DOI: 10.1111/cgf.14613

COMPUTER GRAPHICS *forum*
Volume 41 (2022), number 6 pp. 453–466

Learning Human Viewpoint Preferences from Sparsely Annotated Models

S. Hartwig,[1] M. Schelling,[1] C. v. Onzenoodt,[1] P.-P. Vázquez,[2] P. Hermosilla[1] and T. Ropinski[1]

[1]Ulm University, Ulm, Germany
michael-1.schelling@uni-ulm.de, christian.van-onzenoodt@uni-ulm.de, pedro-1.hermosilla-casajus@uni-ulm.de, timo.ropinski@uni-ulm.de
[2]Universitat Politècnica de Catalunya, Barcelona, Spain
pere.pau.vazquez@upc.edu

Abstract

View quality measures compute scores for given views and are used to determine an optimal view in viewpoint selection tasks. Unfortunately, despite the wide adoption of these measures, they are rather based on computational quantities, such as entropy, than human preferences. To instead tailor viewpoint measures towards humans, view quality measures need to be able to capture human viewpoint preferences. Therefore, we introduce a large-scale crowdsourced data set, which contains 58k *annotated viewpoints for 3220 ModelNet40 models. Based on this data, we derive a neural view quality measure abiding to human preferences. We further demonstrate that this view quality measure not only generalizes to models unseen during training, but also to unseen model categories. We are thus able to predict view qualities for single images, and directly predict human preferred viewpoints for 3D models by exploiting point-based learning technology, without requiring to generate intermediate images or sampling the view sphere. We will detail our data collection procedure, describe the data analysis and model training and will evaluate the predictive quality of our trained viewpoint measure on unseen models and categories. To our knowledge, this is the first deep learning approach to predict a view quality measure solely based on human preferences.*

Keywords: user studies, interaction, perceptually-based rendering, rendering

CCS Concepts: • Computing methodologies → Neural networks; Rasterization; • Human-centred computing → Empirical studies in visualization

1. Introduction

Viewpoint selection is the task to automatically determine an optimal viewpoint for a given 3D model. To support this task, several view quality measures have been proposed [PB96, VBP*09, SS02, VFSH01]. While these measures are based on engineered features, such as entropy and occupancy, they do not consider human preferences. As these preferences are not solely based on a model's geometry, but also its category and probably commonly used depictions, they are hard to capture with conventional view quality measures. Secord *et al.* [SLF*11] proposed to learn human preferences from data instead. However, they still rely on different handcrafted quality measures, which are then used in a linear combination to compute a goodness score for each image. Therefore, within this paper, we propose an alternative to feature engineered view quality measures, by introducing a fully learned view quality measure based on human viewpoint preferences.

To consider human viewpoint preferences during viewpoint selection, we introduce a crowdsourced human viewpoint preference data set, that annotates $58k$ views of ModelNet40 models [WSK*15] chosen from 28 categories. To make such a large scale data collection feasible, we are naturally only able to collect annotations for a subset of all possible viewpoints. Thus, in order to leverage this sparsely annotated data, we exploit standard CNNs to reconstruct dense view spheres, which encode human viewpoint preferences. The thus reconstructed human viewpoint preferences are then used to train two different models for fast inference of this measure. The first model uses CNNs to predict the view quality measure of a single image directly, while the second uses a point cloud-based

sebastian.hartwig@uni-ulm.de

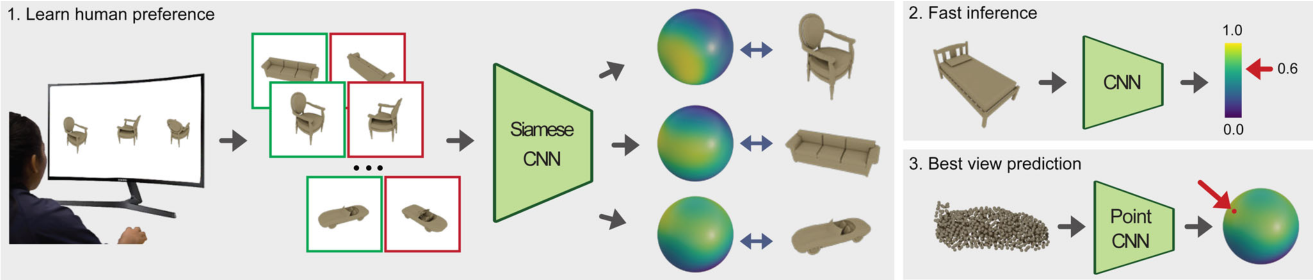


Figure 1: *Within this paper, we present a neural view quality measure learned directly from data. (1) We collected a sparsely annotated data set of human viewpoint preferences for 3220 different models, which enabled us to learn our view quality measure using a Siamese architecture. This large-scale data set allowed our measure to generalize to unseen model categories. Moreover, we demonstrate two methods for fast inference of such measure: (2) we use convolutional neural networks to predict the view quality measure of a single image and (3) we use point convolutional neural networks to predict the best view of a model from 3D data directly.*

architecture [SHVR21] to directly predict human preferred viewpoints based on a 3D model. The process of data collection and learning is illustrated in Figure 1. By using a cross validation study, we demonstrate that our thus learned human view quality measure does not only generalize to 3D models not seen during training, but also to unseen model categories. Thus, the contributions made in this paper are fourfold:

- We provide a large scale human viewpoint preference data set, which contains annotations for 3220 3D models, available at https://github.com/kopetri/human_viewpoint_metric.
- We propose a neural view quality measure learned from user annotated data, that mimics human preferences.
- We demonstrate that the proposed view quality measure generalizes to model categories unseen during training.
- We enable fast prediction of such view quality measures from a single image, as well as of human preferred viewpoints from 3D models directly by using neural networks.

Within the remainder of this paper, we will first discuss the work related to our approach in Section 2, before providing details on crowdsourcing human viewpoint preferences in Section 3. We then describe our view quality measure in Section 4, followed by Sections 5–7 where we describe how we learn such measure using neural networks. Finally, we address limitations of our approach in Section 8 and conclude in Section 9.

2. Related Work

Many aspects can be of importance, when it comes to the definition of the best view for a 3D model. One could argue for aesthetics, information content, recognizability, or culture-bound reasoning. Thus, view quality measures ideally capture these aspects, to measure the quality of a selected view of an object. Early research focused on analysing primitives, e.g. number of visible triangles, projected area or number of degenerated faces in orthogonal projections [PB96, BDP00]. Later, more sophisticated measures were introduced, such as viewpoint entropy [VFSH01], viewpoint Kullback–Leibler distance [NSGP*05], viewpoint mutual information [FSG09] or mesh saliency [LVJ05], to name only a few. In the following, we briefly outline this field, by first discussing conventional view quality measures, before reviewing those which are learned, and those which incorporate human preferences.

Viewpoint selection. Many conventional view quality measures try to capture the properties, which a view should have, in order to make it relevant. Researchers have referred to the optimal properties of a view as its quality, goodness or noteworthiness. The features that were evaluated have increased in complexity over time. Among the dozens of papers one can find, some of them deal with geometric features, such as counts of visible polygons or area from a certain view [PB96, BDP00]. Other papers use information-based measures based on geometric features, such as entropy, mutual information or Kullback–Leibler divergence etc. [VFSH01, FSG09, VFSG06, NSGP*05, TMWS12]. Most of the techniques use brute-force approaches, which require evaluating several hundreds of views. This is costly, since it takes several seconds to some minutes, depending on the complexity of the object, even for modern computers. The interested reader can refer to the survey by Bonaventura *et al.* [BFS*18] where they describe and analyse the most common measures applied to polygonal models.

Learning-based methods. Recently, some techniques that make advantage of learning algorithms have been developed. Some techniques use such algorithms to improve over previous measures [SLF*11], or to assign scores to candidate views or photographs [KTL*17, YLLY19, ZFY20]. In the context of point cloud recognition tasks, viewpoint feature histograms have been proposed, which are used as feature descriptors [RBTH10, RWL*21], which can be used for point cloud registration [AMB17]. In the work of Fang *et al.* [FZS*20], a canonical viewpoint is predicted by a network for point cloud classification. The ambiguity of best viewpoint selection was addressed through a dynamic label generation strategy by Schelling *et al.* [SHVR21], to directly predict high quality viewpoints, which the authors demonstrated for four view quality measures.

Learning on image pairs. Ranking stimuli using paired comparisons is a long-standing technique [Thu27, KS40, Gut46, Ken48, Mos51, BT52]. In 1927, Thurstone [Thu27] introduced the law of comparative judgement. This seminal work introduced a statistical model to determine the user preference with respect to a given set of stimuli. Another commonly used statistical model for ranking from paired comparisons is the one proposed by Bradley and Terry [BT52]. In the last decade, several works have used these ideas in combination with recent advances in machine learning. Liang *et al*. [LG14] used an active learning approach to learn the ranking of a set of images with respect to a human-annotated attribute. However, the images were described by handcrafted features and, as a ranking function, a simple linear model was used. Recently, Zhang *et al*. [ZIE*18] use a neural network to assess the perceptual similarity between images. Their method first computed the difference between features of a pre-trained network for a given image patch as input and a corrupted version of the same patch. These differences were used to learn to rank two corrupted patches, with respect to the original image, based on human annotations. However, none of these works used a deep neural network to learn to rank the preferred point of view from human-annotated images.

Evaluating human preferences. Although, the selection of viewpoints has been studied for some time, in areas such as object recognition, there are not so many studies that have evaluated the preferences of subjects with regard to a certain viewpoint of a synthetic object. Blanz *et al*. [BTB99] evaluated the views users preferred to represent an object (*the best possible impression of the objects shown in a screen*) in two experiments. They gathered the opinions of 36 and 18 users for sets of 14 and 18 models, respectively. Later, Tarr and Kriegman evaluated the perceived similarity by presenting subjects different views for a torus and a bell [TK01]. Jagadeesan *et al*. [JLC*09] analysed the view preferences of 80 users using Amazon Mechanical Turk. In this case, they used only CAD models, and the experiment was run using a small set of prefixed views of five models. Another small set of users' preferences was gathered by Dutagaci *et al*. [DCG10], where they analysed 68 models with 26 subjects. Our goal is to analyse a much wider range of models with a larger number of users, in order to learn a human viewpoint preference predictor, and to investigate how this generalizes beyond model categories. To the authors' knowledge, the most extensive user study to date is the one by Secord *et al*. [SLF*11]. They used a high number of users (524) that choose among pairs of views of 16 different, rather complex models. Their goal was to learn a set of weights, that can be applied to previously developed measures in order to compute the view goodness function of the input images and use the Bradley–Terry model [BT52] to simulate human preferences on viewpoint selection. On the contrary, in this work, we aim to learn human view preferences without relying on predefined metrics or any statistical model.

3. Sparsely Annotated Data Set

Collecting human view preferences poses several challenges. Simply asking for the subjective 'best view' of a model might be sufficient to indicate appealing view directions, but it does not provide insights on how a user would compare two views to each other. Furthermore, selecting the 'best view' is a challenging task for an average user as it involves rotating a 3D model, a task with which some participants might not be familiar with. Consequently, these annotations should be handled with care, and thorough quality assessments are required. In order to capture the relations between different viewpoints, we are interested instead in how humans perceive a view direction in relation to another one.

Asking a user to quantify the quality of a view direction with respect to another is subject to high levels of ambiguity. Different users might perceive values differently, i.e. the same numerical value might have different meanings for different users. With these considerations in mind and inspired by Secord *et al*. [SLF*11], we design our view quality measure from data collected in a discrete forced choice experiment. Thus, we only ask users to decide between 'better' or 'worse', effectively only assessing the order between the presented views, rather than their absolute quality.

Using such setup, we collect a large human-annotated data set that contains sparse annotations for 3220 3D models. We favour a large number of annotated models instead of highly dense annotations for each of them, since neural networks have been shown to generalize well to unseen data, and can thus fill in the missing annotations. In the rest of this section, we describe how we generated this data set.

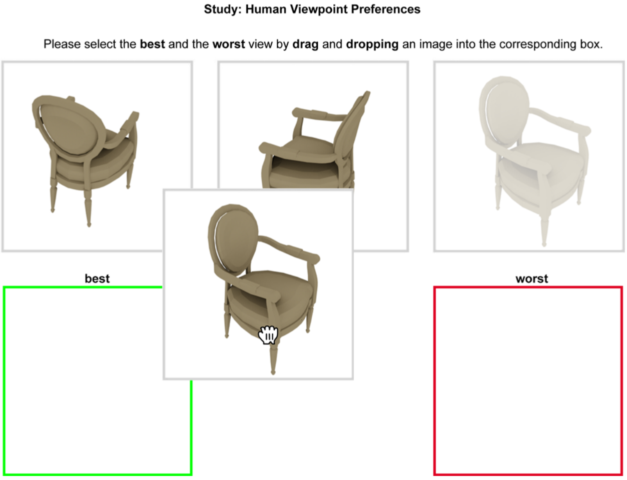
3.1. Online study

In order to collect annotations for our viewpoint data set, we implemented a web application, enabling us to crowdsource the viewpoint preferences via Amazon Mechanical Turk. As mentioned before, to keep the workload low, maintain participants motivation, and therefore, achieve higher quality, we formulate our study as a discrete forced choice experiment. This design choice is also in line with a similar study by Secord *et al*. [SLF*11], which conducted a two-alternative forced choice experiment. Presenting triplets to participants, instead of tuples, increases the number of annotated tuples per interaction of the participant from 1 to 3. Also, exposing participants to only three stimuli does not compromise the observer's channel capacity [Mil56, RA11] enabling stable measurements.

We formulate the annotation task as follows: Given a triplet of views, the participants have to select the views they consider to be the best and worst, respectively, amongst the presented three views. In order to make a selection for a view, the participants have to drag and drop the view into a corresponding box labelled 'best' and 'worst'. The web interface can be seen in Figure 2.

As view preference is dependent on the context in which the images are presented, the instructions given to the participants can have an influence on their choices. To ensure that different participants have the same understanding of the task, we formulate the instructions stated below, following the examples given by previous user studies [BTB99, SLF*11].

> You will be presented **three** images. Your task is to select the views, which are in **your opinion** the **best** and the **worst** views of the presented set. There is no right or wrong.
>
> In some cases, multiple views might be equally good or bad. In this case, try to enforce a decision.
>
> In this experiment, **'best view'** corresponds to the **most familiar view** of an object. Consider that you will show only one view of the objects to another person, and the person should be able to recognize the object as quickly as possible by looking at that view.



Study: Human Viewpoint Preferences

Please select the **best** and the **worst** view by **drag** and **dropping** an image into the corresponding box.

Figure 2: *In our online study, the participant has to select the best and worst view amongst three given views. The web interface provides a drag and drop method to make the selection. A selection can be changed by drag and dropping another view on top of an already selected choice. Each participant has to annotate 50 triplets.*

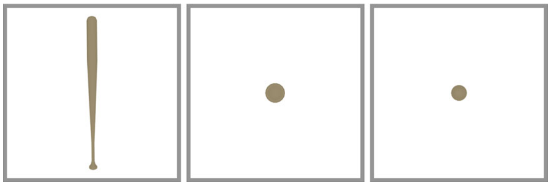


Figure 3: *This attention check is used to detect if a participant did not understand the task or is clicking through the study. The first view is what we consider the best view of a* baseball bat.

Each participant had to label 50 triplets, which took 8 min on average. We also included attention checks to detect participants that did not understand the task or intentionally select bad views, by adding view pair examples showing a definite best and worst view enabling us to filter out bad responses, see Figure 3 for such a check. Data from participants who did not pass this test were not considered in our data set. Additionally, we ensured that each triplet of views is labelled twice by two different participants, in order to get robust annotations. In total, 950 participants finished the described annotation task.

3.2. 3D models

For our data set, we used ModelNet40 [WSK*15], which contains a large set of models for man-made objects spanning several categories. In total, ModelNet40 features 40 model categories, from which we remove 12 categories, which leaves us with 28 categories. We kept categories with more than 115 models (93 training, five validation, 17 testing), and removed categories (*bowl, cup, curtain, person, stool, tent*) with an insufficient number of models. We further excluded the categories (*xbox, cone, glass box, mantel, range hood, stairs*), which exhibit similar shapes in all models. For instance, the shapes in the categories *xbox* and *cone* are virtually identical. For each category, we use 115 models summing up to 3220 models. Lastly, we generate 18 views (six triplets) for each of these 3220 models, which yields 57,960 images in total.

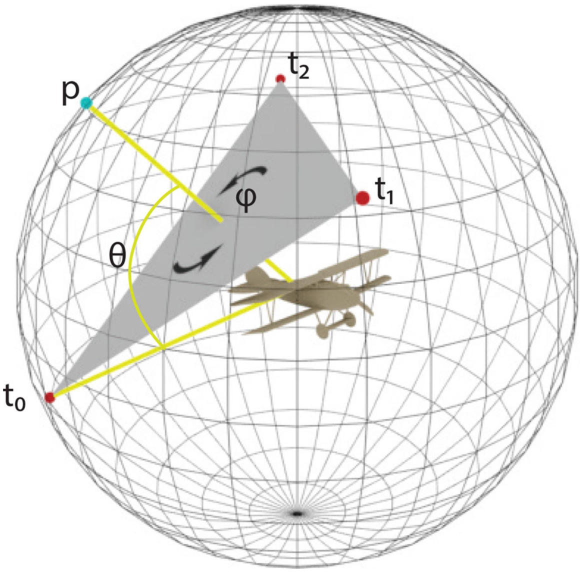


Figure 4: *Our employed view sampling strategy, where red dots represent sampled views from a unit sphere around a model. We first sample a random sphere point* p *displayed in green. Then we generate an equilateral triangle centred around* p *and use triangle points as viewpoints.*

3.3. View generation

To be able to collect viewpoint annotations for the given models, we sample random positions on a unit sphere around each model. Since we are using triplets of viewpoints for the annotation task, and to maximize the difference of information content between views, we sample three camera positions at once using corners of an equilateral triangle (see Figure 4). We obtain these camera positions by first sampling a random point $p \in S^2$ on a unit sphere. Second, we randomize the angle θ between p and the triangle points t_i, $i \in [0, 1, 2]$. Third, we randomly rotate t_i around p by ϕ. Finally, we select each triangle point t_i as eye vector for our cameras. For each camera, we set the target vector to the centre of the model, and the up vector to the positive y-axis. We repeat this process six times for each model, to obtain six triplets, resulting in 18 views per model. Note, that our sparse view sampling generates 13 times fewer views per model compared to Secord *et al.* [SLF*11]. For rendering, we use perspective projection. We place a light located at the camera's position facing the object, and we used ambient occlusion in order to increase visible details.

3.4. Validation

To validate the crowdsourced annotations, we examined the best and worst view agreement between participants. To do so, we count how

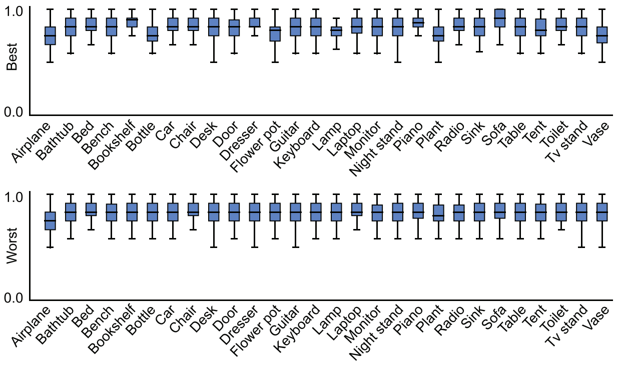


Figure 5: *Agreement of human labels for our selected categories of ModelNet40 for best viewpoints (*top*) and worst viewpoints (*bottom*). As can be seen, participants agreed on both best view and worst view in most of the cases for all categories.*

often a view was labelled as best or worst, and normalize by dividing by the number of participants per view. In Figure 5, we display the thus obtained averaged agreement rate per model category for best view selection (*top*) and worst view selection (*bottom*). As can be seen, participants show high agreement on both best view and worst view in most of the cases for all categories. While categories with model shapes that are rotational invariant, e.g. *bottle*, *flower pot*, *vase*, have a best view agreement rate below 80%, categories of models with clear front and back sides show agreement rates for best views close to 90%, e.g. *bookshelf*, *dresser*, *piano*, *sofa*. Looking at the bottom row in Figure 5, the agreement rate for all categories is above 80%, except for *airplane*, indicating a slight dispute regarding airplanes seen from the bottom.

Moreover, we evaluated if the collected data were consistent among models of the same category. To evaluate this, we selected all view directions that were selected as the best in a triplet for all models in a given category, and use a kernel density estimation (KDE) to approximate this distribution. In order to compute KDE in a spherical domain, we replace the traditional Gaussian function used in KDE with the Von Mises–Fisher distribution [Fis53]. Figure 6 illustrates the resulting density maps on the sphere for each category. We can see that users were consistent with the annotations among different models of the same category, since each category has distributions with localized areas with high density. For example, users prefer to view objects from the category *bookshelf* from the front while *cars* are preferred from a left or right viewpoint.

4. Modelling Human Viewpoint Preferences

The collected data set is composed of pairs of images annotated by humans, which indicates their preference of one image over the other. With these binary classifications for all views of a given model, it is possible to reconstruct a quantitative measure of the view preference distribution and, therefore, possible to capture the global ranking of all views. In the following, we will give a brief mathematical description of the reconstruction from binary classifications.

Let $f : D \to \mathbb{R}$ be a function, defined on a finite set D. In the context of this work, $D \subset S^2$ are the sampled viewpoints on the view sphere, and f is the (unknown) distribution of human view preference for a given model. For each point $v \in D$, we define the *count* of v as

$$count(v) := |\{x \in D : f(x) < f(v)\}|, \tag{1}$$

where $f(x) < f(v)$ indicates that the user prefers view v over x. We define the reconstruction f^* of f as:

$$f^* : D \to \mathbb{R}$$
$$f^*(v) := \frac{count(v)}{|D \backslash \{v\}|}. \tag{2}$$

This reconstruction can be understood as a normalized counter of how many times a view direction was selected over other view directions in the set. Thus, this reconstruction serves as a quantitative view quality measure, that can be used to compare different views of a given 3D model:

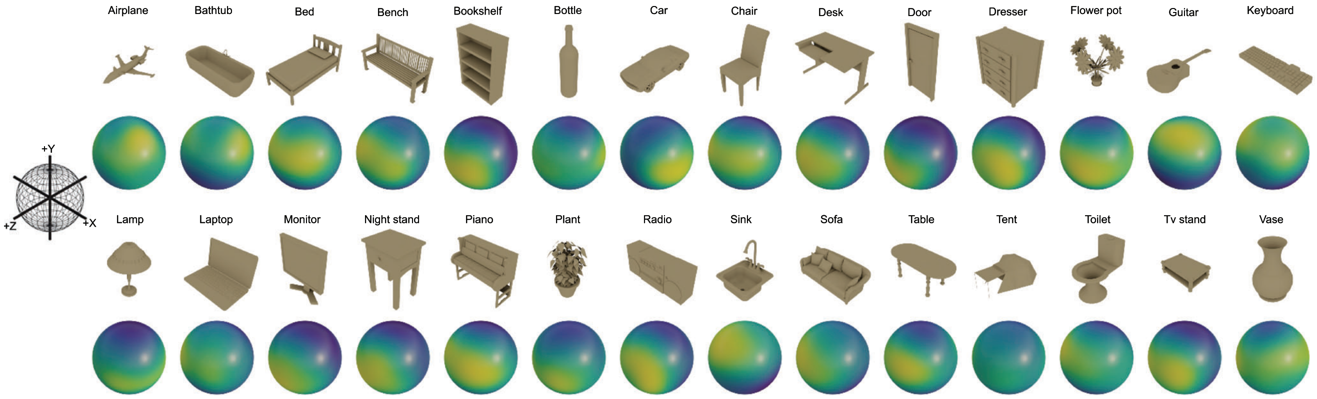


Figure 6: *Collected per-category view preference. For each category, we select all view directions, which were labelled best in a triplet and use a kernel density estimation on the sphere to approximate this distribution. Yellow regions indicate view directions highly favoured by humans. The resulting distributions indicate that, on the data collected, there is consistency among objects from the same category.*

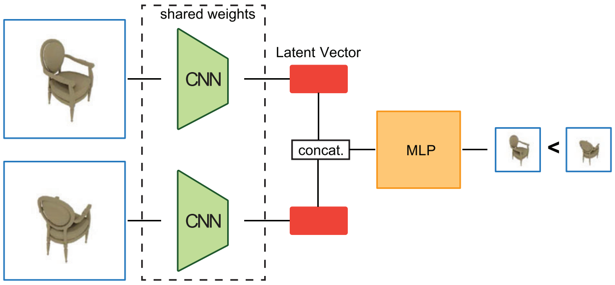


Figure 7: *The architecture of our viewpoint selector consists of a feature encoder and decoder for classification. The input to the network are two images, which are fed successively into the encoder, resulting in two latent vectors of size 128. The decoder concatenates both latent vectors before further processing and outputting a binary prediction.*

Proposition 1. *The reconstruction* f^* *preserves the order induced by* f *on the points in* D, *i.e.*

$$f(v_1) < f(v_2) \Leftrightarrow f^*(v_1) < f^*(v_2), \qquad \forall v_1, v_2 \in D. \quad (3)$$

Proof to this proposition and further insights are provided in Appendix A.

5. View Quality Measure Learning

Computing our measure will require evaluating Equation (1) for all elements in the set D. However, asking a user to annotate all possible pairs of view directions for a given model is not practical. In this paper, we propose to learn Equation (1) from sparse data instead.

5.1. Architecture

Our model follows a Siamese architecture similar to the one proposed in the one-shot classification framework of Koch *et al.* [KZS15]. We consider two images as input, which are generated from the two view directions we aim to compare, v_1 and v_2. These images are then processed by two image encoders, which have shared parameters, resulting in two latent vectors describing a compressed representation of the two input images. The image encoder is a CNN composed of four feature transformation blocks with increasing feature size: 16, 32, 64, 128 and a down sampling factor of 2. Each block is a stack of three layers: convolution, batch normalization and ReLU activation. After the last layer, we append a global average pooling layer, which outputs a latent vector for each image. For our experiments, we use an image resolution of 256 × 256$, and a latent vector with size 128 (see Table B.3 in the appendix for a comparison of different sizes). The two latent representations are then concatenated and processed by a multi-layer perceptron (MLP) with three hidden layers with 256, 128, 32 features each. The last layer applies a sigmoid activation function to output the probability of $f(v_1) < f(v_2)$. Based on these probabilities, we can then estimate f^*. Figure 7 provides an illustration of the used architecture. Note that if our model is trained with random view directions, we can estimate f^* for a 3D shape by sampling directions in the sphere, D, at any resolution.

Previous work has suggested statistical models to predict the probability of $f(v_1) < f(v_2)$ from the goodness of each input [BT52]. This model was used by Secord *et al.* [SLF*11] to learn the goodness function with a linear model. The Bradley–Terry model [BT52] models the variance in the annotations as variance in the goodness score, and solve it with maximum likelihood estimation (MLE). We favour instead a model that directly predicts this probability with a deep neural network and models the variance directly from data using binary cross-entropy, an optimization comparable to MLE. Further, neural networks are able to generalize by exploiting information present in the input images, which probabilistic models do not account for.

5.2. Training

We train our viewpoint selector using the Adam optimizer [KB14] for 200 epochs and a batch size of 16, using a learning rate of 0.001, reducing it by a factor of 0.1 if the validation loss did not improve for 5 epochs. We use binary cross-entropy as our loss function. We regularize our model by using a dropout probability of 0.5 [HSK*12] in the last MLP, and further add Gaussian noise to the ground truth labels with a standard deviation of 0.2, preventing label from flipping and clamping the label to be in range [0,1].

5.3. Evaluation

In this section, we describe the experiments conducted to evaluate our Siamese network.

5.3.1. *Learning the view quality measure*

In this experiment, we evaluated the performance of our network on predicting the human view preference for a given pair of images.

Data set. We use the data described in Section 3 to train the model that mimics human view preferences. First, we select only those triplets in which two users agreed on their annotation. Then, we generate three pairs of annotated images from each triplet for training. Lastly, we divide the resulting images in three different sets: training, validation and testing. We make sure that in each split, we have models from all categories, but images from one model are present in only one of the three sets. The training set is composed of 60,462 image pairs, the validation set is composed of 3576 pairs and the test set contains 9978 image pairs. We call this data set FILTERED. To further augment the data during training, we apply random rotations in the range of [−90, 90] degrees. The network learns to be invariant to rotations around the camera's eye vector, which is only possible since we fixed the camera's up vector. In a scenario with arbitrary camera up vector, this data augmentation strategy would be problematic to rank viewpoints. However, disabling random rotations results in worse performance, see Section 5.3.1. Since our model is not equivariant to the order of the input images in each pair, we also randomly invert the order of the images in a pair and its corresponding label during training.

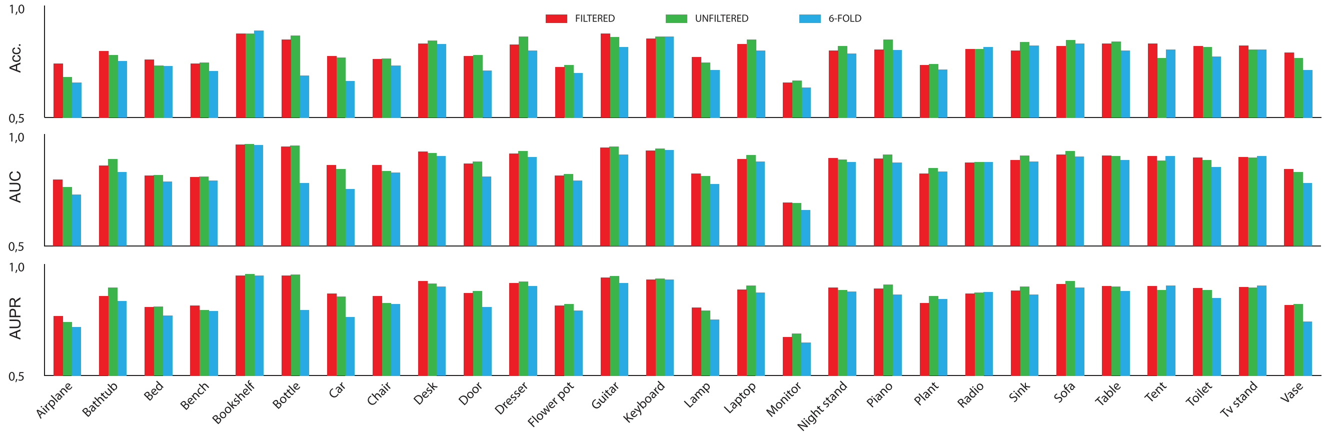


Figure 8: *Performance results of our Siamese model to predict the binary classification for the image pairs in our test set. We evaluate three different training methodologies:* Filtered*, where we train with image pairs in which users agreed in the annotation;* Unfiltered*, where we train with all training data from our data set and* 6-fold*, where the model did not see the respective categories during training.*

Table 1: *Performance results of our best network, Secord* et al. *[SLF*11] and several common view quality measures, which are not based on human preferences. Performance is averaged over all predictions.*

	Accuracy	AUC	AUPR
Ours Unfiltered	**79.9%**	**0.796**	**0.847**
Secord* [SLF*11]	76.7%	0.767	0.825
Secord [SLF*11]	73.4%	0.734	0.800
Above [GRMS01]	69.7%	0.697	0.773
Viewpoint Entropy [VFSH01]	68.4%	0.684	0.763
Surface Visibility [PB96]	63.1%	0.631	0.723
Kullback-Leibler [NSGP*05]	62.1%	0.621	0.716
Silhouette Length [HS97]	59.2%	0.592	0.694
Projected Area [PB96]	52.1%	0.521	0.641
Max Depth [SS02]	50.7%	0.507	0.630
Mutual Information [FSG09]	43.8%	0.438	0.578

*Using optimized weights on unfiltered.
For each metric, we highlight the method performing best.

Results. To measure the performance of the model, we compute three measures: binary accuracy (Acc.), area under the receiver operating characteristic curve (AUC) and area under the precision-recall curve (AUPR). We evaluated the metrics on each pair of images in the test set in the two possible permutations. In Figure 8 (red), we report the resulting values for all metrics and categories. Results show that our model can mimic human preference with high accuracy. For categories such as *bottle*, *dresser* or *table*, the model is able to achieve high values in all metrics. On the other hand, other categories are more difficult to predict, such as *airplane* or *monitor*. Table 1 presents the averaged performance over all predictions, where we compare our method against existing visual metrics and the data-driven approach of Secord *et al*. [SLF*11], which combines existing view quality measures in order to approximate human preferences. We use the original weights presented in the paper, denoted as Secord, in Table 1, and the optimized weights in our training data set, denoted as Secord*. Figure 9 illustrates the reconstructed view quality measure for one model of each category from our test set. Models are rendered from the viewpoint with the highest value based on our measure, which is also indicated by a red dot in the view quality distribution. In the supplementary material, we also provide the predicted view quality distribution for all the models in our data set. Looking at symmetric shapes like *car* or *table*, one can see equal predictions for both, left and right sides. Rotation symmetric shapes like *lamp*, *bottle* and *vase* show arbitrary good viewpoints around the equator.

5.3.2. *Ambiguous annotations*

As human annotations are subject to ambiguity, we also investigated how robust our model is to ambiguous image pairs.

Data set. In our previous experiment, we filtered the data for those triplets in which two users agreed with their annotations. In this experiment instead, we generate a training/validation set, which is composed of all collected data without filtering. Note that even if the users did not agree in the annotation of the complete triplet, they can still agree on some resulting pairs generated from the triplet. The resulting data set, Unfiltered, is composed of 92,364 pairs of images for training, 5700 image pairs for validation, and, as before, 9978 image pairs for testing.

Results. We train the same model as in the previous experiment on this new data set. Figure 8 (*green*) illustrates the results for all metrics and each category. We can see that there is only a small change in performance for most of the categories. Moreover, Table 2 also presents the averaged performance over all predictions, where the model achieves an accuracy of 79.9%, an AUC of 0.796 and a AUPR of 0.847. When we disable random rotations during training, the accuracy drops to 78.9%. These results indicate that our model is robust to ambiguities on the human annotations and also benefits from the additional training data.

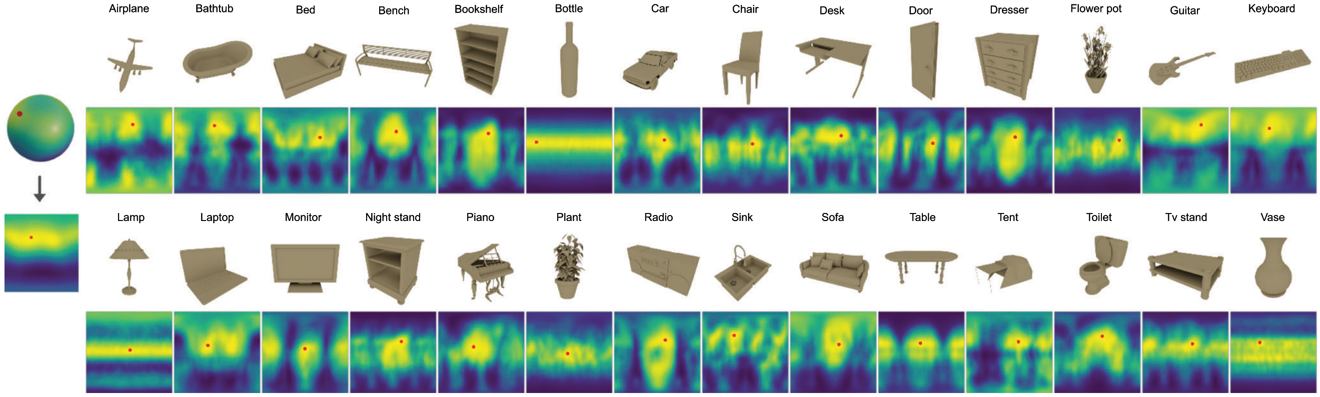


Figure 9: *Visualization of the learned human view quality measure for one model per category. In the upper row, we show the best view for each model sampled from our view quality measure. In the row below, we show the corresponding learned viewpoint distribution, which was reconstructed from our viewpoint selector. The red dot indicates the best viewpoint.*

Table 2: *Performance comparison of our Siamese model for the three data sets.*

	Accuracy	AUC	AUPR
OURS UNFILTERED	79.9%	0.796	0.847
OURS FILTERED	79.7%	0.800	0.850
OURS 6-FOLD	76.2%	0.765	0.824

Table 3: *For our 6-fold cross validation experiment, we randomly split 28 categories into subsets of five categories. Note that for Subset 6, we randomly chose two additional categories to add up to five categories. For each subset, we train a network with the absence of images of the respective categories. During test time for each classifier, we measure performance for each individual category, which has been left out during training.*

Subset 1	Subset 2	Subset 3	Subset 4	Subset 5	Subset 6
Chair	Bathtub	Airplane	Bottle	Bed	Dresser
Monitor	Bench	Bookshelf	Keyboard	Car	Guitar
Sink	Dresser	Door	Laptop	Desk	Plant
Tent	Night stand	Flower pot	Sofa	Lamp	Sink
TV stand	Radio	Piano	Vase	Toilet	Table

5.3.3. *Generalization to unseen categories*

Figure 6 indicates that there is consistency on the view preference for models of the same category. Therefore, measuring the ability of our model to generalize to unseen categories is of key importance to measure the validity of our measure.

Data set. For this experiment, we use a 6-fold cross validation on the 28 categories to evaluate the model. Table 3 illustrates the category types that are left out during training in each fold. We refer to this data set as 6-FOLD.

Results. We train the same model as in the previous experiments on the new data set. Figure 8 (*blue*) presents the resulting metrics for each of the categories. For most of the categories, training with this new data set translates only into a slight drop in performance. For other categories, such as *bottle* or *vase*, the drop in performance is large but still in a good range. When we look at the averaged performance over all predictions in Table 2, we can also see a small drop in performance for all metrics.

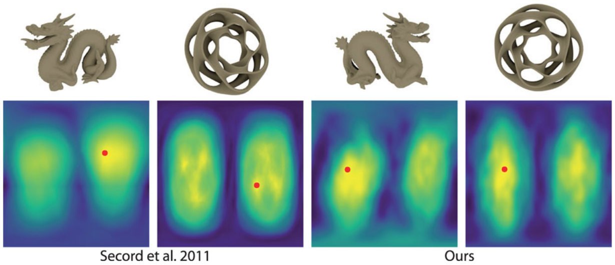


Figure 10: *Generalization of our viewpoint selection network, which has been trained on ModelNet40. On the left, we show the selected best view and corresponding view quality from Secord* et al. *[SLF*11], respectively. On the right, we demonstrate our learned view quality measure. Red dots represent the best viewpoint, which is displayed above. This demonstrates the generalization of our viewpoint selector to novel categories:* dragon *and* heptoroid.

Further, we compute view quality distributions on the sphere for two common models in computer graphics, the Stanford dragon and the heptoroid, using Equation (2) and our Siamese model. In Figure 10, we compare our learned view quality measure against the best model of Secord *et al*. [SLF*11], which approximates human preferences specifically collected for these models. Note that these models are out of the domain for our training data, as they are animals or abstract models, in contrast to the man-made objects of ModelNet40 used for training. The method of Secord *et al*. [SLF*11] selects the right side of the models as best view, whereas our method favours the left side. Despite these differences, our measure generates similar distribution as the ones published by Secord *et al*. [SLF*11].

6. View Quality Measure Inference

Evaluating our quality measure can be computationally expensive since it requires evaluating Equation (1) for all views in a finite set, generating a quadratic cost with respect to the number of sampled views. In order to reduce this cost, we propose to train a convolutional neural network to predict the value of the reconstructed f^*, Equation (2), directly from a single image. With this setup, our measure can be approximated in milliseconds for a single image without comparing it to any other view direction.

6.1. Architecture

For our CNN encoder, we used the same architecture as in our Siamese model, followed by a two-layer MLP with 64 and 32 hidden neurons each. The final prediction is processed by a sigmoid function to transform the prediction into the range [0, 1].

6.2. Training

We train our model using the Adam optimizer [KB14] for 15 epochs, and a batch size of 32, using a learning rate of 0.005, reducing it by factor 0.1 after 10 and 15 epochs. We regularize our model by using weight decay with a factor of 0.001. As a loss function, we use binary cross-entropy loss for this regression problem, since it avoids gradients becoming zero due to the last sigmoid layer.

6.3. Evaluation

In this section, we measure the accuracy of our regression model.

Data set. For this data set, we use the same models from ModelNet40 as used in our user study. We reconstruct the view quality measure for each model as follows. For a given model, we sample 1000 view directions on a Fibonacci sphere around the model, which yields uniformly distributed viewpoints. Next, we generate image pairs for all combinations $C = \binom{1000}{2}$ to select two images from the sampled views. Finally, we reconstruct the view quality measure using Equation (2) and our Siamese model.

We create three different splits of the data: training, validation and test, containing 2.5 M, 168 K and 476 K images each. Note that images from the same model are all contained in the same set. To further augment the training data, we use random rotations of 90° and add Gaussian noise to the images with a standard deviation of 0.002.

Results. We measure the performance of the model with mean squared error (MSE) and coefficient of determination (R^2), for which our regression model achieved a performance of 0.02 and 0.71, respectively. We further analysed the model by plotting the predicted values versus viewpoint quality values in Figure 11. We see that most of the points are close to a perfect regression model, indicated by the diagonal line.

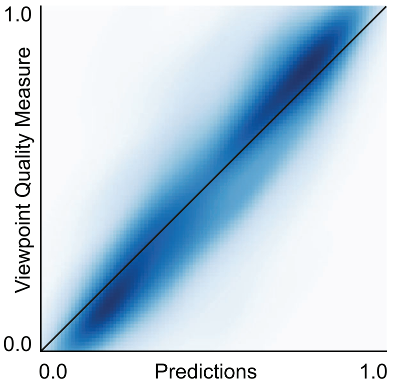


Figure 11: *Viewpoint quality values versus predictions of our regression model. We can see that most of our predictions are close to the viewpoint quality values, close to the diagonal line, indicating a perfect regression model.*

7. Best View Prediction

In many applications, the user is only interested in obtaining the best view from which to inspect a 3D model. This setup will require to evaluate our view quality measure for all views in a finite set, which can be computationally demanding. Recent research has proposed to learn the best view of a 3D model directly from 3D data [SHVR21] according to four established handcrafted view quality measures. This method uses point convolutional neural networks [HRV*18] to analyse the 3D structure of a model, and to predict the best view for the different view quality measures. In this paper, we employ the same architecture as Schelling *et al.* [SHVR21] to predict the best view direction for a 3D model based on our proposed view quality measure instead of the handcrafted quality measures used in the paper. This enables us to perform a prediction within milliseconds.

7.1. Architecture

We use the same architecture as the one proposed by Schelling *et al.* [SHVR21]. The shape encoder uses four point convolution layers [HRV*18] with increasing receptive fields [0.05, 0.2, 0.3, $\sqrt{3}$] and number of features [128,256,1024,2048]. The resulting latent vector is then processed by an MLP with two hidden layers and 1024 and 256 features, in order to predict the best view.

7.2. Training

We train our model to predict the best view for models of all 28 categories at the same time. For the loss, we use the one proposed by Schelling *et al.* [SHVR21], where the ground truth label used in each step is computed based on the prediction of the network. This loss uses two methods to generate the ground truth label, multi labels and Gaussian labels, that are applied at different stages of the training. This procedure avoids inconsistencies during training, due to symmetries in the view quality distribution for similar models. We use an Adam optimizer [KB14] with batch size of 8 and learning rate of 0.001, which is multiplied by 0.75 every 200 epoch. We train for a total of 3000 epochs and switch from multi labels to Gaussian labels after 1500.

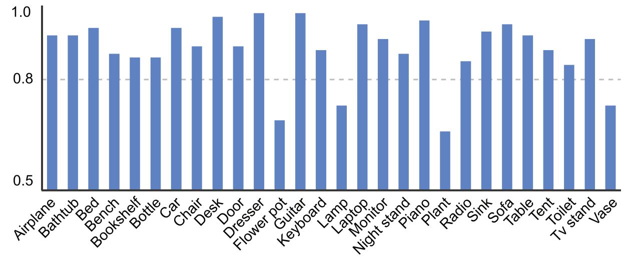


Figure 12: *Normalized view quality of the predicted viewpoints per category of the viewpoint prediction network of Schelling* et al. *[SHVR21] trained on our learned human view quality measure.*

7.3. Evaluation

In this section, we evaluate if our measure can be learned by the best view predictor of Schelling *et al*. [SHVR21] directly from 3D models.

Data set. To train this model, we used the 3D models and view quality measures described in Section 6.3. Moreover, we use 4096 points sampled on the surface of the objects as also done in the PointNet paper [QYSG17], to represent the 3D shape of each object. We divide the data in three data splits: training, validation and testing, where each set contains 2576, 322 and 322 3D models, respectively.

Results. We measure performance with the normalized view quality measure, which ranges from 0 to 1 for each model, as it is normalized based on the minimum and maximum value among all views for a single model. The trained model is able to predict viewpoints with an average normalized viewpoint quality of 0.88. We report per-category results in Figure 12. Looking at individual category accuracy, only four categories (*flower pot*, *lamp*, *plant*, *vase*) have a value lower than 0.8, indicating that the method can learn the best view direction in our view quality measure directly from the 3D shape of a model. Note, that for rotational symmetric models, the best views are in a narrow region close to the equator. That means that a small variation of the latitude in the view prediction translates in a bigger error in our metric than for other shapes, where the best view region has a more uniform shape.

Finally, we provide a qualitative comparison between the best view predicted by Schelling *et al*. [SHVR21] trained on our human viewpoint measure and, four handcrafted measures used by the original paper. In Figure 13, we display viewpoints, which were selected by Schelling *et al.* [SHVR21], conditioned to our human viewpoint measure in the first column, visibility ratio [PB96], viewpoint entropy [VFSH01], viewpoint Kullback–Leibler [NSGP*05] and viewpoint mutual information [FSG09] in the other columns. We can see that the best view predictor trained on our human viewpoint measure provides robust viewpoint selections for different types of models, while handcrafted viewpoint measures select viewpoints with high levels of occlusions, making it difficult to identify the individual objects.

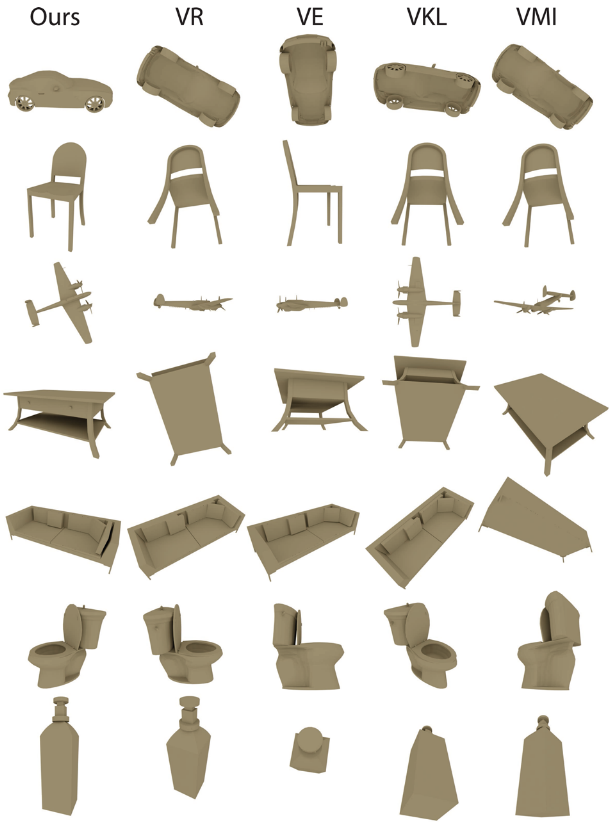


Figure 13: *Visualization of the best viewpoint selected by the network from Schelling* et al. *[SHVR21] for different view quality measures. For columns from left to right, we show the best viewpoint for our human-based measure, visibility ratio, viewpoint entropy, viewpoint Kullback–Leibler and viewpoint mutual information.*

8. Limitations

Our method is not free of limitations. The generalization ability of CNNs enabled us to learn the human view preference from data directly. However, this technology is susceptible to variations of the input data. This makes our models dependent on the rendering algorithm used during training. However, generalization to out-of-distribution rendering algorithms could be introduced with style transfer techniques.

Moreover, our method does not consider the context of such visualizations. Different tasks will select different views as good, e.g. a 3D modelling software will require a good overview of the model while a volume rendered image in the context of medical visualization will favour views where relevant information is shown. Therefore, context-aware view quality measures should be further investigated by collecting adequate data sets. Lastly, in our experiments, we consider the up vector of the camera fixed. Although, recent work has suggested learning the up vector of a model [KTL*17], further experiments are needed to investigate the effect of arbitrary camera rotations on the human preference.

9. Conclusion

Within this paper, we have shown how viewpoint preference of humans can be simulated with modern deep learning techniques. We collected a large-scale data set using Amazon Mechanical Turk to crowdsource viewpoint annotations. This data set enabled us to design a neural view quality measure based on human preferences that is able to simulate human preference with high accuracy. Moreover, we evaluated our learned measure with respect to ambiguous annotations in the human-annotated data, which showed that our technique benefits from additional training data while being robust to the ambiguities at the same time. Lastly, we evaluate the generalization ability of our model to unseen model categories, which resulted only in a small drop in performance. Furthermore, we provide two methods for fast inference of our measure. The first method estimates the view quality measure of an image directly by using convolutional neural networks. The second method uses point convolutional neural networks, to predict the view direction with the best view quality value from a 3D model directly. In the future, we would like to investigate the correlation of human quality measures, as the one proposed in this paper, with the performance of neural networks on different downstream tasks.

Acknowledgements

This work was supported in part by The Federal Ministry of Education and Research funding program - AuCity 3 - Kollaborative und adaptive Mixed Reality in der Hochschullehre am Beispiel des Bauingenieurwesens, between Magdeburg-Stendal University of Applied Sciences, the Bauhaus University Weimar and Ulm University.

Open access funding enabled and organized by Projekt DEAL.

References

[AMB17] Avidar D., Malah D., Barzohar M.: Local-to-global point cloud registration using a dictionary of viewpoint descriptors. In *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)* (2017), IEEE, pp. 891–899.

[BDP00] Barral P., Dorme G., Plemenos D.: Scene understanding techniques using a virtual camera. In *Proceedings of the Eurographics'00, Short Presentations* (2000), A. de Sousa and J. Torres (Eds.).

[BFS*18] Bonaventura X., Feixas M., Sbert M., Chuang L., Wallraven C.: A survey of viewpoint selection methods for polygonal models. *Entropy 20*, 5 (2018), 370.

[BT52] Bradley R. A., Terry M. E.: Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika 39*, 3/4 (1952), 324–345.

[BTB99] Blanz V., Tarr M. J., Bülthoff H. H.: What object attributes determine canonical views? *Perception 28*, 5 (1999), 575–599.

[DCG10] Dutagaci H., Cheung C. P., Godil A.: A benchmark for best view selection of 3D objects. In *Proceedings of the ACM Workshop on 3D Object Retrieval* (2010), pp. 45–50.

[Fis53] Fisher R. A.: Dispersion on a sphere. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 217*, 1130 (1953), 295–305.

[FSG09] Feixas M., Sbert M., González F.: A unified information-theoretic framework for viewpoint selection and mesh saliency. *ACM Transactions on Applied Perception (TAP) 6*, 1 (2009), 1–23.

[FZS*20] Fang J., Zhou D., Song X., Jin S., Yang R., Zhang L.: Rotpredictor: Unsupervised canonical viewpoint learning for point cloud classification. In *Proceedings of the 2020 International Conference on 3D Vision (3DV)* (2020), IEEE, pp. 987–996.

[GRMS01] Gooch B., Reinhard E., Moulding C., Shirley P.: Artistic composition for image creation. In *Proceedings of the Eurographics Workshop on Rendering Techniques* (2001), Springer, pp. 83–88.

[Gut46] Guttman L.: An approach for quantifying paired comparisons and rank order. *The Annals of Mathematical Statistics 17*, 2 (1946), 144–163.

[HRV*18] Hermosilla P., Ritschel T., Vazquez P.-P., Vinacua A., Ropinski T.: Monte Carlo convolution for learning on non-uniformly sampled point clouds. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2018) 37*, 6 (2018), 1–12.

[HS97] Hoffman D. D., Singh M.: Salience of visual parts. *Cognition 63*, 1 (1997), 29–78.

[HSK*12] Hinton G. E., Srivastava N., Krizhevsky A., Sutskever I., Salakhutdinov R. R.: Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580* (2012).

[JLC*09] Jagadeesan A. P., Lynn A., Corney J. R., Yan X., Wenzel J., Sherlock A., Regli W.: Geometric reasoning via internet crowdsourcing. In *Proceedings of the 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling* (2009), pp. 313–318.

[KB14] Kingma D. P., Ba J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[Ken48] Kendall M. G.: Rank correlation methods. Charles Griffin Book Series, Oxford, 1948.

[KS40] Kendall M. G., Smith B. B.: On the method of paired comparisons. *Biometrika 31*, 3/4 (1940), 324–345.

[KTL*17] Kim S.-h., Tai Y.-W., Lee J.-Y., Park J., Kweon I. S.: Category-specific salient view selection via deep convolutional neural networks. *Computer Graphics Forum 36* (2017), 313–328.

[KZS15] Koch G., Zemel R., Salakhutdinov R.: Siamese neural networks for one-shot image recognition. In *Proceedings of the ICML Deep Learning Workshop* (2015), vol. 2.

[LG14] Liang L., Grauman K.: Beyond comparing image pairs: Setwise active learning for relative attributes. In *Proceedings of* *the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 208–215.

[LVJ05] Lee C. H., Varshney A., Jacobs D. W.: Mesh saliency. *ACM Transactions on Graphics 24*, 3 (2005), 659–666.

[Mil56] Miller G. A.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review 63*, 2 (1956), 81–97.

[Mos51] Mosteller F.: Remarks on the method of paired comparisons: Ii. the effect of an aberrant standard deviation when equal standard deviations and equal correlations are assumed. *Psychometrika 16*, 2 (1951), 203–206.

[NSGP*05] Neumann L., Sbert M., Gooch B., Purgathofer W., et al.: Viewpoint quality: Measures and applications. In *Proceedings of the 1st Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging* (Aire-la-Vile, 2005), The Eurographics Association Press, pp. 185–192.

[PB96] Plemenos D., Benayada M.: Intelligent display in scene modeling. new techniques to automatically compute good views. In *Proceedings of the International Conference GRAPHICON'96* (July 1996).

[QYSG17] Qi C. R., Yi L., Su H., Guibas L. J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the Advances in Neural Information Processing Systems* (2017), vol. *30*.

[RA11] Radinsky K., Ailon N.: Ranking from pairs and triplets: Information quality, evaluation methods and query complexity. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining* (2011), pp. 105–114.

[RBTH10] Rusu R. B., Bradski G., Thibaux R., Hsu J.: Fast 3D recognition and pose using the viewpoint feature histogram. In *Procceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010), IEEE, pp. 2155–2162.

[RWL*21] Ru C., Wang F., Li T., Ren B., Yan X.: Outline viewpoint feature histogram: An improved point cloud descriptor for recognition and grasping of workpieces. *Review of Scientific Instruments 92*, 2 (2021), 025010.

[SHVR21] Schelling M., Hermosilla P., Vázquez P.-P., Ropinski T.: Enabling viewpoint learning through dynamic label generation. *Computer Graphics Forum* (2021). http://doi.org/10.1111/cgf.142643.

[SLF*11] Secord A., Lu J., Finkelstein A., Singh M., Nealen A.: Perceptual models of viewpoint preference. *ACM Transactions on Graphics (TOG) 30*, 5 (2011), 1–12.

[SS02] Stoev S. L., Straßer W.: A case study on automatic camera placement and motion for visualizing historical data. In *Proceedings of the IEEE Visualization, 2002. VIS 2002.* (2002), IEEE, pp. 545–548.

[Thu27] Thurstone L. L.: A law of comparative judgment. *Psychological Review 34*, 4 (1927), 273–286.

[TK01] Tarr M. J., Kriegman D. J.: What defines a view? *Vision Research 41*, 15 (2001), 1981–2004.

[TMWS12] Tao J., Ma J., Wang C., Shene C.-K.: A unified approach to streamline selection and viewpoint selection for 3D flow visualization. *IEEE Transactions on Visualization and Computer Graphics 19*, 3 (2012), 393–406.

[VBP*09] Vieira T., Bordignon A., Peixoto A., Tavares G., Lopes H., Velho L., Lewiner T.: Learning good views through intelligent galleries. *Computer Graphics Forum 28* (2009), 717–726.

[VFSG06] Viola I., Feixas M., Sbert M., Groller M. E.: Importance-driven focus of attention. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 933–940.

[VFSH01] Vázquez P.-P., Feixas M., Sbert M., Heidrich W.: Viewpoint selection using viewpoint entropy. In *Proceedings of the VMV* (2001), vol. 1, Citeseer, pp. 273–280.

[WSK*15] Wu Z., Song S., Khosla A., Yu F., Zhang L., Tang X., Xiao J.: 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 1912–1920.

[YLLY19] Yang C., Li Y., Liu C., Yuan X.: Deep learning-based viewpoint recommendation in volume visualization. *Journal of Visualization 22*, 5 (2019), 991–1003.

[ZFY20] Zhang Y., Fei G., Yang G.: 3D viewpoint estimation based on aesthetics. *IEEE Access 8* (2020), 108602–108621.

[ZIE*18] Zhang R., Isola P., Efros A. A., Shechtman E., Wang O.: The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 586–595.

APPENDIX A: PROOFS

In this section, we show that it is possible to reconstruct a function f on a finite set D solely from information about the binary classifications $f(v_1) < f(v_2)$, $v_1, v_2 \in D$, up to composition with a strictly monotonically increasing function. First, let us recall Prop. 1 from Section 4.

Proposition A.1. *This reconstruction* f^* *preserves the order induced by* f *on the points in* D*, i.e.*

$$f(v_1) < f(v_2) \Leftrightarrow f^*(v_1) < f^*(v_2), \qquad \forall v_1, v_2 \in D. \tag{3}$$

Proof. We show both directions subsequently.

'$\Rightarrow$':

Let $v_1, v_2 \in D$ with $f(v_1) < f(v_2)$.

Then

$$\{x \in D : f(x) < f(v_1)\} \subset \{x \in D : f(x) < f(v_2)\}$$

as

$$f(x) < f(v_1) \Rightarrow f(x) < f(v_2).$$

By the definitions (2) and (1) it follows that

$$count(v_1) < count(v_2) \Rightarrow f^*(v_1) < f^*(v_2).$$

'$\Leftarrow$':

Let $v_1, v_2 \in D$ with $f^*(v_1) < f^*(v_2)$.

Assume that $f(v_1) > f(v_2)$, then the first part of the proof implies $f^*(v_1) > f^*(v_2)$, which is a contradiction.

Further, assume $f(v_1) = f(v_2)$, then apparently $count(v_1) = count(v_2)$ and thus $f^*(v_1) = f^*(v_2)$ by definition (2), which is also a contradiction.

Thus $f(v_1) < f(v_2)$. □

Corollary A.1. *The reconstruction* f* *of* f *satisfies*

$$f^* = g \circ f, \tag{A.1}$$

for a strictly monotonically increasing function $g : \mathbb{R} \to \mathbb{R}$.

Proof. Let $\mathcal{I}$ be the image of f, i.e. $\mathcal{I} := \{f(v) : v \in D\}$.

Further, let $D^+ \subset D$, s.t. $f|_{D^+} : D^+ \to \mathcal{I}$ is bijective.

Then $f|_{D^+}$ is invertible on D^+, i.e. $\exists f^{-1} : \mathcal{I} \to D^+$.

Then the function g, defined as

$$g : \mathcal{I} \to \mathbb{R}$$
$$g(x) := f^* \circ f^{-1}(x),$$

satisfies $f^* = g \circ f$.

It is left to show that g is strictly monotonically increasing on $\mathcal{I}$.

Let $x_1, x_2 \in \mathcal{I}$ with $x_1 < x_2$. As $x_1, x_2 \in \mathcal{I}$ there exist $v_1, v_2 \in D^+$ s.t. $x_1 = f(v_1), x_2 = f(v_2)$.

From Prop. A.1, Equation (3), we can infer $f^*(v_1) < f^*(v_2)$. Then

$$\begin{aligned} g(x_1) &= f^*(f^{-1}(x_1)) = f^*(v_1) \\ &< f^*(v_2) = f^*(f^{-1}(x_2)) \\ &= g(x_2). \end{aligned}$$

□

This is also the best possible reconstruction from binary classifications, as is clarified below.

Proposition A.2. *Let* $f_1, f_2 : D \to \mathbb{R}$ *be two functions, s.t.* $f_2 = g \circ f_1$ *for a strictly monotonically increasing function* $g : \mathbb{R} \to \mathbb{R}$. *Then the two functions* f_1 *and* $f_2 := g \circ f_1$ *yield the same binary classifications, i.e.*

$$f_1(v_1) < f_1(v_2) \Leftrightarrow f_1(v_1) < f_2(v_2).$$

Proof. The strict monotonicity of g directly implies that

$$\begin{aligned} \forall v_1, v_2 \in D : f_1(v_1) < f_1(v_2) &\Leftrightarrow g(f_1(v_1) < g(f_1(v_2)) \\ &\Leftrightarrow f_2(v_1) < f_2(v_2). \end{aligned}$$

□

APPENDIX B: ABLATION STUDIES

In this section, we describe the ablation studies we carry out to validate our design decisions:

Handcrafted versus neural features. To evaluate the improvements introduced by the neural feature extraction module, i.e. the convolutional neural network, we substituted the features computed by this module with a list of different handcrafted view quality measures as in Secord *et al.* [SLF*11]. The rest of the Siamese network remains the same, the features of both images are concatenated and processed by the MLP, which predicts the final probability. The results of the experiments are presented in Table B.1, where we can see that, as expected, neural features always achieve better performance than handcrafted view quality measures.

Data set sparsity. In our data set, we annotated each model with 18 views. However, previous work has favoured the number of annotated views per model instead of a large variety of different annotated models, using 240 images per model instead [SLF*11]. Therefore, in this experiment, we evaluate which setup allows for a better generalization of our neural network, large number of models annotated with a few images or few models more densely annotated. We selected five categories from ModelNet40, *airplane, chair, flower pot, sofa, toilet*, and selected seven models for each. For each model, we rendered 240 images, resulting in a total of 8400 images, and collected human preferences as described in Section 3. We call this new data set DENSE. Moreover, we created another data set with comparable number of images, by selecting 93 models for each of the five categories from our original data set, resulting in a total of 8370 images. We call this new data set SPARSE. We train our model

Table B.1: *Performance comparison between handcrafted versus neural viewpoint features. We also compare how the final probability is computed, with a goodness score in the Bradley and Terry model [BT52] versus using a neural network. The results indicate a better performance for neural extracted features in combination with a neural network to rank viewpoints.*

Features	Ranking	Acc.	AUC	AUPR
NEURAL	NEURAL	**79.9%**	**0.796**	**0.847**
NEURAL	GOODNESS SCORE	78.7%	0.787	0.840
HANDCRAFTED	NEURAL	77.8%	0.778	0.834
HANDCRAFTED	GOODNESS SCORE	76.7%	0.767	0.825

For each metric, we highlight the method performing best.

Table B.2: *Effect of the sparsity of annotations on the prediction ability of our model. A large number of models annotated with only a few images yields a higher generalization ability of our model than fewer models more densely annotated.*

	#models		#images				
Data set	cat.	total	model	total	Acc.	AUC	AUPR
Sparse	93	465	18	8370	**75.6%**	**0.760**	**0.820**
Dense	7	35	240	8400	74.4%	0.750	0.812

For each metric, we highlight the method performing best.

Table B.3: *Performance comparison using different sizes for the latent code in our viewpoint selection network.*

Size	Acc.
64	78.5%
128	**79.9%**
256	78.8%

For each metric, we highlight the method performing best.

with both data sets and compare the performance of our network. We can see in Table B.2 that the model trained with the Sparse data set achieves higher accuracy on all metrics, confirming that the neural network generalizes better with numerous different models even if the annotations are highly sparse.

Probability computation. Previous work [SLF*11] has suggested computing a goodness of score from the image features that can be used in the Bradley and Terry model [BT52] to compute the final probability of $f(x) < f(v)$. We use instead a neural network that takes both feature vectors and predicts the final probability. In this ablation study, we compare both methods with neural and handcrafted image features. Results are presented in Table B.1, where we can see that our neural probability predictor always achieves higher accuracy than the goodness score approach used by Secord *et al.* [SLF*11].

Features size. In this experiment, we evaluate the effect of the feature vector size in the final accuracy of the model. We train different models with feature vector sizes of 64, 128 and 256. We can see in Table B.3 that a feature vector size of 128 results in the higher accuracy.

Table B.4: *We averaged the performance of our models over 10 training runs and report the mean and standard deviation of all metrics.*

Data set	Acc.	AUC	AUPR
Unfiltered	79.9% ± 0.004	0.796 ± 0.004	0.847 ± 0.003
Filtered	79.7% ± 0.010	0.800 ± 0.010	0.850 ± 0.008

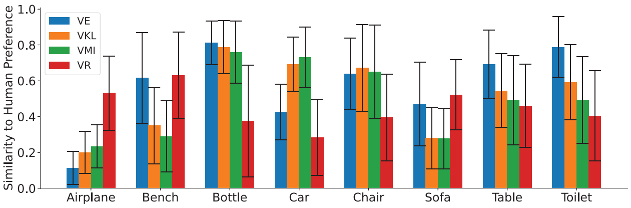


Figure C.1: *Comparison between handcrafted quality measures and their similarity to our human based quality measure on the best view prediction task.*

Performance statistics. In a last experiment, we measure the variance in performance of our model over several training runs. Therefore, we average the scores of 10 models optimized using identical hyperparameters. In Table B.4, we show the averaged results for models trained on the unfiltered and filtered data set, reporting the mean and standard deviation of the corresponding metric.

APPENDIX C: QUANTITATIVE RESULTS ON THE BEST VIEW PREDICTION TASK

In Section 7.3, we showed qualitative comparison results between our human-based and four handcrafted view quality measures. In this section, we quantitatively evaluate how handcrafted view quality measures perform using our view quality metric as ground truth. First, as an upper bound, we measure how the network of Schelling *et al.* [SHVR21] performs when it is trained with our view quality measure. For each predicted viewpoint, we compute the nearest neighbour to one of the 1000 sampled points in the sphere and compute the mean accuracy per category: airplane 92%, bench 87%, bottle 86%, car 94%, chair 89%, sofa 95%, table 92% and toilet 84%.

To evaluate the handcrafted view quality measures, we use the same procedure, but we train the network of Schelling *et al.* [SHVR21] using the different handcrafted measures instead. In Figure C.1, the results show strong differences between view quality measure and category.