of Advanced Driver Assistance Systems are vulnerable to new types of malicious attacks, and their properties are often not well understood. This paper demonstrates the theoretical and practical possibility of deliberate physical adversarial attacks against deep learning perception systems in general, with a focus on safety-critical driver assistance applications such as traffic sign classification in particular. Our newly developed traffic sign stickers are different from other similar methods insofar that they require no special knowledge or precision in their creation and deployment, thus they present a realistic and severe threat to traffic safety and security. In this paper we preemptively point out the dangers and easily exploitable weaknesses that current and future systems are bound to face.

**Keywords:** adversarial attacks, neural networks, deep learning, image classification

**Zusammenfassung:** Das Aufkommen neuer autonomer Fahrsysteme und Funktionen – insbesondere von Systemen mit einer Umgebungserkennung auf Basis Maschinellen Lernens – bringt signifikante Veränderungen der Sicherheitsrisiken im Verkehr mit sich. Derartige Fahrerassistenzsysteme sind anfällig für neue Formen böswilliger Angriffe und die Eigenschaften dieser Systeme sind oft noch nicht ausreichend untersucht. Dieser Beitrag zeigt die theoretische und praktische Möglichkeit gezielter physikalischer Angriffe gegen Deep-Learning basierte Erkennungssysteme im Allgemeinen, mit einem Fokus auf sicherheitskritische Anwendungen der Fahrerassistenz wie der Verkehrszeichen-Klassifikation im Besonderen. Unsere neu entwickelten Verkehrszeichenaufkleber unterscheiden sich von anderen ähnlichen Methoden insofern, als dass sie keine besonderen Kenntnisse oder Präzision bei der Erstellung und ihrem Einsatz erfordern. Mit diesen Aufklebern demonstrieren wir eine realistische und ernsthafte Bedrohung für die Verkehrssicherheit. Präventiv weisen wir mit diesem Beitrag auf Gefahren und leicht ausnutzbare Schwachstellen hin, die aktuell und zukünftig zu erwarten sind.

**Schlagwörter:** adversarische Angriffe, neuronale Netze, tiefes Lernen, Bildklassifizierung

## 1 Introduction

Advanced Driver Assistance Systems (ADAS) have successfully found their way into our everyday lives because they make driving more comfortable, and also because they are known to increase transportation safety in general [2]. But it also stands to reason that driver assistance systems can easily become compromised if the environment perception they depend upon is unreliable. Both the transportation industry and academic research have seen a surge in interest regarding various environment sensors like cameras, LiDAR, radar, IR imaging, etc., as well as their combined application for robust detection such as in Li et al. [11]. Difficult visibility conditions such as fog can also greatly influence both human and automated perception of the traffic situation [6]. With the introduction of machine learning – and in particular, deep learning – based perception systems, a new class of unpredictable errors has emerged,

***Corresponding author: Henrietta Lengyel,** Department of Automotive Technologies at the Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, Budapest, Hungary, e-mail: lengyel.henrietta@kjk.bme.hu, ORCID: https://orcid.org/0000-0002-6440-7374
**Viktor Remeli, Zsolt Szalay,** Department of Automotive Technologies at the Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, Budapest, Hungary, e-mails: remeli.viktor@kjk.bme.hu, szalay.zsolt@kjk.bme.hu