No-Reference Point Cloud Quality Assessment

Feature Relevance Assessment for No-Reference Quality Assessment of 3D Point Cloud Data

IN5000: Master Thesis

Gwennan Smitskamp

Delft University of Technology

TU Delft

No-Reference Point Cloud Quality Assessment

Feature Relevance Assessment for No-Reference Quality Assessment of 3D Point Cloud Data

by

Gwennan Smitskamp

Student Name	Student Number
G.M. Smitskamp	4349822

Professor:	Prof. Dr. Pablo Cesar
Daily Supervisor:	Dr. Irene Viola
Project Duration:	September, 2021 - September, 2022
Faculty:	Electrical Engineering, Mathematics and Computer Science, TU Delft

Thesis committee:	Prof. Dr. Pablo Cesar	TU Delft, CWI, supervisor
	Dr. Petr Kellnhofer	TU Delft
	Dr. Irene Viola	CWI, external expert

Cover: Autodesk ReCap Scan Regions example (Modified)

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

CWI
Centrum Wiskunde & Informatica

Preface

With this thesis I am in the final stage of obtaining my Master of Science degree by finishing the Master Computer Science at the TU Delft. I started this project when everyone was slowly returning to the office after the Covid19 pandemic, and it feels like we are still not at the end of it. It was great to get to work outside my own home again after doing all my master courses inside. I am happy to have found a thesis internship at a national research institute.

I did my research in Amsterdam at Centrum Wiskunde & Informatica, where I did my thesis under supervision of Irene Viola and Pablo Cesar. I want to thank them for their guidance and insights during this project, and providing me with data and information needed for the project. I hope my work contributes to the work they are doing in the field of interactive systems in 3D environments. I wish the Distributed & Interactive Systems Group much luck and more fun in the future, as I really enjoyed my time with them, both inside and outside the office.

Gwennan Smitskamp
Delft, September 2022

Summary

The need to capture environments and objects in 3 dimensions to produce a high quality digital representation is proving to be useful in many applications in the world, where there is an increasing dependence on digital spaces. Point Clouds are a data type to represent 3D objects and scenes. During the processing of the point cloud data, undesired changes to the point cloud can be introduced or information can get lost. Automatic detection and classifying of the quality of a point cloud is needed, to regulate this data type for use in 3D environments, and speed up the processing of point cloud objects in real-time applications.

In this thesis, we look into the problem of no-reference point cloud quality assessment. Full-reference quality metrics compare a distorted point cloud with its original, while no-reference or blind metrics only look at a single point cloud. This task is more challenging than full-reference, as there is less direct information available to work with. We approach the challenge of how the perceptual quality of a point cloud can be estimated when a high-quality reference point cloud is not available, by analyzing if any point cloud properties correlate with the perceptual quality.

We try to find point cloud properties that are effective to use in a model to predict a user given quality score. We select multiple local properties to transform into global descriptors, by taking multiple statistical attributes of the data on all the points to combine into scalars. We analyze the created global descriptors to find linear correlations with the quality score, and found many features that to some measure linearly correlate with some distortion types. With the selected global descriptors, we use Support Vector Regression to find the fitting of the features to the quality score. Varying performances are achieved by training the models on different data subsets, and different fitting kernel functions.

The results show some distortions on unknown objects are properly predictable using fitted weights on this distortion. However, a single defining quality descriptor is not yet found. While the results show potential, our model is not robust and fast enough to perform a reliable assessment of quality in a real-time environment. We are just at the beginning of exploring no-reference quality assessment, and while the current methods are not yet appliable in real-time scenarios, future work can fill many knowledge gaps to reach this goal.

Contents

1

Introduction

In section 1.1 we introduce the research field which leads to the problem of this thesis. Next, the research questions we are looking to answer to address the thesis problem are explained in section 1.2. The contributions to the research field are summed up in section 1.3. How the rest of this thesis report is structured can be found in section 1.4.

1.1. Research field

The need to capture environments and objects in 3 dimensions to produce a high quality digital representation is proving to be useful in many applications in the world, where there is an increasing dependence on digital spaces. Humans experiencing virtual reality with higher resolution VR headsets, the computer vision of autonomous cars, and precise and efficient environment scans in the construction industry are examples of applications of the **3D point clouds**, a set of points with x,y,z coordinates representing their position in a space.

Figure 1.1: MediaScape XR, real time virtual reality demo. Images: CWI

To produce useful and trustworthy results for these applications, the point clouds used require a minimum fidelity or quality. For VR experiences, this quality is defined

by how users perceive the 3D objects and if this is visually pleasing or disrupting the experience. In the field of 3D Graphics Standardization for the point clouds, there is still much space to contribute to the metrics to use for automatically judging the quality of a point cloud. While there has been progress in predicting the quality of a degraded point cloud given a reference point cloud, **objective metrics for blind quality assessment**, when no reference point cloud is available, has just recently begun showing potential with few new methods.

1.1.1. Challenge

To estimate quality with a reference point cloud available, descriptive properties of the whole point cloud or individual points are compared between a point cloud and its ground truth reference to estimate a degradation score between them. Many methods have been designed for this task, referred to as full-reference quality assessment, varying in parts like descriptors or distance measures, some methods have managed to get very accurate estimates of the visual quality of point clouds. A challenge of estimating a quality score of a point cloud in a no-reference setting is the lack of ground truth comparison to find how much it differs from ”perfect”. Examples of situation where a no-reference quality estimation might be needed include estimating the success of a reconstruction after a compressed point cloud is transferred, or judging if a new recording of a point cloud object was successful.

A way to be able to estimate the quality of a point cloud in these situations could be to find a combination of descriptors that correlate with the quality score. We are working in the context of social interactive systems, so this score can be gathered from a user study where human users have rated the quality of experience of digitally interacting with the 3D point cloud objects. The goal of this project is to investigate multiple descriptive properties of the point cloud datatype to find their effectiveness in estimating perceptual quality of a point cloud when there is no ground truth point cloud available. The point cloud type that will be focused on is static surface point clouds, meaning a single frame of a point cloud observation of the visible outer layer of an object.

1.2. Research Questions

With the goal to **estimate perceptual quality of a point cloud when no high-quality reference point cloud is available** in mind, the following questions follow:

Research question: How can the perceptual quality of a point cloud be estimated when a high-quality reference point cloud is not available?

To find a method to answer this question, a hypothesis is that there are properties that high quality point clouds share, and this property value changes when the quality goes down. Since the whole point cloud is to be judged, these properties would be global descriptors, leading to the following question:

Sub question: Can global descriptors be used to create a model to estimate the

perceptual quality of a point cloud?
To test the hypothesis, multiple of these global descriptors need to be evaluated to test for the potential to contribute effectively to the model. We hypothesize that there exist descriptors that correlate with perceptual quality, which will be effective in the model.

Sub question: Do any global descriptors correlate with perceptual quality?
There are already multiple global descriptors to test, but also local and point features have shown potential to correlate with quality in existing full-reference quality assessment methods. To design a no-reference metric, we want to adjust some of these features to global descriptors and make use of these as well.

Sub question: Can (existing) local features of point clouds be adapted to global descriptors?

Another option would be to design completely new descriptors to best fit the task, however we want to first review the current work in point cloud descriptors before extending this already huge set of options.

1.3. Contribution

- Investigation of the effectiveness of multiple types of features (color and geometry) of point clouds and different kinds of statistical distribution models, in no-reference point cloud quality assessment context.
- Implementation of pipeline to easily produce and evaluate no-reference point cloud quality assessment models.
- Potential no-reference point cloud quality assessment metric to use in the pipeline of point cloud processing in real time applications.

1.4. Project Plan

To answer these research questions and contribute to the field, we follow the following plan and report structure.

The first step is getting familiar with point clouds in the context of quality assessment. The current state-of-the-art in full and no reference point cloud quality assessment is reported in a small literature study, together with more background of the processing of point clouds in the context of real time applications. After this, a selection of point cloud descriptors is explored on the potential to indicate a quality score, together with ways to transform these descriptors to global features to use in quality prediction. We then set up an experiment to find the correlation between the designed features and a quality score. The highest potential descriptor from the correlation analysis and more relevance analysis will lead to a feature selection to use in a supervised machine learning prediction model for the quality score. The performance of this no-reference prediction model can be compared to previous metrics, however due to the few direct comparable methods, this study is to be seen more as an exploration of feature relevance and potential in this upcoming field. The conclusion and discussion will answer

the research questions and review the experiment and results critically, we end with future work suggestions.

2

Background and Related work

In this chapter, the related work and a knowledge base of the research field and thesis context is presented. After a quick introduction into the 3D data types in section 2.1, we walk through a couple of processes that are usually included in the point cloud processing pipeline in real time applications in section 2.2. Then we will focus on the quality assessment state-of-the-art in the point cloud field in both full- and no-reference context in section 2.3.

2.1. Introduction to Point Cloud 3D data

Point Clouds are a data type to represent 3D objects and scenes. The basis of a point cloud is a set of points with x,y,z coordinates representing their position in a space, the data structure is unconnected and does not need to be ordered in any way. Next to location data, points can have any additional properties like color, reflectances, etc. Another datatype for 3D objects is the Polygon Mesh, a structural build consisting of connected points forming (commonly triangle) surfaces.

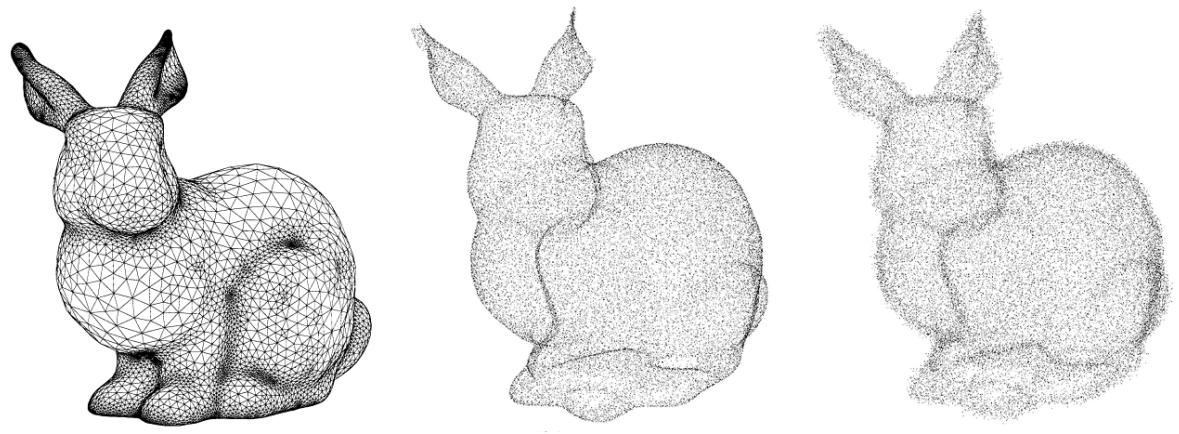


Figure 2.1: Stanford bunny surface mesh, point cloud and noisy point cloud

In figure 2.1, the surface of the Stanford bunny 3D object is rendered as a mesh and as point clouds. The mesh forms a solid surface layer, while the point cloud points are a dense sampling of the surface. By adjusting to proper point size and colors, a point cloud can directly be rendered into a pleasing visual portrayal of the object, without a solid surface, the human perception still interprets it as such. This does require a very dense sampling of the object surface, leading to point clouds being large data structures of millions of points.

Efficient storage, transmission, compression and reconstruction algorithms ensure point clouds are a very efficient datatype for real-time virtual reality applications. However, during the processing of the point cloud data, undesired changes to the point cloud can be introduced or information can get lost. These changes can impair on the visual quality in the final application (third object in figure 2.1). It is a time-consuming task to manually assess on every step to make sure the point cloud is still usable for a visual application. Introducing an autonomous quality assessment module in real-time systems could be a useful addition to the point cloud processing field.

2.2. Point Cloud processing pipeline

Point clouds are used in many applications, the main context of this project is the application for real-time social virtual reality experiences. This includes capturing a real-life object, like the user of the experience, and transforming this capture to a life 3D rendering in the virtual environment. This virtual environment is rendered on a separate (hardware) system from the capture (hardware) system, and the data has to be transferred between them to create this setting. A summary of this process is the pipeline that includes the acquisition, compression and reconstruction, and rendering of the point clouds.

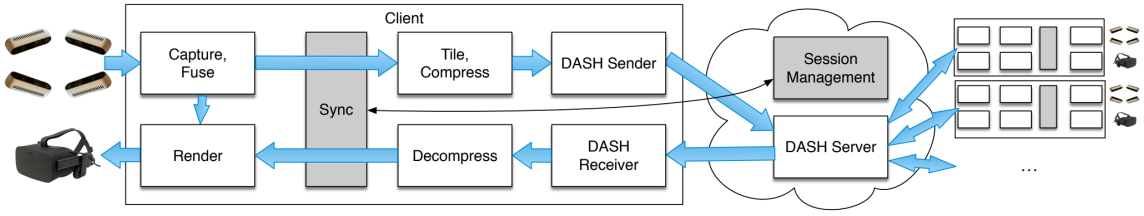


Figure 2.2: Example of a real time capture and display volumetric videoconferencing pipeline [21]

2.2.1. Acquisition of point cloud objects

Point cloud capture and construction from sensor data can be captured by a single view laser sensor like LiDaR, laser sensors calculate the distance to an object by projection from the round-trip delay of (electromagnetic) waves from the sensor to the object and the direction of the laser to get a location coordinate [18]. Another sensor to capture 3D data is depth sensors, like the Kinect. The Kinect uses a near-infrared laser pattern projector and an IR camera to triangulate points in 3D space. Together

with an RGB camera, this results in a colored 3D scene construction [14]. When using multiple sensors, the recorded information can be combined and processed to a complete point cloud object with a complete surface of all view angles.

If such hardware is not available, a set-up with multiple camera angles can serve as a base to translate the 2d images of real-life to 3D objects [34]. With the known orientation and position of the cameras, matching key-points in their recorded 2D images can be projected into a depth map through triangulation techniques.

Point clouds can also be directly created digitally, for applications with scenes that are not directly based on real life scenarios. VR (or just 3D) scenes can be set in fantasy settings designed digitally, and players can be represented by avatars (virtual identities) that are not based on any 3D sensor data. A very simple way to add some 3D information on the movements of a player into the VR/3D environment is with a couple of sensors for the location of the VR-headset and the location of the handheld controllers, to model positioning of the avatar after the user.

2.2.2. Compression and reconstruction

Using point cloud in the context of placing a user in a virtual reality, it can be the case that the point cloud needs to be transmitted to a different (part of) the system or location, which has bandwidth limitations over network connections. Since point clouds can require a large amount of space to store all data points, and therefore transmission bandwidth, compression and (after transferring) reconstruction tasks are needed for efficient processing.

The Motion Picture Experts Group (MPEG) has selected two point cloud compression(PCC) technologies as standard practice, video-based PCC (V-PCC) and geometry-based PCC (G-PCC). The V-PCC method is based on 3D-to-2D patch projections, saving the 2D images and metadata needed to reconstruct the point cloud with high accuracy [16]. Using a G-PCC method for compression is based on encoding the point cloud in 3D space using efficient data-structures, multiple variants of algorithms are summed under this compression method. The point cloud is described into an Octree construction [31], or surface approximation by a series of triangles in a pruned Octree (triangle soup), and encoded by a transforming tool like Region Adaptive Hierarchical Transform (RAHT) or the Lifting Transform.

Next to projection-based and geometry-based encoding, compression techniques can also be based on (Deep) learning. Neural network implementations using geometry and/or color information for point cloud compressions are showing promising results [36]. In figure 2.3 we see the reconstruction results of the MPEG methods, a learned point cloud geometry compression, and an octree based compression used in a popular point cloud processing project Point Cloud Library [37].We see in the error map that the reconstructed point cloud contains some distortions compared to the original. This error map is an example of full reference quality assessment, in this case based on mean-squared-error with point-to-point distance.

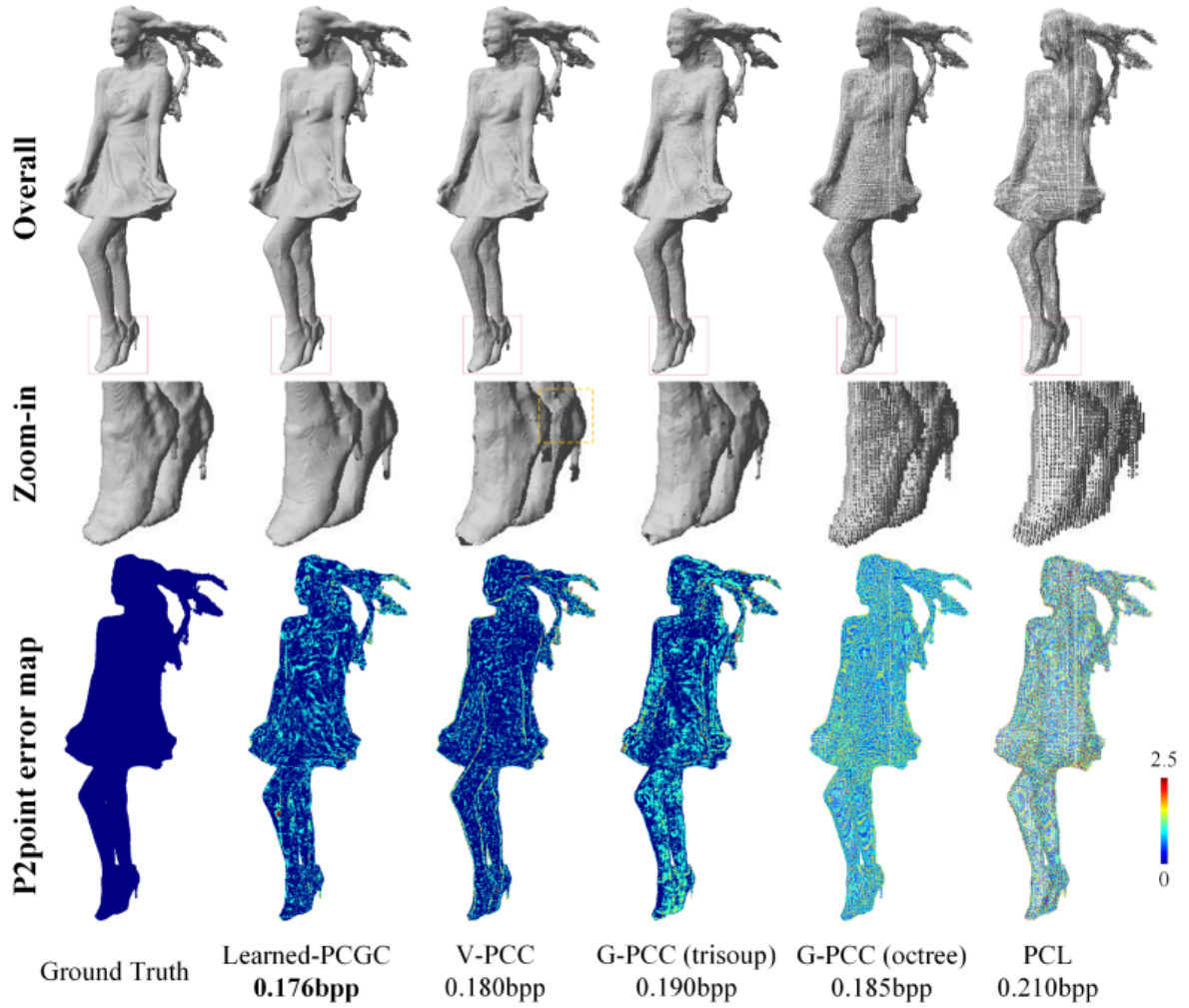


Figure 2.3: Visual comparison of "redandblack" with reconstruction after compression for ground truth, Learned-PCGC, V-PCC, G-PCC, and the octree-based PCL [46]

2.2.3. Rendering 3D environment

A point location is a mathematical concept with the interpretations that at that point there should be a surface visible. To display this surface, the point information needs to be rendered in such a way the interpretation of the data is represented by the visualization. The simple idea is to render a sphere at each point, without filling the space between in case the spheres don't touch [38], or add a surface reconstruction algorithm to show a mesh [25]. To view these digital objects, every virtual space will be projected on some 2D screen, like for both computer monitors and VR glasses. The choice of visualization method can influence the subjective quality evaluation in user studies, which is something to be aware of in this field of study.

2.3. Quality Assessment of Point Clouds

Automatic quality assessment of point clouds can improve the field by reducing the effort of manually assessing the performance of a point cloud processing task. Another

word for a quality assessment method is a quality metric. In this context, the goal of a metric is to estimate the perceptual quality of a point cloud. Given a perceptual quality score, the performance of a metric can be evaluated on accuracy. The state-of-the-art in point cloud quality assessment (PCQA) does not have a single standard of performance evaluation, like each basing their performance scores on experiments with different datasets. When a new dataset releases, they often come with a performance evaluation of current state-of-the-art metrics [6, 53].

A way to split metrics in approach and application is whether the metric uses a reference point cloud or not to measure the quality of a point cloud. Full-reference quality metrics compare a distorted point cloud with its original, while no-reference or blind metrics only look at a single point cloud. Using some information of the original point cloud but not the complete one is referred to as reduced reference. More ways to categorize metrics is the type of the target point cloud (with or without color values, static or dynamic, volumetric or surface), or if it is designed for a specific target distortion and noise. The methods used in metrics can be categorized in terms of approaches like based on projection/multi-view, (deep-)learning, histograms, topology etc. The focus of the quality assessment of this project is a no-reference metric for static point clouds based on global geometric and color information, in this section we do look broader across categories to gain enough insight to produce most optimal results.

2.3.1. Full reference

Full reference metrics compare point, local or global features of a distorted point cloud to the corresponding part of the original. A base of finding an error score is to measure the mean Euclidean, Hausdorff, or other distance measures of all corresponding points between the two point clouds, referred to as point-to-point [9, 22]. The projected distance along the normal direction is called point-to-plane, and can be normalized as a Peak Signal to Noise Ratio metric (PSNR) [43]. A single point to a group of points and their distributions can be measured, point-to-distribution, Javaheri et al. [23] used this in a metric using a Mahalanobis distance.

From the location information of all the points, geometry-related quantities can be extracted to further investigate the similarities between a distorted and an original point cloud. We will broadly go over which types the following metrics use, but in the next chapter we will go into more detail for specific descriptors.

Dispersion statistics of geometry-related quantities together with angular similarity between the normal vectors [7] and curvature values, combined with the color-related quantities like luminance values have been used to produce a structural similarity score [2]. Meynet et al. [32] try combinations between 3 geometric comparisons and 5 color comparisons to find the best performing feature vector for their PCQM metric. Using Principal Component Analysis (PCA) on local point sets extracts eigenvalues and eigenvalues per point to use to calculate more geometry descriptors, PointPCA [4] used 15 PCA descriptors and the luminance values to compare the point clouds.

The histogram distance between point clouds of the color histogram on the lumi-

nance channel and YCbCr channels, together with a measure of difference between the correlograms associated with reference and distorted point cloud form a color focused metric [45]. Both color histograms and geometric distance histograms are combined in the metric BitDance [10].

Yang et al. [52] compared the point potential energy of point clouds to assess the distortion. The histograms of local luminance patterns of two point clouds are compared using the Euclidean distance for the metric of Diniz et al. [12, 11].

When projecting the 3D point cloud onto a 2D space, image quality assessment methods serve effective point cloud quality assessment metrics as well [40, 42]. With the GraphSIM metric [50], a graph is constructed for both reference and distorted point clouds to calculate the similarity index for the quality assessment.

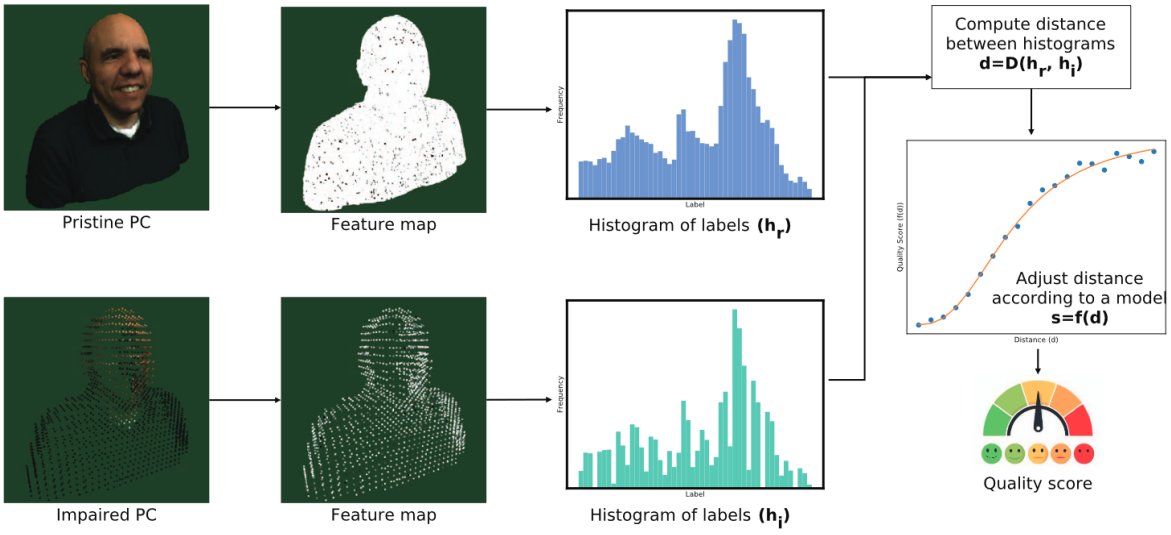


Figure 2.4: Pipeline of Full Reference PCQA method [11]

2.3.2. Reduced reference

When the original point cloud is not completely available, all these full-reference metrics will no longer be usable to assess the quality of the available point cloud. Viola et al. [44] sketched the situation of compressing a point cloud to transfer and then assess the level of degradation introduced by coding and transmission distortions. They introduced a reduced reference metric, where a few features were calculated before compression and saved in a few bits, to send along with the compressed point cloud and use these to compare and formulate a perceptual quality score. The color fluctuation over a geometric distance, and the global feature of the color block mean variance are used in the reduced-reference metric of Liu et al. [29].

2.3.3. No reference

When no original point cloud to compare the point cloud of which you need a quality rating for is available, a no-reference metric is needed. This task is more challenging than full-reference, as there is less direct information available to work with. The com-

mon approach to this problem is a learning-based metric, where a prediction model is created based on point cloud properties relative to the quality score.

A set of hand-crafted features is used to create a prediction model in Chetouni at al. [8] (geometric distance, mean curvature and gray-level) and Zhang et al. [57] (eigenvalue and color based features) to learn a mapping to the ground truth mean opinion score. Liu et al. [28] maps the point cloud to 2D images using a multi-view projection strategy to extract a 384-dimensional feature vector, which is fed into two learning modules that combined calculate the quality of the degraded point cloud, this is extended in a following paper of Fan at al. [13]. IT-PCQA [51] uses established image quality assessment method to create a deep-learning and projection-based metric. Another deep learning model [30] was designed with a sparse convolutional neural network to extract the hierarchical features, and predict quality scores with a regression module of a grand new dataset they created for these learning-based approaches. A metric focused on predicting the quality of V-PCC encoded point clouds using a no-reference bitstream-layer model has been explored by Liu at al. [27].

It should be noted that some of these metrics have not been published and may have not gone through in-depth peer-review.

2.3.4. Data sets

Databases that are used to create and evaluate the PCQA metrics need a value of the perceptual quality gathered in a user study. A dataset usually contains multiple point cloud objects, each distorted with some distortion type, with multiple levels of the distortion. These point clouds are then reviewed in a user study by multiple people on some subjective scale, like how distracting the imperfections are. The average subjective score of each point cloud forms the ground truth quality value, usually labeled as the mean opinion score MOS, on a scale of one to five or ten.

In the work that introduced the WPC database [42], they summarized the available data for PCQA in the following table. The dataset LSPCQA used in the no-reference deep learning approach of Liu et al. [30] is not yet published.

Database name	Attribute	Source contents	Distortion type	Subject-rated point clouds
IRPC [24]	None, Color	6	PCL, G-PCC, V-PCC	54
vsenseVVDB [54]	Color	2	V-PCC	32
vsenseVVDB2 [55]	Color	8	Draco+JPEG, G-PCC, V-PCC	164
G-PCD [3]	None	5	Octree-pruning, Gassian noise	40
RG-PCD [7]	None	6	Octree-pruning	24
M-PCCD [6]	Color	8	G-PCC, V-PCC	244
PointXR [5]	Color	5	G-PCC	100
NBU-PCD 1.0 [7]	Color	10	Octree	160
CPCD 2.0 [19]	Color	10	G-PCC, V-PCC, Gassian noise	360
SJTU-PCQA [53]	Color	9	Octree, downsampling, color and geometry noise	378
ICIP2020 [35]	Color	6	G-PCC, V-PCC	90
3DMDC [33]	Color	5	QGeo, QCol, SGeo, SCol	80
SIAT-PCQD [48]	Color	20	V-PCC	340
WPC [41]	Color	20	Gassian noise, dowsampling, G-PCC, V-PCC	740

Table 2.1: Available databases with quality scores according to Su et al. [42]

The following set of databases offer a mix of distortion types and objects and will be used in this thesis: The SJTU-PCQA database [53] provides 378 public point cloud objects with a subjective quality score. It consists of 9 objects, each distorted with 7 types

of distortion on 6 levels. The 7 distortions are Octree-based compression (OT), Color Noise (CN), Downscaling (DS), Downscaling and Color noise (D+C), Downscaling and Geometry Gaussian noise (D+G), Geometry Gaussian noise (GGN), and Color and Geometry Gaussian noise (C+G). ICIP2020 [35] has 6 point cloud objects, each distorted with 3 types of distortions on 5 levels, the G-PCC variants of Octree-Lifting and TriSoup-Lifitng, and the V-PCC distortion. The IRPC [24] dataset has 6 objects, each with a low, medium and high quality version of G-PCC, V-PCC and an Octree-based compression scheme. The final dataset of M-PCCD [6] has 8 objects, each with 5 distortion types, the G-PCC variants of Octree-Lifting and -RAHT and TriSoup-Lifting and -RAHT with 6 levels of distortions, and V-PCC with 5 distortion levels.

2.3.5. Distorted point clouds

When a point cloud is introduced to changes that are unintentional or undesired, we have a distorted point cloud. Distortions can influence the visual looks of a point cloud, or reduce the informational value of it for computer vision tasks. Distortion can already be introduced by faulty recording of the 3D object, or be a byproduct of the processing and usage of the point cloud. A common introduction to distortion is the process of compressing and reconstruction of a point cloud, we see in the dataset table 2.1 G-PCC and V-PCC is a common distortion type to include in a database. The distortions in the datasets we will be using in this experiment are the following:

- Octree-based compression: adjusting the tree resolution by octree pruning removes leaf nodes.
- Color Noise: adjusting the color value of all or a subset of the points in the point cloud.
- Downscaling: Removing points from a point cloud randomly or systematically.
- Geometry Gaussian noise: shifting points with a random geometric shift with a Gaussian distribution.
- Video-based Point Cloud Compression (V-PCC): V-PCC can increase blockiness of a point cloud,
- Geometry-based Point Cloud Compression (G-PCC): the TriSoup process may create false edges and triangle shaped holes. Octree-based encoding suffers from a downsampling distortion.

Table 2.2: Distortion types in the dataset of Liu et al. [30], covering a wide range of impairments during production, coding, compression, transmission and presentation of point clouds. Reported with best full-reference metric.

No	Distortion Types	Typical Situation	Best FR-PCQA	No	Distortion Types	Typical Situation	Best FR-PCQA
#1	Color noise	Production	h.PSNRyuv	#18	Rayleigh noise	Transmission	PSNRyuv
#2	Color quantization with dither	Coding	PSNRyuv	#10	Saltpepper noise	Production, Transmission	MPED
#3	Contrast change	Production	h.PSNRyuv	#20	Change of color saturation	Presentation	PCQM
#4	Spatially correlated noise	Production	PSNRyuv	#21	Uniform noise (white noise)	Transmission	PSNRyuv
#5	Down sample	Coding	mseF,PSNR (p2point)	#22	Uniform geometry shifting	Transmission	MPED
#6	Gamma noise	Presentation	PSNRyuv	#23	VPCC-lossless G and lossy A	Compression	MPED
#7	Additive Gaussian noise	Production	PCQM	#24	VPCC-lossy G and lossless A	Compression	MPED
#8	Gaussian geometry shifting	Transmission	h.,PSNR (p2plane)	#25	VPCC-lossy G and lossy A	Compression	GraphSIM
#9	High frequency noise	Compression	PCQM	#26	AVS-limitlossy G and lossy A	Compression	mseF,PSNR (p2plane)
#10	Local loss	Transmission	GraphSIM	#27	AVS-lossless G and limitlossy A	Compression	h.PSNRyuv
#11	Local offset	Transmission	mseF,PSNR (p2point)	#28	AVS-lossless G and lossy A	Compression	h.PSNRyuv
#12	Local rotation	Transmission	h.,PSNR (p2point)	#29	GPCC-lossless G and lossy A	Compression	PCQM
#13	Luminance noise	Presentation	h.PSNRyuv	#30	GPCC-lossless G and nearlossless A	Compression	PCQM
#14	Mean shift (intensity shift)	Production	h.PSNRyuv	#31	GPCC-lossy G and lossy A	Compression	h.,PSNR (p2point)
#15	Multiplicative Gaussian noise	Production, Transmission	PCQM	#32	Octree Compression	Compression	h.,PSNR (p2point)
#16	Poisson noise	Production	PSNRyuv	#33	Poisson Reconstruction	Reconstruction	mseF,PSNR (p2point)
#17	Quantization noise	Coding	PCQM				

3

Theoretical basis

In this chapter, the difference in approach between full reference and no reference quality assessment metrics is demonstrated, and the idea of how to use this knowledge in the thesis experiment follows from this in section 3.1. We'll then look at the possible local point cloud descriptors to use in our no-reference metric in section 3.2, and how to transform them into global features in section 3.3. A process to make a selection which features to continue to use is a quality metric is explained in section 3.4, and the method to make a quality prediction model finishes this chapter in section 3.5.

3.1. Observation

As stated in section 2.3, point cloud quality metrics can be based on different methods, the point-based metrics will be the base of this experiment. Our goal is finding point properties that can be used in no-reference metrics. In full-reference metrics using point to point comparisons, the descriptor difference indicates a quality score. Since this relative difference indicates a quality value, it might be the case that the direct value can be an indicator of quality as well. If this is the case, the point descriptors from full-reference metrics could be repurposed in a no-reference context. Where the difference value is fitted to the quality value in full-reference, we can experiment with the other statistical information from the point properties to see if a fit exists.

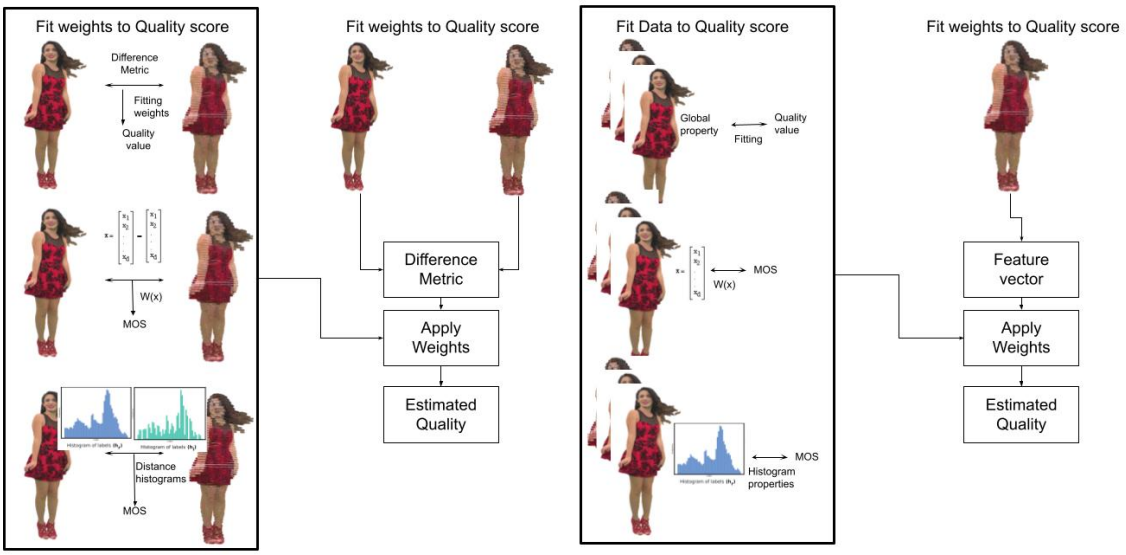


Figure 3.1: Process of fitting Full-Reference versus the process of fitting No-Reference metrics

Question Do any point features exist that have correlated statistical values with the quality value of the point cloud?

To answer this question, the following information is needed: Which point (cloud) properties have potential to correlate with the quality? Can a global value be extracted from these properties to fit to a quality score? Which global values should be selected to best fit to a quality score? How can the selected features predict the quality score?

3.2. Point cloud properties

From the available information of x,y,z coordinate, color values, etc., more descriptors can be extracted for a point, an area or the whole point cloud. Direct geometry based or color based (or combined) features can be extracted, like global means and variances of these values, more advanced local features (like entropy, energy, sparsity and curvature statistics per point), or global and local histograms and distributions. Most descriptors of a point are usually calculated using the surrounding points in the point cloud, like its k-nearest neighbors (KNN). It should be noted that the data of a point cloud does not have to be ordered in a specific way, so finding these k neighbors is an operation on the whole point cloud.

To find high potential features, we can search through the entire point cloud research field and test descriptors used in tasks like classification and object recognition, an overview of the state-of-the-art in point cloud descriptors was made by Han et al.[17]. However, we choose to start the exploration of this field with descriptors already used in full-reference metrics, as these have high potential to work and are thus a good kick-off point for this experiment.

PCA features

The set of geometric descriptors is based on Principal Component Analysis(PCA) to estimate shape properties of a point or area of the point cloud[4]. PCA is a dimensionality reduction method, that defines a set of principal components as linear combinations of the original data points. In a 3 dimensional context, the first component corresponds to the direction in which the projected data has the largest variance, then the next component maximizes the variance in a direction orthogonal to the first one. A common way of finding these components is using the covariance method, where we will use the found eigenvectors(principal components) and mainly their eigenvalues as a basis for the geometric features we are estimating. The following steps explain this method: A neighborhood $\mathbf{P}_i$ of k nearest points is defined for each point $\mathbf{p}_i$ in point cloud P:

$$\mathbf{P}_i = \text{knn}_k(\mathbf{p}_i)$$

The covariance matrix of this set is computed with :

$$C_i = \frac{1}{|\mathbf{P}_i|} \sum_{n=1}^{|\mathbf{P}i|} (\mathbf{p}_n - \bar{\mathbf{p}}_i)(\mathbf{p}_n - \bar{\mathbf{p}}_i)^T \tag{3.1}$$

and three eigenvalues can be found through eigenvalue decomposition, where $\lambda_1 > \lambda_2 > \lambda_3$:

$$C_i \cdot v_j = \lambda_j \cdot v_j, j \in 1, 2, 3$$

With these eigenvalues, various geometry features can be calculated, which have been successfully used in full-reference metrics:

- Curvature: the amount by which a curve deviates from being a straight line
- Anisotropy: the variations in geometric properties along different directions
- Linearity: the similarity to a straight line
- Planarity: similarity to a planar surface
- Sphericity: how closely the shape of an object resembles that of a perfect sphere
- Variation: describes the smoothness of the planar patch
- Omnivariance: gives the volume of the neighborhood and indicates whether the point cloud is locally scattered.
- Eigenval entropy: a measure of the order or disorder of 3D points
- Sum of eigenvalues: the total variation of the points from the center of a neighborhood

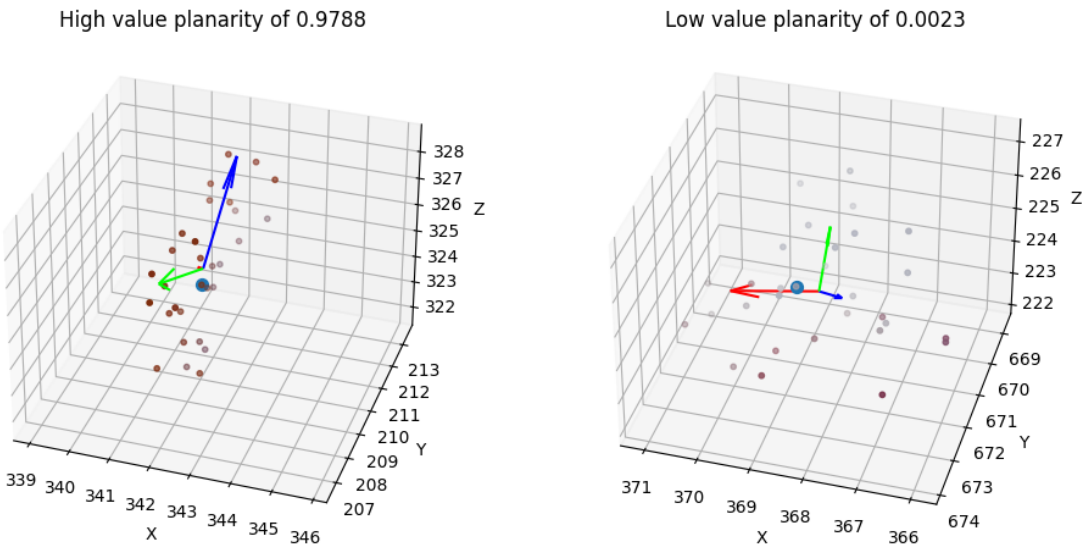


Figure 3.2: Example Principal Components Planarity values

Table 3.1: Definition of geometric descriptors

Feature	Definition
Curvature	$\lambda_3/(\lambda_1+\lambda_2+\lambda_3)$
Anisotropy	$(\lambda_1-\lambda_3)/\lambda_1$
Linearity	$(\lambda_3-\lambda_2)/\lambda_1$
Planarity	$(\lambda_2-\lambda_3)/\lambda_1$
Sphericity	λ_3/λ_1
Variation	$\lambda_1/(\lambda_1+\lambda_3+\lambda_3)$
Omnivariance	$\sqrt[3]{\lambda_1+\lambda_3+\lambda_3}$
Eigenval entropy	$\sum_{i\in\{1,2,3\}}\lambda_i\cdot ln(\lambda_i)$
Sum of eigenvalues	$\lambda_1+\lambda_2+\lambda_3$

The features are all calculated with the estimated PCA eigenvalues based on the neighborhood of the point in question. The result of these calculations differ per neighborhood size k, if k is too small a single outlier can cause an inaccurate estimation,

however too many neighbors might not detect enough individual properties and distortions we want to extract.

Color attributes

Many full reference quality assessment methods incorporate some color attributes in the assessment for point cloud with color values [45]. Color values can be represented in multiple ways, in point cloud data it is usually stored as RGB values. Other color spaces calculated from the RGB values can give additional information in terms of correlation with distortion. An example is the luminance value, where one can imagine the perceived brightness of an object strongly correlates to the perceptual experience of a human viewer (too dark or too bright or inconsistent). Different color spaces could better correlate with point cloud distortions. The CIELAB is used in the framework of Zhang et al. [57], calculated by converting RGB to XYZ, and XYZ to LAB (illuminant=D65, observer=2). L represents perceptual lightness, and a and b are the color scales. Hue and Chroma from the HCL color space are also included in the experiment

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 2.7688 & 1.7517 & 1.1301 \\ 1.0000 & 4.5906 & 0.0601 \\ 0 & 0.0565 & 5.5942 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.2}$$

$$f(t) = \begin{cases} \sqrt[3]{t} & \text{if } t > \delta^3, \delta = \frac{6}{29} \\ \frac{t}{3\delta^2} + \frac{4}{29} & \text{otherwise}, \delta = \frac{6}{29} \end{cases} \tag{3.3}$$

$$L^\star = 116\ f\left(\frac{Y}{Y_\mathsf{n}}\right) - 16 \tag{3.4}$$

$$a^\star = 500\left(f\left(\frac{X}{X_\mathsf{n}}\right) - f\left(\frac{Y}{Y_\mathsf{n}}\right)\right) \tag{3.5}$$

$$b^\star = 200\left(f\left(\frac{Y}{Y_\mathsf{n}}\right) - f\left(\frac{Z}{Z_\mathsf{n}}\right)\right) \tag{3.6}$$

$$Hue = \begin{cases} \arctan(b^\star/a^\star) & \text{if } \arctan(b^\star/a^\star) \geq 0 \\ \arctan(b^\star/a^\star) + 360° & \text{otherwise} \end{cases} \tag{3.7}$$

$$Chroma = \sqrt{a^{\star 2} + b^{\star 2}} \tag{3.8}$$

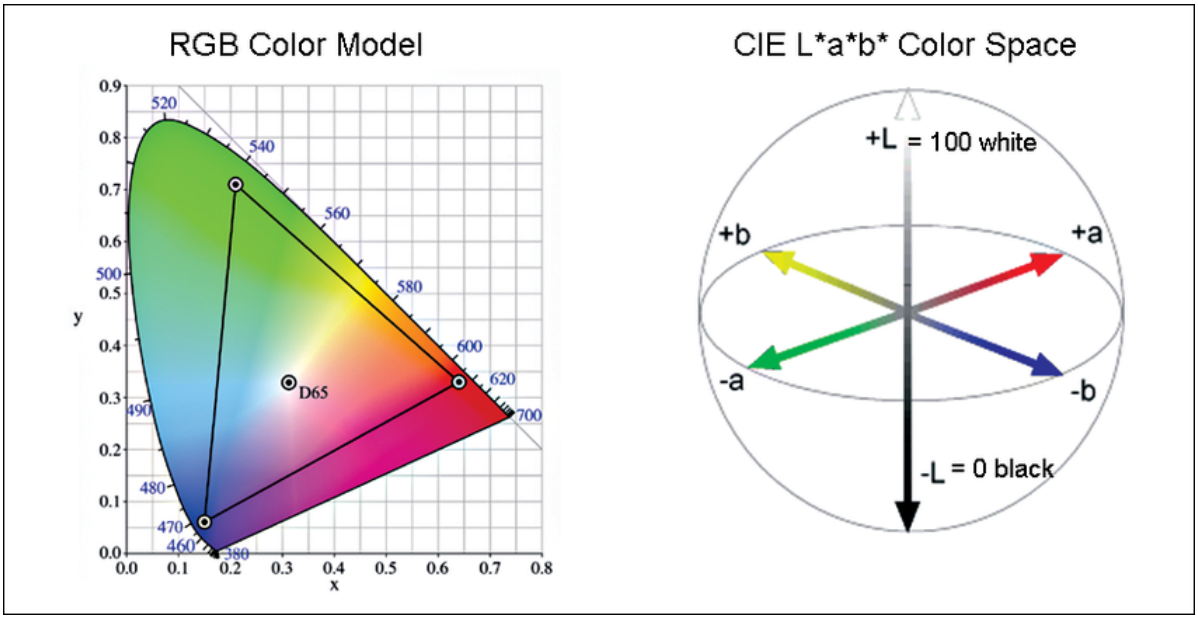


Figure 3.3: Visual representation of the color spaces RGB and CIELAB

Density Feature

A feature describing the average distance of the neighbors of a point could be described as a density value. This would without any other processing give information of the point cloud internal differences, as finding how many points are in dense areas and how many points are on a sparse surface, like holes or outliers. Since point cloud can be recorded in different scales, this feature might not be reliable when reduced to solely a global mean, but the distribution of the values are more interesting.

$$D_i = \frac{1}{|\mathbf{P}_i|} \sum_{n=1}^{|\mathbf{P}i|} ||\mathbf{p}_i - \mathbf{p}_n||_2 \tag{3.9}$$

Potential energy and Color fluctuation

To define more properties for a point and its neighbors, location and color values can be combined to describe a point by the differences within the local area. One way to call the difference between two points is the amount of energy is needed to transform one to produce the other. It would cost more energy to move a point a longer distance, or change the color completely. The proposed property in [52] compared this to the gravitational potential energy relative to the Earth's surface defined as $E = mgh$, where m is the mass, h is the distance to the origin point and g reflects the gravitational field. They related m to the point color information, h to the distance related to point coordinate information, and g to a pooling weighing factor to distinguish isometrical distortion. This feature is referred to as Potential Energy Discrepancy (PED), or CV-PED (from computer-vision) when the color attribute is not available and only location information is used to calculate $E' = gh$.

In this calculation, a point is $\mathbf{p}_i = [x, y, z, R, G, B] = [\mathbf{p}_i^O, \mathbf{p}_i^I]$, in which $\mathbf{p}_i^O = [x, y, z]$ and $\mathbf{p}_i^I = [R, G, B]$, w_j are weights depending on the color space, here for RGB w_R :

$w_G : w_B = 1 : 2 : 1$.

$$m_{\mathbf{p}_i,\mathbf{p}_0} = \begin{cases} \sum_{j=1}^{3} w_j |(\mathbf{p}_i^I)_j - (\mathbf{p}_0^I)_j| + 1 & \text{if } \mathbf{p}_i^I, \mathbf{p}_0^I \neq \phi \\ 1 & \text{otherwise} \end{cases} \tag{3.10}$$

$$g_{\mathbf{p}_i,\mathbf{p}_0} = (1 + e^{-\frac{||\mathbf{p}_i^O - \mathbf{p}_0^O||_2^2}{\sigma^2}})^{-1} \tag{3.11}$$

$$h_{\mathbf{p}_i,\mathbf{p}_0} = ||\mathbf{p}_i^O - \mathbf{p}_0^O||_2^2 \tag{3.12}$$

A following paper [49] proposed comparing this energy property to a string with elastic energy, which they called Elastic Potential Energy Similarity (EPES).

$$W_{\mathbf{p}_i,\mathbf{p}_0} = \int_0^{||\mathbf{p}_i^O - \mathbf{p}_0^O||_2} -\mathcal{K}_{\mathbf{p}_i,\mathbf{p}_0} l \cdot dl = \frac{1}{2} \mathcal{K}_{\mathbf{p}_i,\mathbf{p}_0} ||\mathbf{p}_i^O - \mathbf{p}_0^O||_2^2 \tag{3.13}$$

$$\mathcal{K}_{\mathbf{p}_i,\mathbf{p}_0} = \left[\sum_{j=1}^{3} w_j |(\mathbf{p}_i^I)_j - (\mathbf{p}_0^I)_j| + 1 \right] \times ||\mathbf{p}_i^O - \mathbf{p}_0^O||_2^2 \tag{3.14}$$

where $\mathcal{K}$ is the spring elastic coefficient, and l is the elongation of the spring. The $\mathcal{K}$ variable is nicknamed EPES-CO as a potential useful feature with less mathematical processing steps as well.

Another version of this descriptor was used in a reduced reference metric [29], where it was referred to as the Color Fluctuation over Geometric Distance (CFGD). The CFGD value of a point is the mean value of the neighboring color intensity differences weight by distance of the current point.

$$CFGD_i = \frac{1}{|\mathbf{P}_i|} \sum_{n=1}^{|\mathbf{P}_i|} \frac{\sum_{j=1}^{3} w_j |(\mathbf{p}_i^I)_j - (\mathbf{p}_0^I)_j|}{||\mathbf{p}_i^O - \mathbf{p}_0^O||_2^2} \tag{3.15}$$

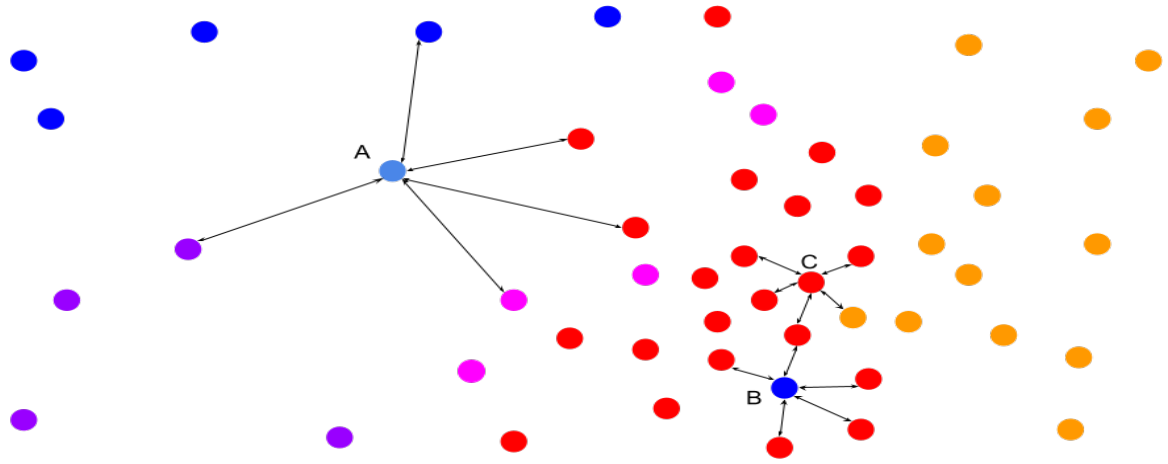


Figure 3.4: 2D energy features visualization with k=5, A has high energy (high euclidean and color distance), B has medium energy (high color distance), C has low energy (low distances)

Voxel patterns

Voxels are predefined groupings of the points of a point cloud based on bounding boxes of the same size in a discrete 3D grid. This method of defining neighborhoods introduces a way to not only compare neighbor points but also quickly find neighbor areas as the grid supports this easy indexing. The voxel properties include the grid index as location, and a color value which is calculated as the mean color value of all the points in the voxel.

In a full reference metric, the feature Local Luminance Pattern is proposed[12], which extracted the luminance value of each voxel and its direct neighbors, and binned these values in ranges with a bitwise label, summed up to produce a binary label that can be converted to a label value. A predecessor of this work from the same authors was Local Binary Patterns[11], where the bitwise labels were ordered based on the location of the neighbor and the bit is set to true if the luminance value is greater than the middle voxel. In this calculation, $C(v_i)$ returns the desired color value from voxel v_i (gray value in this case), N is the neighborhood size of the binary pattern, in figure 3.5 visualized as $N = 9$.

$$LBP_i = \sum_{n=0}^{N-1} (C(v_n) > C(v_i)) \cdot 2^n \tag{3.16}$$

A reduced reference metric uses another voxel feature, the Color block mean variance (CBMV)[29]. Here, the CBMV value of a voxel is calculated by the differences of point colors within a voxel, extracting the variance of each voxel.

$$CBMV_i = \sqrt{\frac{1}{|v_i|} \sum_{j=1}^{|v_i|} (C(x_{ij}) - \mu_i)^2} \tag{3.17}$$

$C(v_i)$ returns the desired color value from voxel v_i, $|v_i|$ is the amount of points in the voxel, and x_{ij} is the j-th point in the i-th voxel, and μ_i is the mean color value of the i-th voxel.

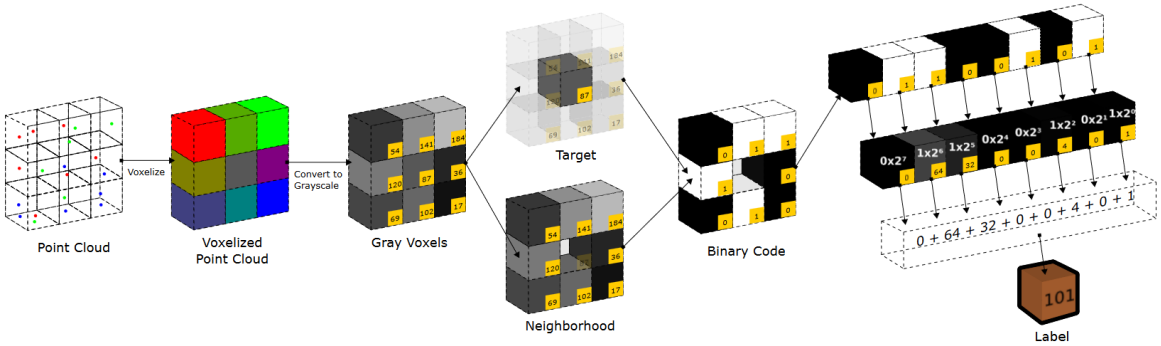


Figure 3.5: Local Binary Patterns[11] label calculation visiulisation

3.3. Statistical parameters

The features under investigation are all point or local features, which is how full-reference metrics can compare point to point distance to decide the quality difference. These local features have to be transformed to a global feature, since we can not compare point to point in no-reference metrics, and want a prediction of the quality value per point cloud. For each property, a high dimensional vector the length of the number of points or voxels in the point cloud is currently the information we have. We are looking for global behaviors to assess if these are indicators of quality. Smaller feature vectors need to be extracted to describe its behavior over the global point cloud. We can not assume an order in the data, as point cloud points do not have to be ordered in any way to represent a point cloud.

Summary statistics

Some basic statistics can already show much information about a dataset[44]. For features where lower or higher values directly correlate with the quality value, the Mean value is all you need. If this is the case, but some large outliers disrupt the calculation, the Median or the Mode might say more. If all values of a feature should be the same to have a high quality score, Standard deviation can describe this. Entropy and Sparsity can detect the information density, and Energy can report a measure of homogeneity.

Table 3.2: Definition of basic statistical properties

Feature	Definition
Mean	$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$
Median	$Md = \frac{x_{(N/2)}+x_{(N/2)+1}}{2}$
Mode	$Mo = x_i \vert P(x_i) > P(x_j), j \neq i$
Standard deviation	$\sigma = \sqrt{\sum_{i=1}^{N} p_i (x_i - \mu)^2}$
Entropy	$H = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log_b \mathrm{P}(x_i)$
Energy	$En = \sum \mathrm{P}(x_i)^2$
Sparsity	$Sp = \frac{\vert Z \vert}{N}, Z = \{x_i \vert P(x_i) \neq 0\}$

Generalized Gaussian distribution

The feature can be described to fit a probability distribution when shown in a histogram. If the distribution has a single peak around the mean value, we can try to fit a distribution function by estimating the parameters. The Generalized Gaussian distribution describes a continuous probability distribution using a shape parameter and a scale parameter around the mean expected value. If the shape parameter is 2 it is equal to a normal distribution, and if the shape parameter is 1 it is the Laplace distribution.

$$GGD(x;\gamma,\sigma^2,\mu) = \frac{\gamma}{2\beta\Gamma(1/\gamma)} exp(-(\frac{\mid x-\mu \mid}{\beta})^{\gamma})$$

$$\beta = \sigma\sqrt{\frac{\Gamma(1/\gamma)}{\Gamma(3/\gamma)}} \text{ and } \Gamma(\alpha) = \int_0^{\infty} t^{\alpha-1}e^{-t}\,dt$$

The parameter γ can be estimated using the generalized Gaussian ratio function[39]. This ratio uses the estimate for the modified mean of the absolute values $E^2[|X|] = (1/N)\sum \mid (x_i-\mu) \mid$ and the variance σ^2. The ration between these two ρ, is the same value with the correct γ in the following equation

$$r(\gamma) = \frac{\sigma_X^2}{E^2[|X|]} = \frac{\Gamma(1/\gamma)\dot{\Gamma}(3/\gamma)}{\Gamma^2(2/\gamma)} \tag{3.18}$$

The estimate for γ can be found by finding the solution to $\hat{\gamma} = r^{-1}(\rho)$ using a lookup table. This estimate with the value for β is extracted as information for the quality estimation.

The asymmetric version of the generalized Gaussian distribution uses the specific variance of the left and right side of the mean peak, to find the distribution with two functions:

$$AGGD(x;\nu,\sigma_l^2,\sigma_r^2,\mu) = \begin{cases} \frac{\nu}{(\beta_l+\beta_r)\Gamma(1/\nu)} exp(-(\frac{|x-\mu|}{\beta_l})^{\nu}) & ,x<\mu \\ \frac{\nu}{(\beta_l+\beta_r)\Gamma(1/\nu)} exp(-(\frac{|x-\mu|}{\beta_r})^{\nu}) & ,x\geq\mu \end{cases}$$

$$\beta_l = \sigma_l\sqrt{\frac{\Gamma(1/\nu)}{\Gamma(3/\nu)}} \text{ and } \beta_r = \sigma_r\sqrt{\frac{\Gamma(1/\nu)}{\Gamma(3/\nu)}}$$

In this case we calculate a supporting y value first and then calculate a ratio value to find the estimate for ν according to [26]

$$y = \frac{E[(x-\mu)^2|x<\mu]}{E[(x-\mu)^2|x\geq\mu]} \tag{3.19}$$

$$r(\nu) = \frac{\sigma_X^2}{E^2[|X|]} * \frac{(y^3+1)(y+1)}{(y^2+1)^2} = \frac{\Gamma(1/\nu)\dot{\Gamma}(3/\nu)}{\Gamma^2(2/\nu)} \tag{3.20}$$

The information extracted from this distribution is the estimated shape ν, the two variances σ_l^2 and σ_r^2 and the difference value of $\eta = (\beta_l-\beta_r)\frac{\Gamma(2/\nu)}{\Gamma(1/\nu)}$.

Gamma and Beta distribution

The Gamma and Beta distributions can fit a couple different shapes of the probability histogram, depending both on 2 parameters. The estimated parameters of the gamma distribution have already been shown to be useful descriptors in mesh quality estimation [1].

$$Gamma(x;a,b) = \frac{b^a x^{a-1} e^{-bx}}{\Gamma(a)} \text{ for } x > 0$$

The input values for this function have to be positive and starting at zero, so to extract the information the distribution data is shifted such that the lowest value is 0, $x = x - \min(x)$. From this distribution, we can estimate the parameters for the Gamma distribution with the following formulas:

$$a = (\frac{\mu}{\sigma})^2 \text{ and } b = \frac{\sigma^2}{\mu} \tag{3.21}$$

The beta distribution is a generalization of the uniform distribution, and is bounded to an $[0,1]$ interval. It can be interpreted as a ratio of gamma distributions. As it is bounded, the feature vector needs to be rescaled with $x = \frac{x - \min(x)}{\max(x) - \min(x)}$ in order to estimate its parameters.

$$Beta(x;c,d) = \frac{\Gamma(c+d)x^{c-1}(1-x)^{b-1}}{\Gamma(c)\Gamma(d)}$$

The c and d parameters of the beta distributions are estimated with the following formulas:

$$c = \mu^2(\frac{1-\mu}{\sigma^2}) - \mu \text{ and } d = c(\frac{1-\mu}{\mu}). \tag{3.22}$$

Thus, our global descriptor vector F comprises 17 values: μ, Md, Mo, σ, H, En, Sp, GGD (γ, β), AGGD ($\nu, \eta, \sigma_l^2, \sigma_r^2$), Gamma (a, b) and Beta(c, d).

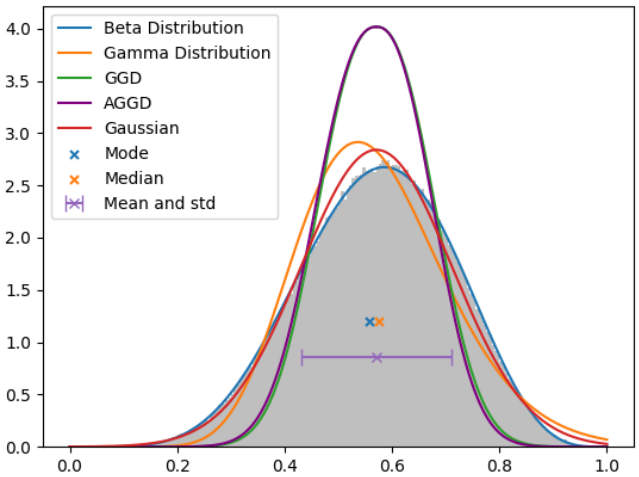


Figure 3.6: Example of calculated distributions

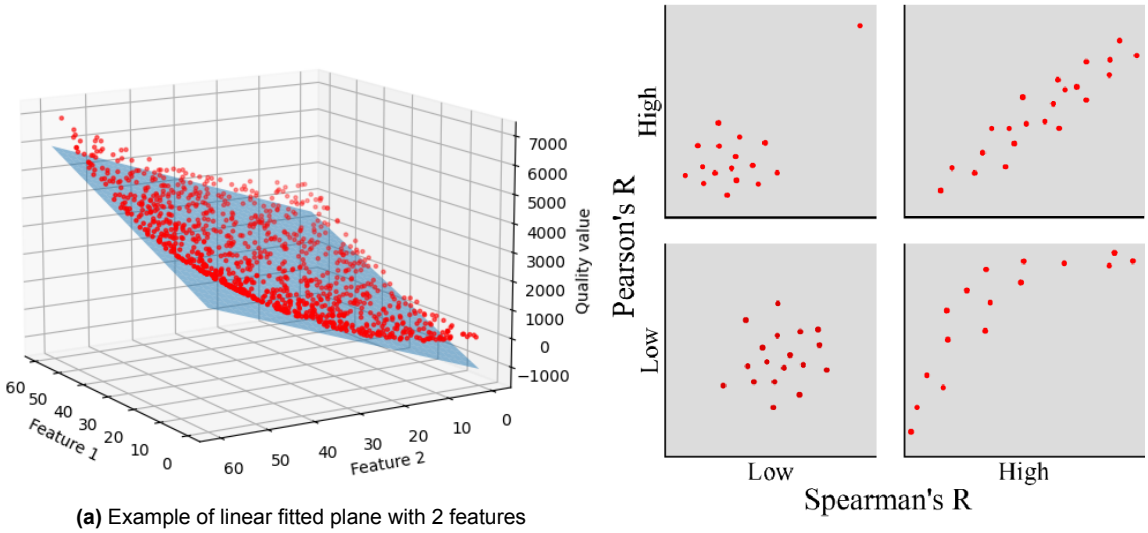
3.4. Selection based on feature relevance

For each point property described in section 3.2, we created a feature vector of global behaviors in section 3.3. Of the multiple global behaviors we extracted, we do not expect all of these features to have a correlation or any prediction power in a quality prediction model. A selection process can be done in multiple ways. Fitting the features to a quality value to create a prediction model and test its performance is a task that can be executed quickly, however it is very time-consuming when this has to be done for thousands of feature variables. This method is in the machine learning field referred to as a Wrapper Method. This approach is also depended on the chosen fitting or learning technique, while we first want to assess the robustness of the feature itself. This method can thus be used when comparing a couple of feature sets, but should not be the base selection method. To determine the potential of individual features, a simple correlation test between the feature value and the quality value for each point cloud shows us the direct prediction power, referred to as a Filter Method. Weinmann et al.[47] uses of several criteria to rank features based on relevance for the task of semantic interpretation of point clouds. We will be using the Pearson Correlation which tells us the similarity to a linear correlation, and the Spearman correlation which assesses monotonic relationships.

$$\text{Pearson: } \rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}$$

$$\text{Spearman: } r_s = \rho_{\text{Rank}(X),\text{Rank}(Y)} = \frac{\text{cov}(\text{Rank}(X),\text{Rank}(Y))}{\sigma_{\text{Rank}(X)}\sigma_{\text{Rank}(Y)}}$$

We do not expect features to individually correlate perfectly with quality, the goal of the selection is to find the minimal set of contributing features. The features also do not have to linearly correlate with the quality, as there are more ways to fit to a model. If two features that have the exact same behavior are found, using both of them instead of one will not increase the performance of a quality predicting model, finding correlation between the features is another step in the selection process where redundant information can be dropped.



(a) Example of linear fitted plane with 2 features

(b) Pearson and Spearman correlation visualization (Swan and Sandilands, 1995)

Figure 3.7: SVR example and correlation relationship

3.5. Fitting features to quality

To create a quality prediction model based on the selected global features, a support vector machine regressor (SVR) is modeled to fit the feature vector data to the desired quality value. SVR finds the mapping function $f(x_i)$ between the feature data input x_i to the quality score output f_i within an error margin ϵ. This means for a linear kernel function f to solve minimizing $1/2\|w\|^2$ with constraint $|y_i - \langle w_i, x_i \rangle - b| \leq \epsilon$, where x_i is our feature data per point cloud, y_i is its quality score, w and b are the learned weights and offset of the prediction model, as $\langle w_i, x_i \rangle - b$ is the prediction for a sample that should be within range ϵ of the target quality value y_i. With radial basis function kernel $f(x, w) = \sum_j w_j exp(-\frac{||x-x_j||^2}{2\sigma^2})$ we minimize $1/2\|w\|^2 + |y_i - f(x_i, w)| - \epsilon$.

4

Experimental setup

In this chapter, the methodology of the experiment is described. After an overview of the setup, we go over how each step is implemented: the preprocessing, local feature extraction, global feature vector extraction, and model design.

4.1. Pipeline

To investigate the performance of the proposed features, data from as many point clouds and distortions as possible are used to generate the feature data. This way, we can detect if some distributor values correlate with specific distortions or object types, as well as create a prediction model robust to as many distortions as possible.

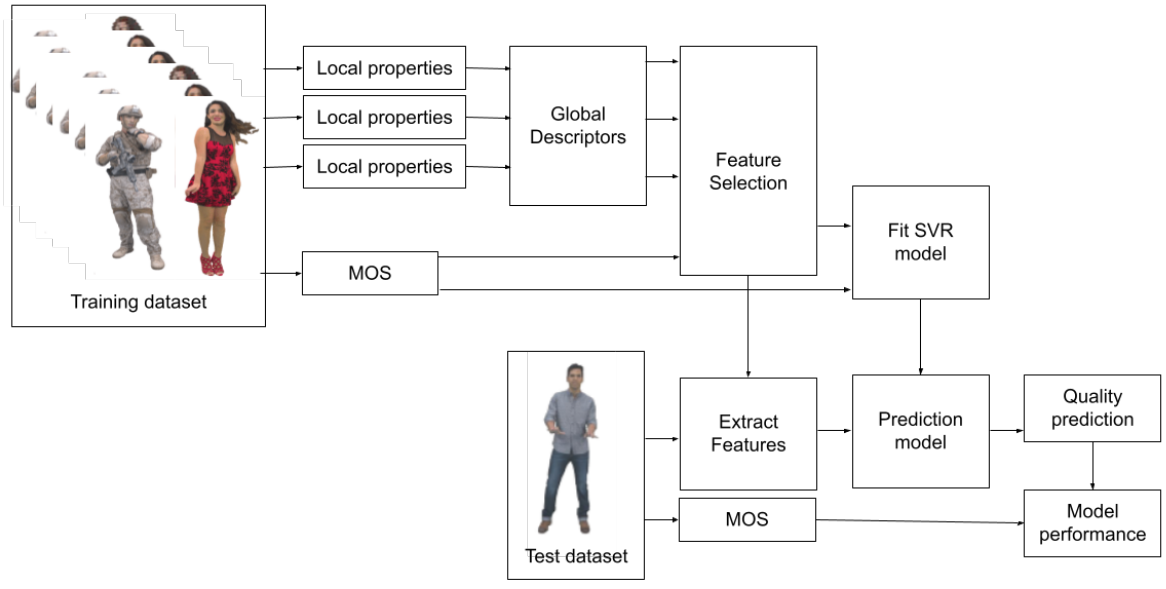


Figure 4.1: Experiment pipeline

The pipeline designed to produce and evaluate results can mostly be created in Python, where some algorithms of feature and statistic calculations are efficiently implemented in Python libraries. The input for the pipeline is point cloud files, and the output is both a prediction model and performance scores of the model. MatLab is used in a small part of the preprocessing of the data as Python does not have an efficient implementation for one of the algorithms.

Multiple packages for (point cloud) processing are available in Python.

- NumPy: for efficient array and matrix operations.
- PyTorch3D and Torch: optimized implementations of several functions for 3D data
- Open3D: offers a set of data structures and algorithms highly optimized for 3D data
- PyntCloud: Library specialized for point cloud processing
- SciPy stats: functions for calculating various statistics
- Panda: data structures and algorithms for efficient data manipulation and indexing
- MatPlotLib: visualizations of all kinds of data plots.

Preprocess data

For a database of point cloud files, the system gathers the name, point coordinates, and point colors. In Python, we use the data reader of Open3D to read the PLY files of the point clouds. As a pre-processing task, neighborhoods per point are calculated in an efficient implementation, as are normal vectors per point. The point information of a point cloud is not ordered in a set way, so the algorithm of finding K-nearest neighbors (KNN) is a time and space consuming task. The most efficiently implemented algorithm available that is suitable in this pipeline is in Matlab. For the experiment this is done separately from the main Python script and for each point cloud file an additional file with the KNN information is saved.

Extract feature data per point

With the available information for each point cloud, calculations for each local feature are implemented either combined or individually, depending on the similarity of the algorithm of the different features, like the features based on Principal Component Analysis are extracted together.

Some algorithms are already implemented into a Python library or codebases from previous papers, some need to be written from scratch, for all some adjustment is needed to fit it in this pipeline.

The features are calculated in series, going over each point every time. Another strategy would be to calculate all features for 1 point and calculate the points in series, but since most implementations are most efficient to calculate a feature for all data points at the same time, the first option is implemented.

The PCA features are calculated using mainly NumPy with a little support of the point cloud data structure from the PyntCloud library. The potential energy features make use of NumPy and PyTorch methods. Translating between color spaces are simple to implement using NumPy, but are also already available in the scikit-image color module.

For each feature, multiple implementations are used to generate data, as to find the best version of implementation and parameters.

Extract distribution of point feature vector

When the feature values of all points are extracted, the statistical distributors are immediately calculated. For each feature, only this data is stored, as this is the input data for fitting the regression model, and answering the research question. Like the features, most algorithms are already implemented, and needed a little adjustment to fit in this pipeline.

The Python libraries NumPy and SciPy Stats provide functions for the Summary statistics. For the estimation of the parameters of GGD, AGGD, Gamma and Beta distributions, previous work[57] provides us with a base of the implementation with numerical calculations for (A)GGD parameter estimations, as well as the Gamma parameter estimation formula. This was extended with the Beta parameter estimation, adjusted to fit the new datatypes, and improved for correctness.

Model predictor to fit MOS

The first part of creating a final model is creating a high potential feature vector. To find the optimal version of a global description for each feature, different kinds of analysis are used to make a selection. To find individual relevance of a feature, correlation between the ground truth quality score and the feature values are reviewed. To eliminate redundant information, correlation between features is used to make a smaller selection. To compare or review feature sets, a support vector machine regressor(SVR) is modeled to fit the feature vector data to the desired quality value. We follow the framework defined in a the work of Zhang et al. [57] with slight adjustments; to split the data in training and testing sets, of all point cloud objects, 2 are selected as the test set (1 in Zhang et al.), and the rest are training data. All combinations of this split are scored and averaged to a final performance score of the model. In most previous work, the performance was only reported within a dataset, in this experiment cross dataset results are also reported.

SVR To analyze the performance of different feature variants/sets, we fit an SVR to subsets of data (like a single dataset). From these models, we can analyze multiple ways which version of the feature variant performs best. For this part of the experiment, no optimization of the prediction model is needed as we are only trying to find the best correlating version of the features, and we will after this use these findings in an experiment to create an optimized model.

To create the optimal quality predictor, an SVR with a set of handcrafted features will be optimized. This includes both parameter tuning and feature selection. The feature selection is based on the findings of the first part of the experiment, and will be optimized on both performance and computational efficiency. The parameters that can be adjusted for an SVR include; which kernel to be used in the algorithm, we limit the choices to linear, polynomial, and radial basis kernel function; the kernel coefficient Gamma or the degree parameter; a regularization parameter C; and the tolerated error Epsilon.

Dataset The data used in the experiments is summed up in Table 4.1. When analyzing the performance per distortion type, we can combine some distortions that are generated using different version of the algorithm, but show the same behavior in the features we are experimenting with. In the end we hope to find one prediction model, so the results will be combined anyhow. The G-PCC variants are already categorized under this term, so the data labeled with distortions ”Octree-Lifting”, ”Octree-RAHT”, ”TriSoup-Lifting”, ”TriSoup-RAHT”, and ”GPCC” are candidates to report combined. The performance on distortion types that have combinations with other types in the dataset of SJTU, can be pooled together per type with overlapping subsets, so performance of the Downscaling distortion can be based on the data of DS, D+C and D+G labeled combined data. Some datasets have used a different range of ground truth quality scores, in experiments with combined datasets these scores are all standardized to a scale of 1-10.

Object	Dataset	Available distortions
Statue	SJTU	CN, DS, OT, GGN
Shiva	SJTU	CN, DS, OT, GGN
Hhi	SJTU	CN, DS, OT, GGN
Unicorn	SJTU	CN, DS, OT, GGN
Redandblack	SJTU, ICIP2020	CN, DS, OT, GGN, G-PCC, V-PCC
Romanoillamp	SJTU, M-PCCD,	CN, DS, OT, GGN, G-PCC, V-PCC
Loot	SJTU, M-PCCD, ICIP2020, IRCP	CN, DS, OT, GGN, G-PCC, V-PCC
Soldier	SJTU, M-PCCD, ICIP2020	CN, DS, OT, GGN, G-PCC, V-PCC
Longdress	SJTU, M-PCCD, ICIP2020, IRCP	CN, DS, OT, GGN, G-PCC, V-PCC
Ricardo10	ICIP2020	G-PCC, V-PCC
Sarah9	ICIP2020	G-PCC, V-PCC
Facade	IRCP	G-PCC, V-PCC
Frog	IRCP	G-PCC, V-PCC
House	IRCP	G-PCC, V-PCC
Mask	IRCP	G-PCC, V-PCC
Amphoriskos	M-PCCD	G-PCC, V-PCC
Biplane	M-PCCD	G-PCC, V-PCC
Head	M-PCCD	G-PCC, V-PCC
The20smaria	M-PCCD	G-PCC, V-PCC

Table 4.1: OT: Octree-based compression, CN: Color Noise, DS: Downscaling, GGN: Geometry Gaussian noise, V-PCC : MPEG Video-based Point Cloud Compression, G-PCC: MPEG Geometry-based Point Cloud Compression

Performance analysis We analyze the performance of each individual feature, as well as the combination of all features, by scoring the prediction against the ”ground truth” values of the MOS reported in the dataset. The predicted values of the test set of each run is scored on 3 different correlation ratings: the Spearman Rank Correlation Coefficient (SRCC), Pearson Linear Correlation Coefficient (PLCC) and Root Mean Square Error (RMSE) to account for monotonicity, linearity and accuracy, respectively. This is according to ITU-T recommendation P.1401 [20], using these established methods, we can compare performance of the models with previous work.

5

Results

With the framework described in Chapter 4, all features with their variable parameters are scored on the relevance of fitting a prediction model for quality scores. In this chapter, each feature is analyzed to explain the results from the experiment, and we make a selection of the highest potential feature set in section 5.1. Then we look further into which of these features work in which cases by splitting the tests per distortion type in section 5.2, and we look for more feature sets that perform well in these split experiments. These two sections provide us with two feature sets, one is a combination of individual high potential features, and the second set contains feature combinations that together perform well for specific cases.

After this analysis, we create the final quality prediction models for multiple experiment setups to test and compare the performance as a no-reference quality metric in section 5.3.

5.1. Feature Analysis

For each feature, we try to find the optimal implementation and parameter setting to find the useful relations between the ground truth quality assessment of distorted point clouds and the feature values, to most accurately evaluate the potential for this feature in the no-reference quality assessment context. To get the most robust and efficient prediction model, we want to reduce the amount of features and remove features that are not contributing to the model. Our goal is to significantly reduce the current feature set of 500+ features by handcrafted feature selection.

Each subsection contains a selection of the best and unique global features, most subsections start with a different analysis to find the best implementation variation of the specific features. The statistics we calculated for each feature is the set of μ, σ, H, GGD (γ, β), AGGD ($\nu, \eta, \sigma_l^2, \sigma_r^2$), Gamma (a, b), Beta(c, d), Md, Mo, En, and Sp, as described in Section 3.3. This set is in some figures written out as {mean, std, entropy, ggd1, ggd2, aggd1, aggd2, aggd3, aggd4, gamma1, gamma2, beta1, beta2, median, mode, energy, sparsity}.

PCA features

The geometric features under review have already been proven to be useful in predicting quality scores in other QA methods, and the gathered data seems to align with these findings. To find the best set of PCA features to use in a prediction model, we'll first decide the size of the neighborhood from which the eigenvalues of the covariance matrix are used to calculate the features, then find which statistical behavior best correlates with the quality value.

Neighborhood definition As hypothesized, the result of the PCA calculations differ per neighborhood size k, we are looking for a robust yet informative k for each feature.

All PCA features have been calculated with $k \in \{5, 8, 10, 12, 20, 30\}$, and their global statistics have been used to fit a linear SVR with the quality scores of the SJTU dataset to compare performance per k in Figure 5.1.

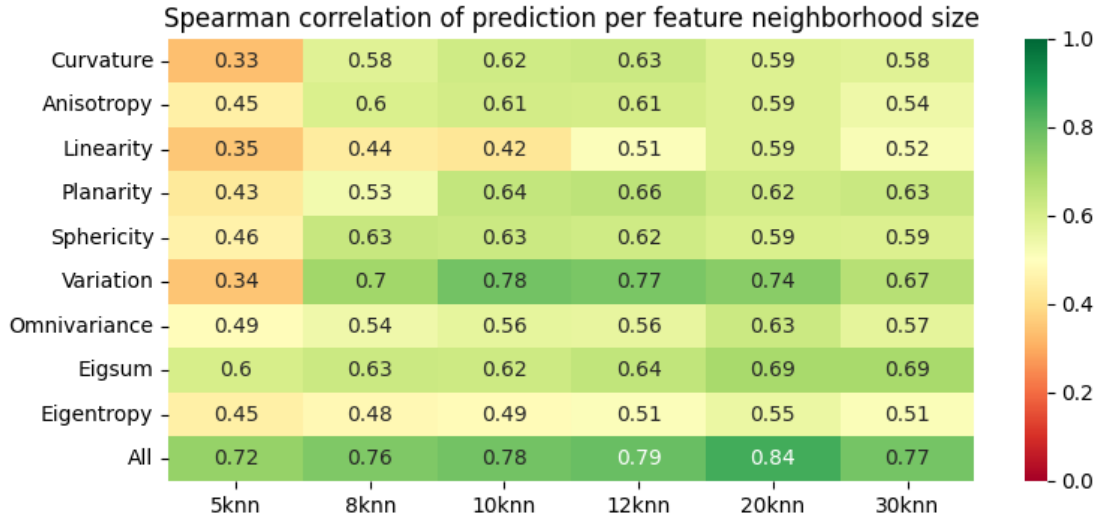


Figure 5.1: Heatmap of the Spearman correlation coefficient between the predicted quality and the MOS values. Each cell is a model with the feature vector containing all global statistics of the feature on the y-axis calculated with a different neighborhood size in the x-axis.
Values are only to be compared within this table, as model parameters differ per experiment run.

The trend to be observed in the combined feature data model is larger neighborhood size leads to a better performance, however with little increase. Per individual feature, we see some small and some big performance differences. The Eigenvalue Sum and Eigenvalue Entropy are barely influenced by the value of k, Linearity and Omnivariance need at least 20 neighbors to perform, and the rest need a k of 10-20 to have good results without clear outperformers. Using 30 neighbors to calculate the features does show for some features an information loss, some perform negligible better. In terms of computation optimization of these features, ideally a single PCA computation of eigenvalues is needed to calculate all these features, and thus a single

k is to be chosen as most optimal. For the following experiments, $k = 20$ for the PCA features.

Global behaviors For each feature, we calculate the Spearman and Pearson correlation with the MOS value to find linear or monotonic relationships between these. Both negative and positive correlation coefficients are beneficial to use in a predicting model, we are looking for the features with scores close to -1 or 1, not 0.

In Figure 5.2a, the Spearman correlation coefficient of the eigenvalues without additional processing are also reported, because adding these did not cost any extra processing in the experiment (already calculated). These are not included in the experiment, as little grounding was found in the literature study to examine these as features. The correlation coefficient of eigenvalues are comparable with many PCA features so for now we can exclude these again from the experiment, but in future work the relevance of these features should be examined as they show potential.

We definitely do not see many individual features that properly correlate with the MOS value in these heatmaps. This is to be expected and not a problem as we will be using multiple in a final prediction model. We select features that have a Spearman correlation coefficient of at least .35 or Pearson above 0.3 for further analysis.

If we test these features to fit a linear prediction model with the 4 available datasets (Different setup from Figure 5.1), the performance of these 26 features is already averaging PLCC=0.6162 and SRCC=0.6098. Some global features describe the same behaviors but are calculated slightly different, if these end up with the same values it is not contributing to have them both in a prediction model. A correlation between features is our next feature selection method, the Spearman correlation is selected as the most appropriate basis for this task.

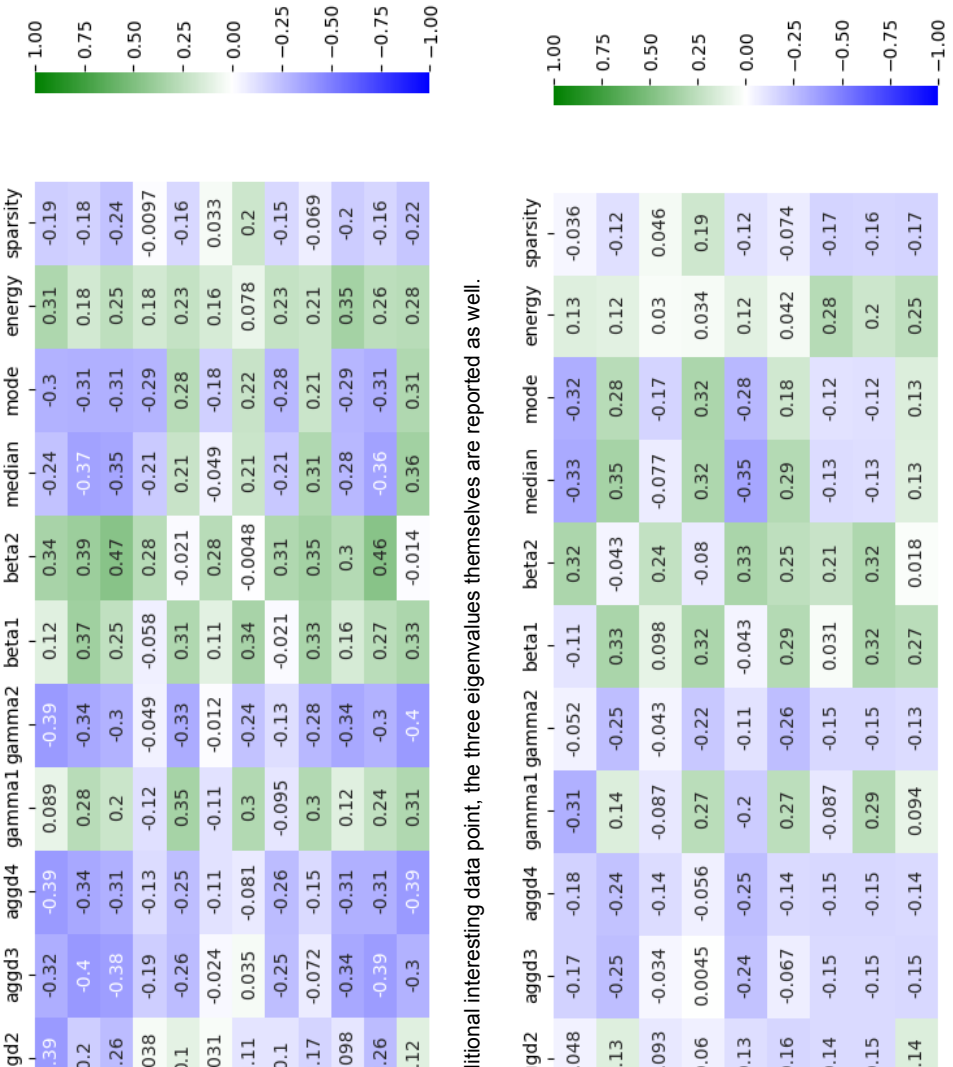
The features with correlation coefficient 1 or -1 provide the same information to a prediction model, so we can eliminate some features in these cases. The similar feature groups are

- Curvature$\{\mu, Md\}$, **Anisotropy**$\{\boldsymbol{\mu}, Md\}$, Sphericity$\{\mu, Md\}$
- Eigsum$\{\mu, \sigma_l^2, Md\}$, **Eigentropy**$\{\mu, Md, \sigma_r^2, \boldsymbol{b}\}$

Most of the similar features should not look surprising. When little outliers have influence over the calculated mean value, the median and the mean should have similar results. We also see that some PCA features might describe the same behavior for point cloud data, which leads to the same distribution values and correlating scores. If a point cloud has high Sphericity, the mean Curvature value is probably high as well. One of the features with the best correlation coefficient with the MOS value is selected from these correlating sets (in bold).

After this selection, we test another linear prediction model based like before this selection, and the performance is increased slightly, PLCC=0.6228, SRCC= 0.6144, showing the effectiveness of removing redundant information from the feature pool.

Figure 5.2: Heatmap of the correlation coefficient of each global descriptor (x-axis) per PCA feature (y-axis) with the MOS value. Data from all available datasets is used.
Values are only to be compared within this table, as model parameters differ per experiment run.



(a) Spearman correlation coefficient. For an additional interesting data point, the three eigenvalues themselves are reported as well.

(b) Pearson correlation coefficient.

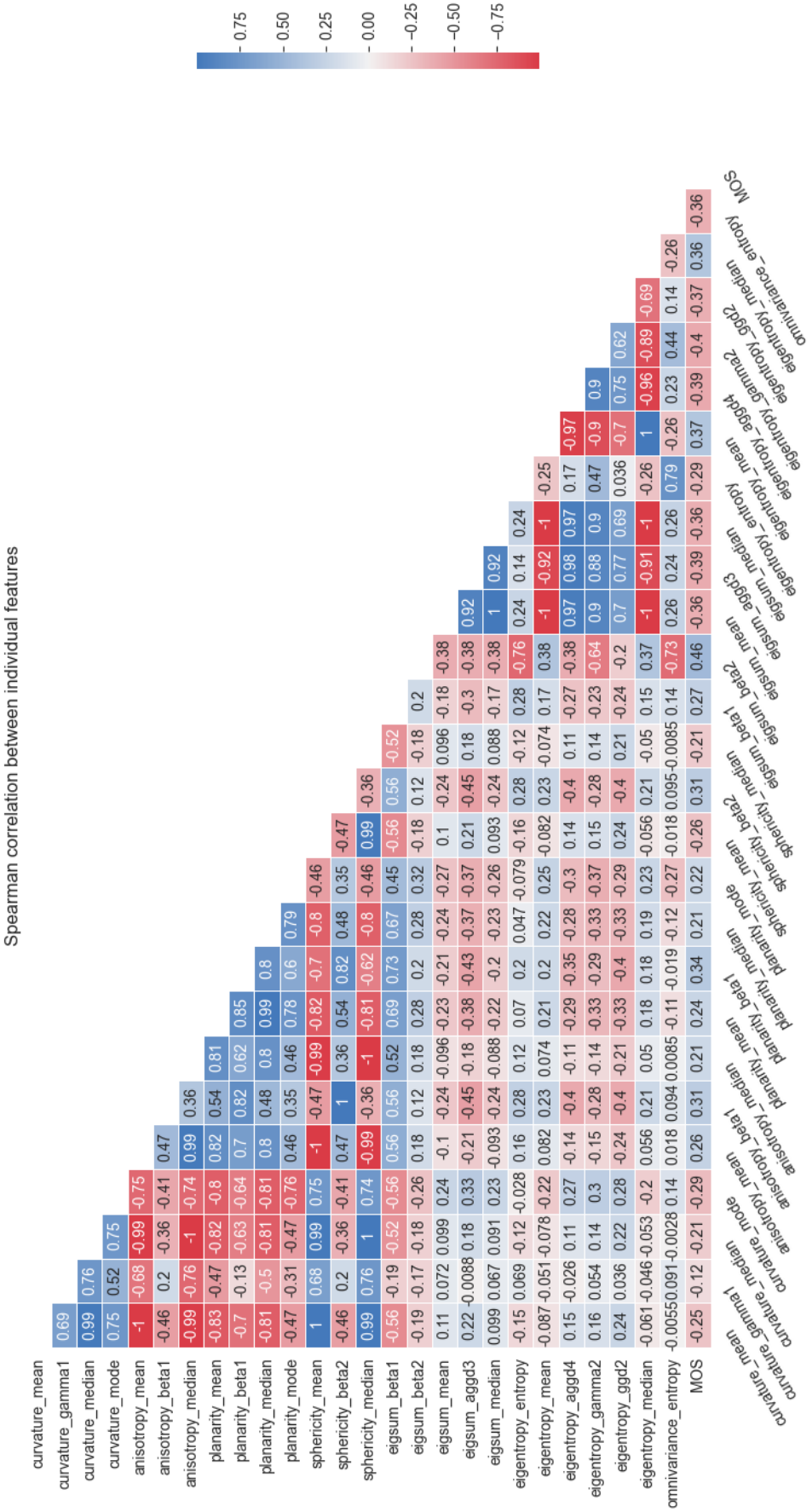


Figure 5.3: Spearman correlation coefficient between the high potential feature selection.

Feature selection From reviewing the feature relevance of the PCA features, the result of the feature selection is the following set: {Curvature$\{a, Mo\}$, Planarity$\{\mu, Md, Mo, c\}$, Eigentropy$\{H, b, c\}$, Omnivariance$\{H\}$, Eigsum$\{c, d\}$, Anisotropy$\{\mu\}$, Sphericity$\{d\}$}

The contribution of these features in a linear prediction model can be shown by the (absolute) weight assigned after data standardization. We visualize this for each dataset to get a feel of the differences when fitting the same feature selection to different training data. The following normalized contributing weights are found for all datasets:

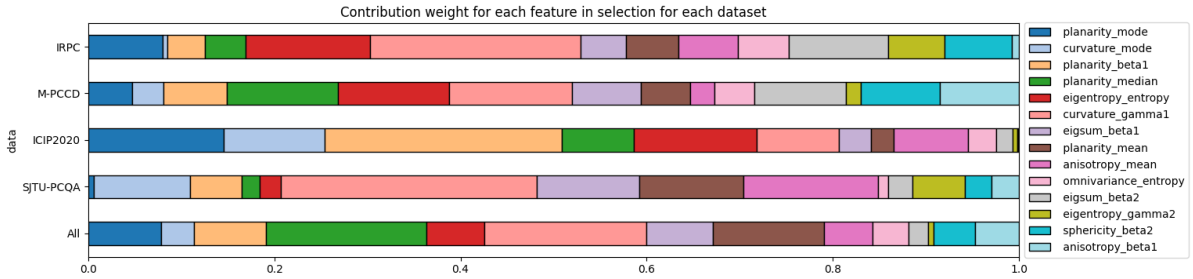


Figure 5.4: Contribution weights of linear fitted model with features from selection per dataset

We indeed observe, different features contribute more in different datasets. The datasets differ in distortion types and their ratios. There is no feature in this set that contributes most to every model, no feature in this set has a universal linear correlation with all types of distortion tested.

Color Attributes

Of all the calculated color attributes, we expected the CIELAB color space to best correlate with the MOS quality, since this space best describes human visual color range. Let's start with comparing models using all attributes from the color spaces with a linear model on all data:

Color space	SRCC	PLCC	RMSE
all color attributes	0.2621	0.2576	5.1202
LAB	**0.3263**	**0.3485**	**3.0695**
RGB	-0.0264	-0.0398	3.9943
XYZ	0.3187	0.3434	3.3481
HCL	0.2694	0.2656	3.7471

Table 5.1: Performance of linear fitted models with color features, reported are Spearman Rank Correlation Coefficient (SRCC), Pearson Linear Correlation Coefficient (PLCC) and Root Mean Square Error (RMSE)
Values are only to be compared within this table, as model parameters differ per experiment run.

And indeed it does show the CIELAB color space as best performance, closely followed by XYZ space. The overall performance is not very high, which is explained by the dataset composition and the nature of this feature, color values. When analyzing the individual features, we need to adjust the data to be able to make useful observations.

Global behaviors For each feature, we calculate the correlation coefficient with the MOS value to find relationships between these. Like in the PCA feature analysis, both negative and positive correlation coefficients are beneficial to use in a predicting model, and we are looking for the features with scores close to -1 or 1, not 0.

In Figure 5.5 we can immediately see that this feature has little correlation with the quality score over the whole dataset as many values are close to 0. This is because these pure color features only detect color distortions, and most distortions in the dataset are not adjusting the color values but rather the location information of the points. This will mean that in a quality assessment model based on only color features, tested on a dataset of which the majority is non-color distorted point clouds, the quality predictions will probably be random.

If we only look at the point cloud subset with color noise, we will be able to select features that will be useful in a setting to detect color quality. When training and testing a linear model using all color attributes on point cloud with only color noise, the performance jumps up from the values in table 5.1 to SRCC=0.5302 and PLCC=0.6073.

Given this knowledge we restart the individual feature analysis and look again at the correlation between features and MOS values.

In Figure5.6, it is a lot clearer that there does exist a correlation with the calculated color features and the quality score of the color noise distorted point clouds. The a* and b* attributes of the CIELAB space and the Chroma and Hue attributes of

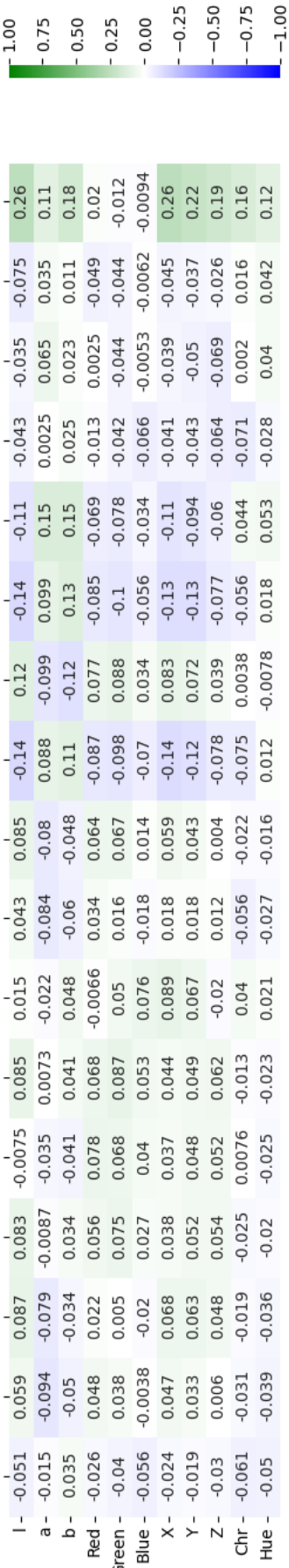


Figure 5.5: Heatmap of the Spearman correlation coefficient of each global descriptor (x-axis) per color attribute (y-axis) with the MOS value. Data from all available datasets is used. *Values are only to be compared within this table, as model parameters differ per experiment run.*

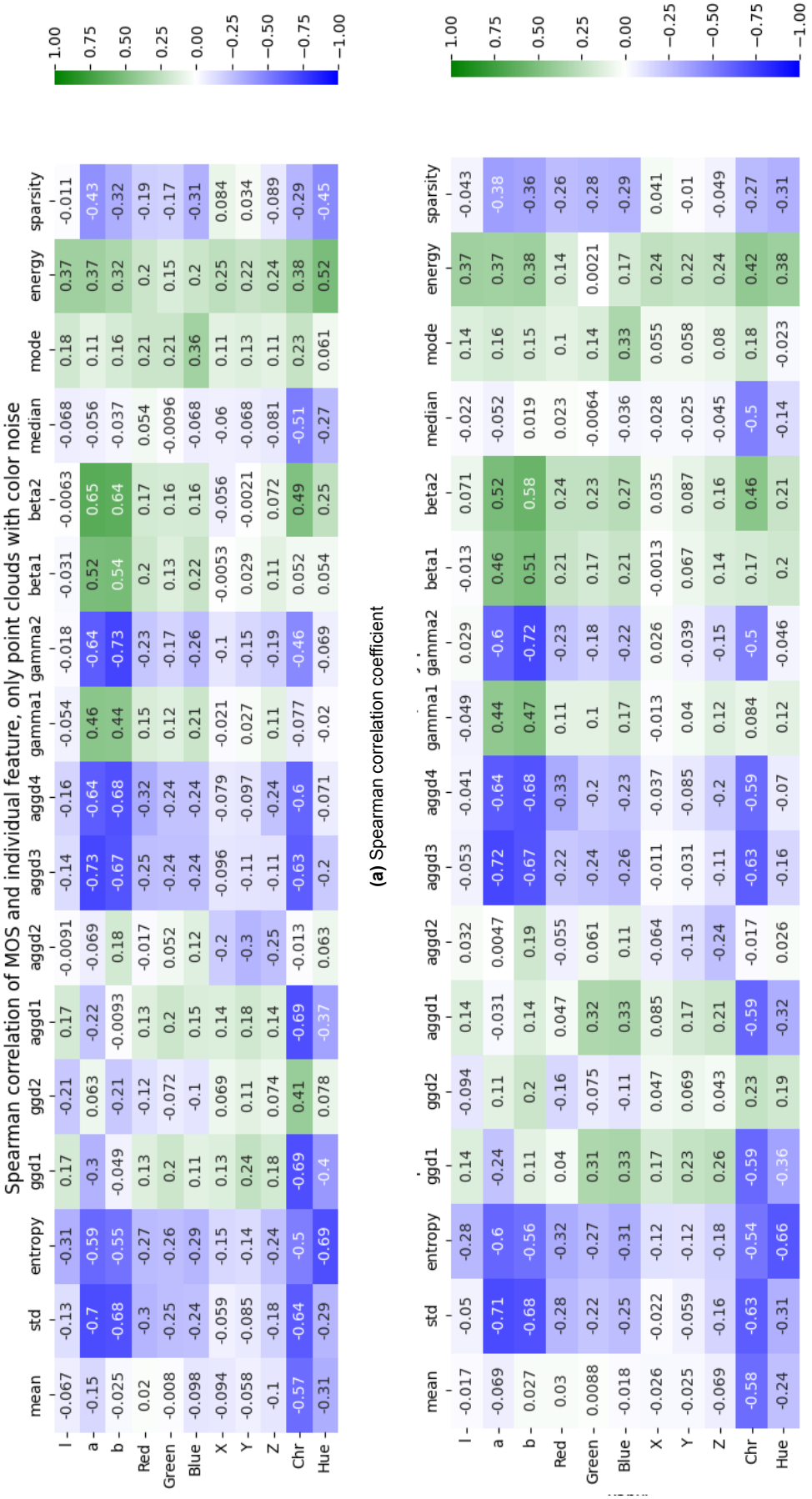


Figure 5.6: Heatmap of the correlation coefficient of each global descriptor (x-axis) per color attribute (y-axis) with the MOS value. Data from a color focused subset of the available datasets is used.
Values are only to be compared within this table, as model parameters differ per experiment run.

the HCL space have many correlation coefficients above 0.5. It is interesting to see the Luminance attribute fall behind in this analysis. Luminance has been one of the main features analyzed in this research field (PCQA) as important color attribute. The choice in data selection in this experiment is likely the main cause in this difference.

The performance of a linear model using a selection of 24 of these features is improved to PLCC=0.721 SRCC=0.7251 for only color noise distortion, and worsened to PLCC= 0.1859 SRCC=0.1495 over the complete dataset.

We continue again with further feature elimination using correlation-based analysis. In Figure5.7, we see these color features all are more alike than the PCA features, almost all values are above 0.5.

It is not surprising to see such high correlation, as color values are highly dependent from each other to produce the correct visual color together. The most correlating features of which some can be removed are:

- a*$\{\sigma_r^2, b, c, d\}$, **a***$\{\boldsymbol{\sigma, \sigma_l^2}\}$, Chroma$\{\sigma, \sigma_l^2, \sigma_r^2\}$
- **b***$\{\boldsymbol{\sigma_l^2}\}$, b*$\{\sigma_r^2, \sigma\}$
- **b***$\{\boldsymbol{d}\}$, b*$\{c\}$
- Hue$\{En\}$, **Hue**$\{\boldsymbol{H}\}$
- Chroma$\{\gamma\}$, **Chroma**$\{\boldsymbol{\nu}\}$
- Chroma$\{\mu\}$, **Chroma**$\{\boldsymbol{Md}\}$

The features with the best correlation coefficient with the MOS value is selected from these correlating sets (in bold).

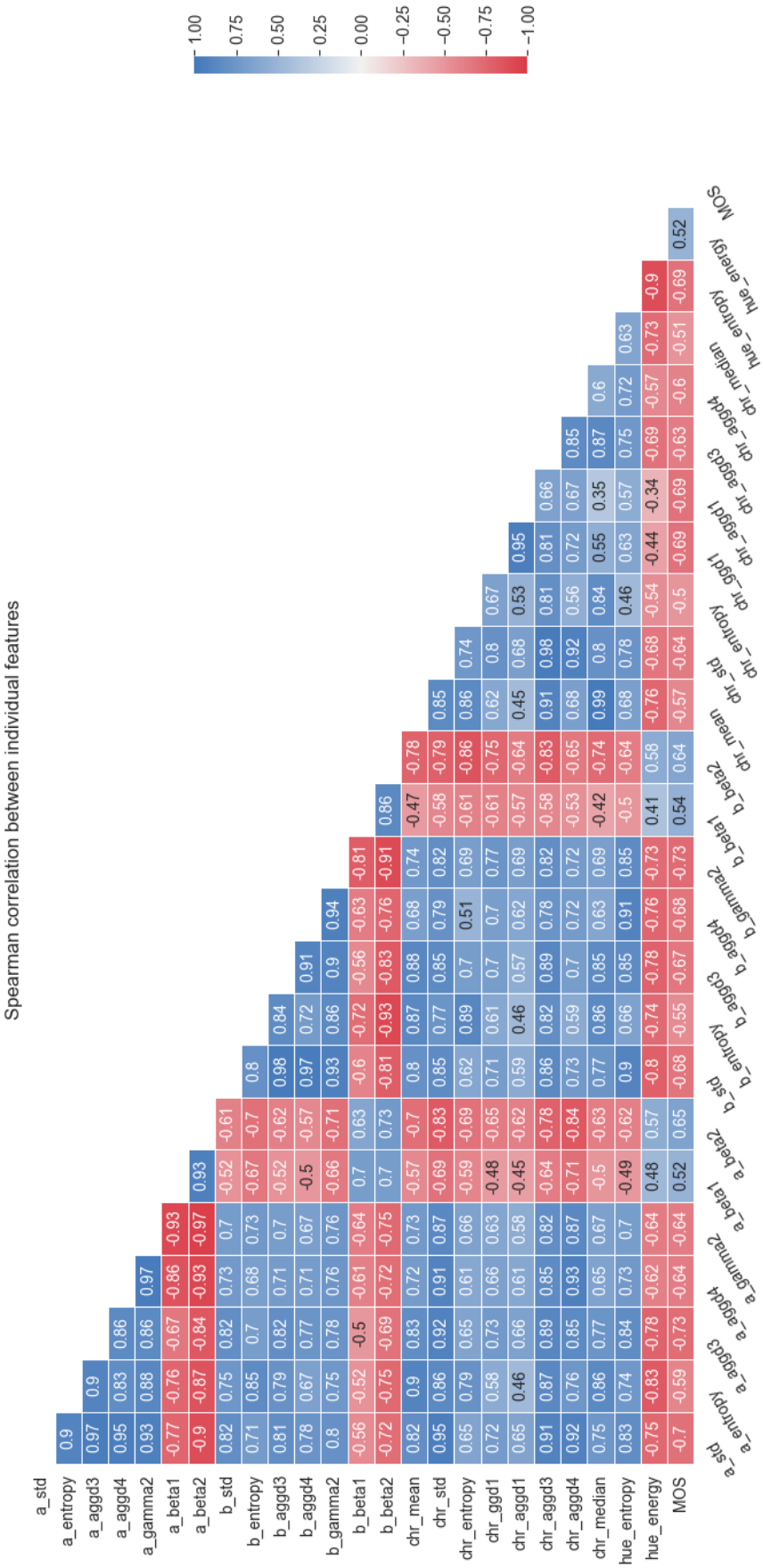


Figure 5.7: Spearman correlation between the high potential feature selection. Data from a color focused subset of the available datasets is used.

Feature selection From reviewing the feature relevance of the color features, the result of the feature selection is the following set {a*$\{\sigma, H, \sigma_l^2\}$, b*$\{H, \sigma_l^2, b, d\}$, Chroma$\{\sigma, H, \nu, Md\}$, Hue$\{H\}$} The performance of a linear model based on these features is in the case of only color noise point clouds improved to PLCC=0.7788 and SRCC=0.7772, with the overall performance on all data still low at PLCC=0.2290, SRCC=0.1850.

When analyzing the contributions of each color feature in models trained per dataset, we this time add the subset of color noise as one of the categories.

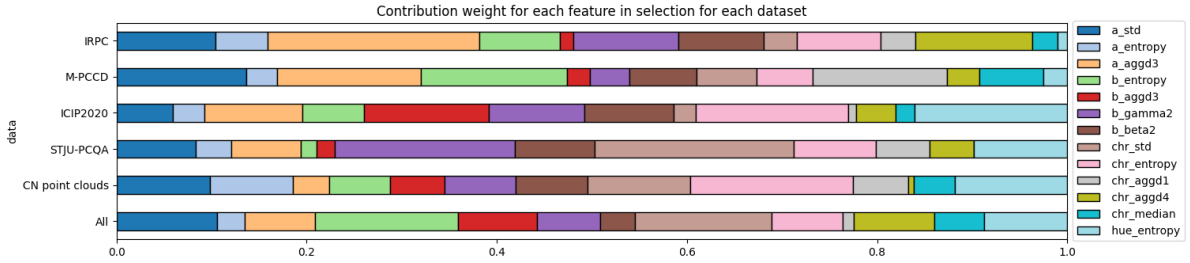


Figure 5.8: Contribution weights of linear fitted model with color features from selection per dataset

The selection seems to be uniquely fitted to every distortion subset (dataset) again, no feature that has a universal linear correlation with all types of distortion is found.

Potential energy

To summarize this subsection in section 3.2, the investigated methods of calculating potential energy values are: PED = mgh, CV-PED = gh, EPES-CO = mh, EPES = $1/2(mh)(h^2)$, CFGD = m/h, where m is the color distance, h is the geometric distance, and g is the pooling weighing factor.

In the following table, we see all these methods can have prediction power in different cases. The performance per model is reported per distortion type, as to introduce the differences of energy detected per calculation method. The performance on the distortions are all very similar for the first 3, and EPES and CFGD behave differently.

	OT	CN	DS	D+C	D+G	GGN	C+G	Octree	TriSoup	V-PCC
PED	0.433	-0.085	0.661	0.687	0.765	0.696	0.722	0.661	0.345	0.312
CV-PED	**0.470**	0.037	**0.672**	**0.713**	0.766	**0.815**	**0.805**	0.665	0.458	0.427
EPES-CO	0.432	-0.066	0.668	0.694	**0.771**	0.700	0.727	0.677	0.291	0.313
EPES	0.349	-0.109	0.445	0.457	0.412	0.318	0.405	0.714	0.143	-0.561
CFGD	-0.003	**0.316**	0.155	0.312	0.369	0.285	0.110	**0.834**	**0.741**	**0.798**

Table 5.2: PLCC performance of prediction models fitted using all global statistics of the feature on the y-axis. On the x-axis, the test results are split in the distortion type, of which the PLCC performance numbers are reported. The dataset used are SJTU and ICIP separately, the data is fit to the first 7 in a separate experiment run than the last 3.
Values are only to be compared within this table, as model parameters differ per experiment run.

The differences in performance between PED, EPES-CO and CV-PED is minimal, and the additional calculation of EPES seems to remove (linear) correlation and performs worse. Including all factors in the energy formula (PED) does not seem to contribute positively. Weirdly enough, the color noise has not been detected using these calculations, the color difference value might not be correctly scaled against the distance value, causing it to barely contribute to the calculations.
The color fluctuation feature performs very differently, which is expected since it does describe a different behavior. With PED, far neighbors have more influence, while with CFGD, closer points do. We can see the CFGD feature performs very well in distortions that lead to color blurring (G-PCC, V-PCC). .

Neighborhood definition In the previous table, a neighborhood of 20 nearest neighbors was used to calculate the energy values. The calculation of the energy implementation is resulting in the total energy of the neighborhood, not the average per neighbor. This means bigger neighborhoods result in a higher mean value of this feature. As long as the feature is always calculated with the same neighborhood size, this should not matter for the prediction model. For CV-PED, the performance of each neighborhood size is displayed in Figure5.9. The performance of $k = 20$ seems to be performing best, like the PCA features.

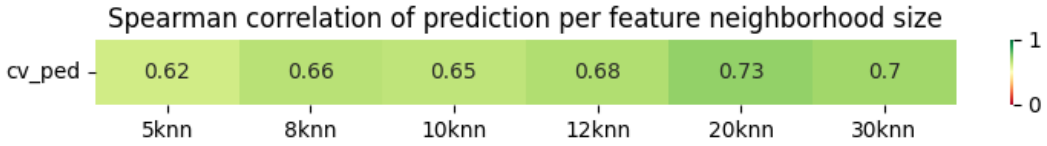
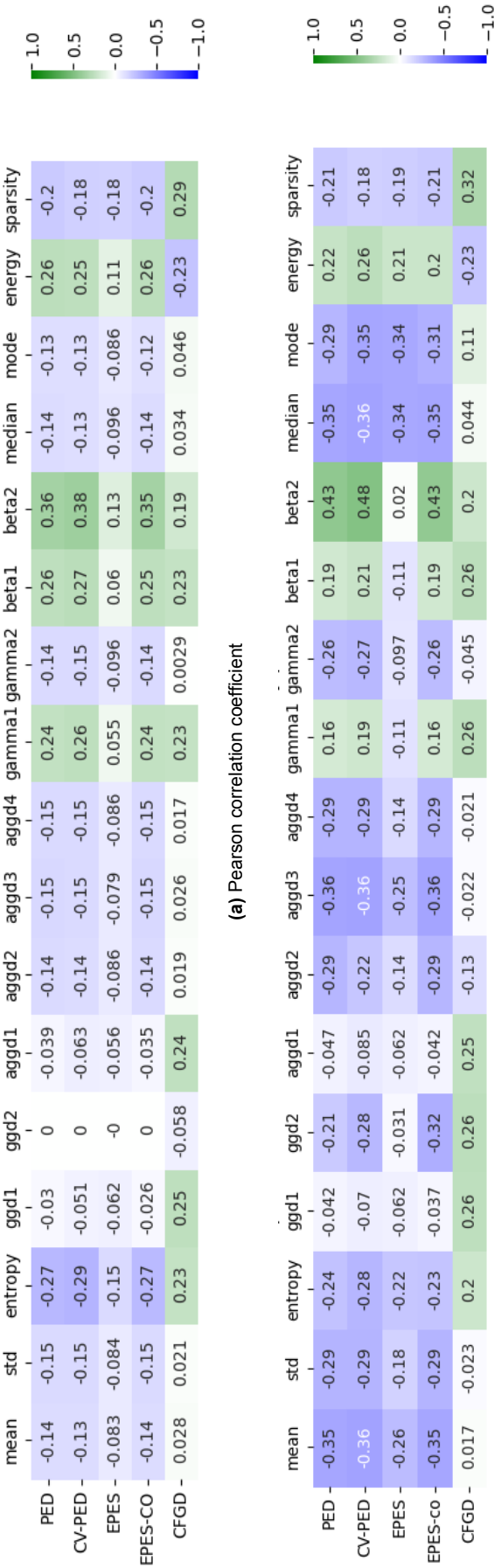


Figure 5.9: Heatmap of the Spearman correlation coefficient between the predicted quality and the MOS values. Each cell is a model with the feature vector containing all global statistics of the feature on the y-axis calculated with a different neighborhood size in the x-axis.
Values are only to be compared within this table, as model parameters differ per experiment run.

Global behaviors The values of the energy features are not limited or scaled to a smaller range in its formula. For point clouds where the density of the points is lower, so the points are farther away from each other, the PED and EPES values are high, and the opposite is true for the CFGD values. This is part of the behavior we are trying to detect with this feature, so scaling with the density would undermine the information given by this feature. The current feature formula might not translate to a comparable mean statistic of the distribution, but some distribution descriptors are not dependent on the range of the data values.

From Figure 5.10, we again follow the now known steps of selecting the features with the highest correlation with MOS, and perform another elimination round based on the correlation between the features themselves. This time this process included an extra step of handcrafting the set, as the correlation between the features did not give a clear indication which to remove due to the unique range of the data values. By experimenting with a Wrapper method approach of recursive feature elimination, the most redundant features could be removed without lowering the performance scores.



(a) Pearson correlation coefficient

(b) Spearman correlation coefficient

Figure 5.10: Heatmap of the correlation coefficient of each global descriptor (x-axis) per energy attribute (y-axis) with the MOS value. Data from all available datasets is used.
Values are only to be compared within this table, as model parameters differ per experiment run.

Feature selection After correlation-based and recursive-based elimination we are left with the following feature set: {CV-PED$\{\mu, H, d\}$, EPES$\{Md\}$, EPES-CO$\{\mu\}$, CFGD$\{H, \nu, a, c, d, Sp\}$}.

Of which the contribution weight of the features are assigned per dataset as follows:

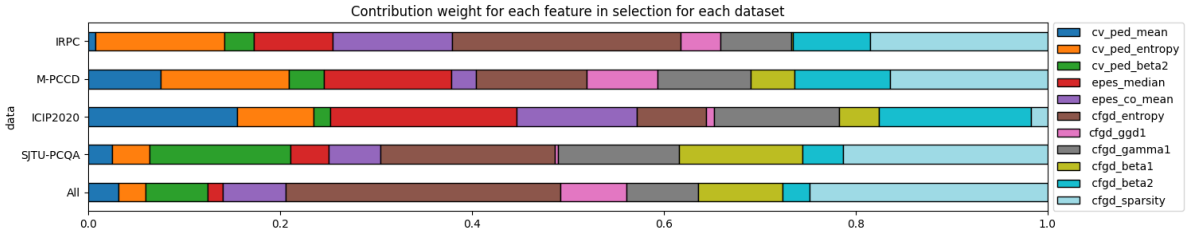


Figure 5.11: Contribution weights of linear fitted model with energy features from selection per dataset

Density Feature

The feature of Density, we defined as average distance of neighbors of a point. The amount of neighbors needed to make for a useful feature in quality prediction, can be concluded from Figure 5.12. From 20 neighbors, the performance is promising, with little differences for $k \in \{20, 25, 30\}$, so any neighborhood size from these can be used.

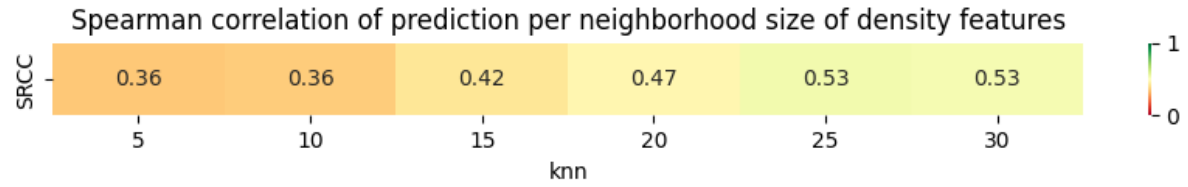


Figure 5.12: Heatmap neighborhood size of density feature
Values are only to be compared within this table, as model parameters differ per experiment run.

Global behaviors From the correlation table of Figure 5.13, we can see that the correlation for the larger neighborhoods are indeed behaving similarly. The other features (in this chapter) showed results of $k = 20$ performing best, for sake of consistency, we will keep this trend of using a neighborhood of size 20.

Since the set of density features is only of size 17, manual recursive feature elimination based on Figure 5.13 produces a set of 4 features that together perform best: {Density$\{H, \eta, b, d\}$}

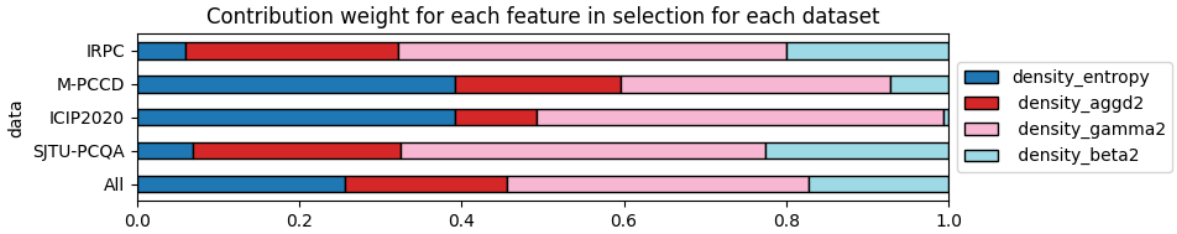


Figure 5.14: Contribution weights of linear fitted model with Density features from selection per dataset

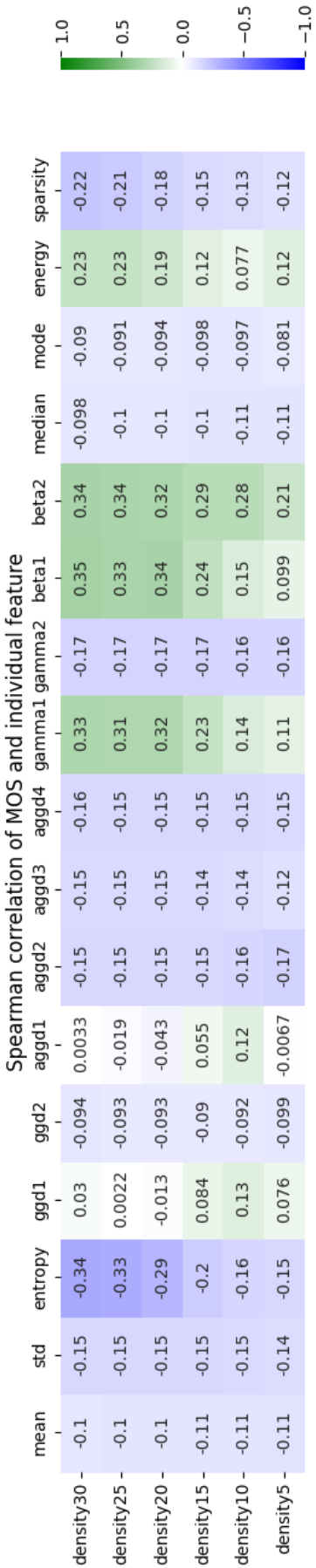


Figure 5.13: Heatmap of the Spearman correlation coefficient of each global descriptor (x-axis) per density feature calculated with different neighborhood sizes (y-axis) with the MOS value. Data from all available datasets is used. *Values are only to be compared within this table, as model parameters differ per experiment run.*

5.2. Distortion Analysis

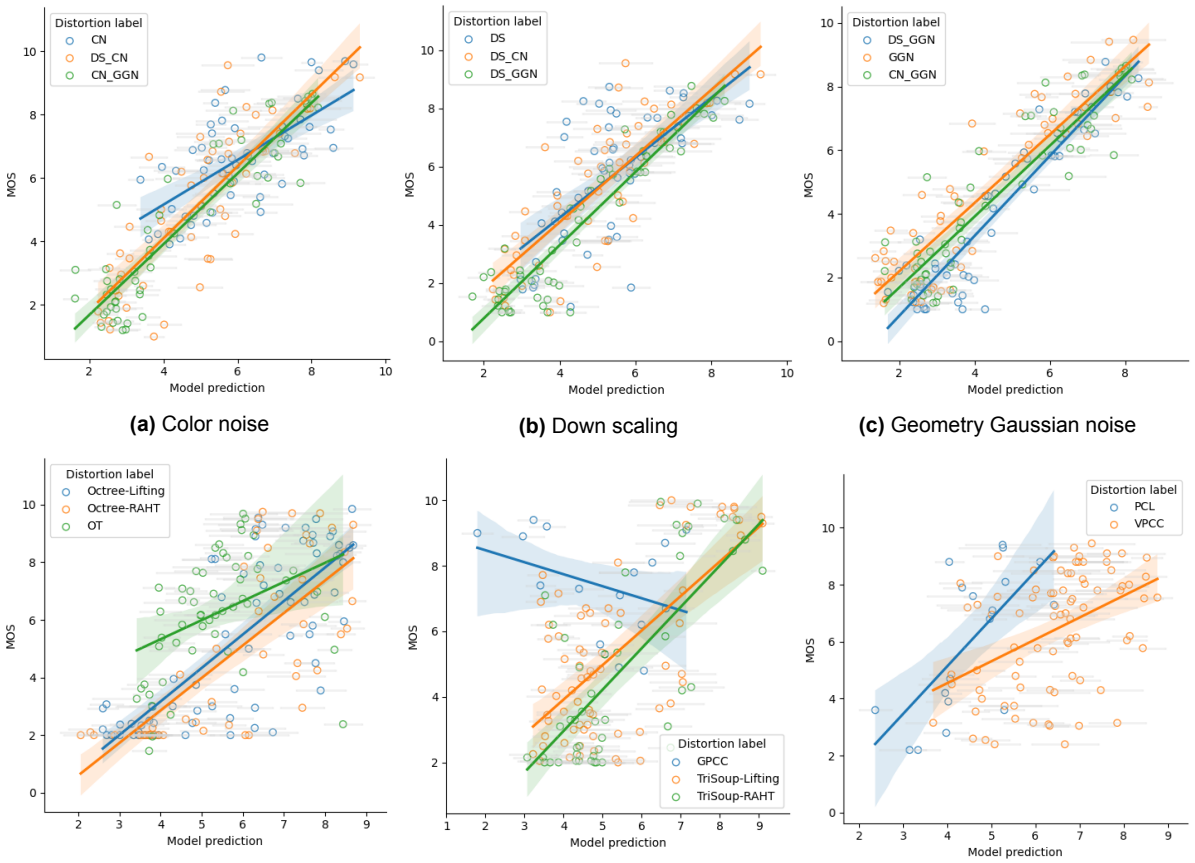
In the Feature Analysis, we constructed a set of features for each type with high potential to perform well in quality assessment. We continue the experiment by taking into account the presence of different types of distortions. In the ideal situation, we aim to find features which perfectly describe the visual quality of a point cloud. A more realistic observation in this experiment is the following: no feature like this is in the feature pool we are assessing in this thesis. We can see this in both the direct correlation tables of Figures 5.2, 5.5, 5.10, and 5.13 where no individual feature has a correlation coefficient above $r = 0.5$, and in the differences of weight contribution per feature for each dataset in Figures 5.4, 5.8, 5.11, and 5.14, since all datasets have a differently skewed distortion type distribution. In this section we first analyze the of the found feature set in the last section for different distortions, including prediction performance per distortion, and correlation of each feature and each distortion. After this we do another feature elimination experiment, where we use feature ranking for smaller SVR models based on data with a single distortion, to produce another feature set with more targeted quality prediction power.

Prediction performance per distortion Some performances of linearly fitted SVR models were reported in the last section. These results are the average correlation coefficient from all test sets. The test sets were all possible combinations of two point cloud objects, including all their available distortions. In Table 4.1 all 19 objects and their distortions are reported, forming 171 experiment data partitions in training and test data. To get an idea of how the differently distorted point clouds are predicted, we can gather all predicted values from the test sets and analyze each case.

We visualize the current performance in this step of the experiment, that is an SVR model using the feature set of the 41/44 selected high potential features, per distortion type. In Figure 5.15 we see a scatter plot of the average prediction of each point cloud object, against the MOS value of this object, each data point is paired by a shaded bar describing the standard deviation of the predictions of this point cloud quality. The 766 data points are split per distortion type and label to clearly show the differences in performance. The objects that have combined distortions are shown in the plot for both distortion types.

Some distortions are less clearly labeled than others, for the dataset IRPC it is not clearly defined which G-PCC variant is used in the compression, they reference to an older version of the MPEG standard G-PCC codec that does not seem to be available as an MPEG standard[1], and mention it using both Octree- and TriSoup-based methods[24].

[1] `https://mpeg.chiariglione.org/standards/mpeg-i`



(a) Color noise

(b) Down scaling

(c) Geometry Gaussian noise

(d) Octree-based compression and G-PCC

(e) TriSoup-based G-PCC, and Undefined G-PCC

(f) V-PCC and Octree-based PCL

Figure 5.15: Average prediction per point cloud against true MOS value, shaded bars are STD value of this average. Linear regression estimated trend line fitted to these value pairs. Prediction based on SVR($kernal = rbf$) models as described in experiment, with feature set of size 41 as described in last section.

We can observe a couple common and different behaviors. We do see that overall the trend lines of the predictions are similar to an x=y line, which is a goal of a perfect prediction model. The spread of the predictions are small (good) for some distortions (CN, DS, GGN, OT), and unfortunately greater for others (G-PCC, V-PCC, PCL). The G-PCC labeled point cloud objects from the IRPC seem to behave very differently from the other objects and look like they are predicted very bad. Why this is the case might be concluded in the next part of the distortion analysis.

Correlation per distortion To find why and how the selected features from the feature analysis show potential for quality assessment, we again look at the correlation coefficient for each individual feature and the MOS, this time split into subsets of the dataset with the same distortion type. The Spearman correlation coefficient can be seen in Figure 5.16, and the Pearson correlation coefficient can be found in the appendix A.1. The heatmap is split in two, since a column is needed for each of the 41 features.

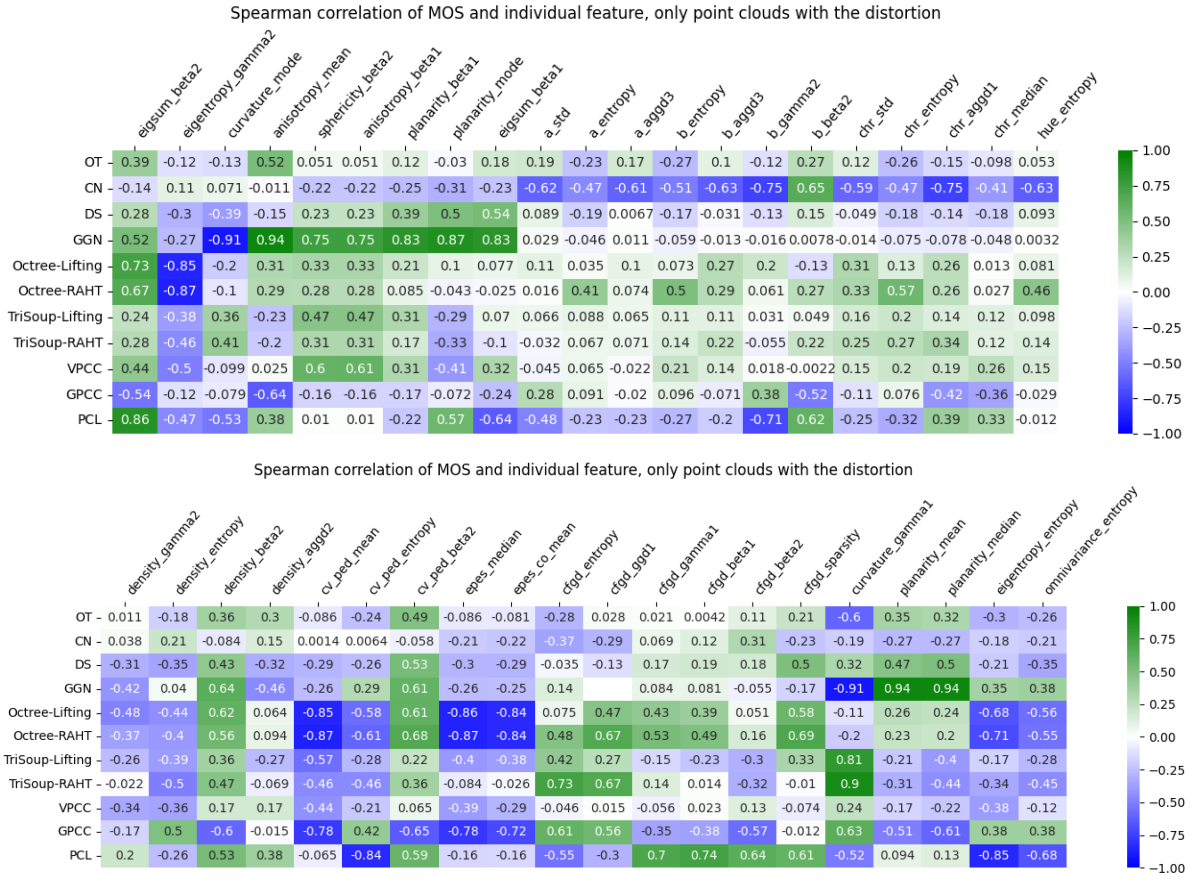


Figure 5.16: Heatmap of the Spearman correlation coefficient of each feature (x-axis) for a subset with distortion label (y-axis) with the MOS value.

From this figure, a reason for the difference in performance of the G-PCC distortion and the others can be found. This distortion type has the opposite behavior from all other types for many features in terms of positive or negative correlation coefficient. In Figure 5.17 the feature values for mean of Anisotropy is scattered against the MOS values of the Geometry Gaussian noise (GGN) and G-PCC objects. While for GGN objects the quality increases from 6 to 9 between the feature value range of 0.75-0.9, for G-PCC objects the opposite is true. For many features the distortions behave differently in sign of the correlation coefficient, and for G-PCC the most.

Figure 5.17 shows scatter plots of a feature with a good correlation coefficient and the MOS of a distortion subset. The point cloud objects are labeled to be able to see how they behave individually.

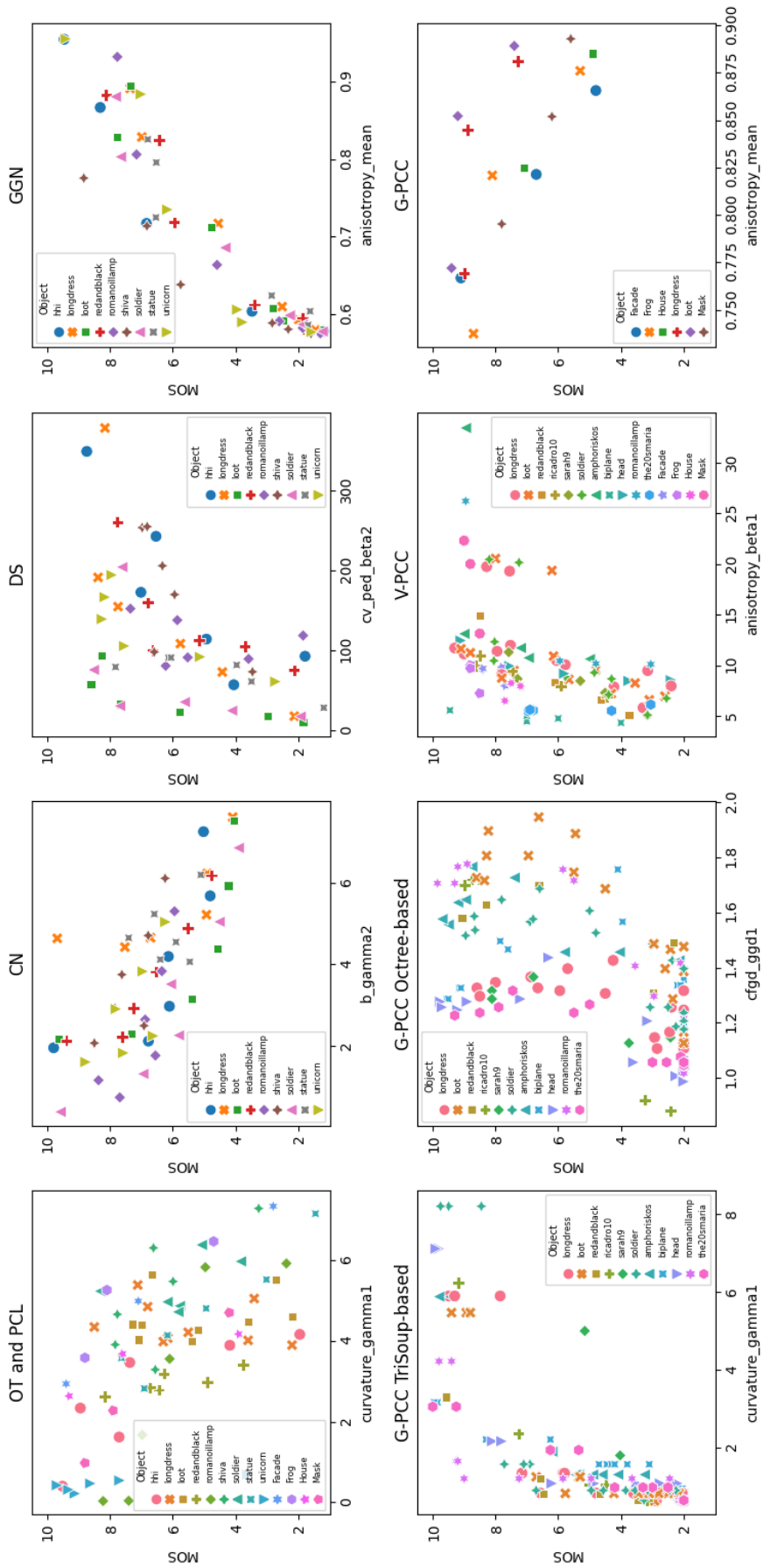


Figure 5.17: Scatter plot of a single feature value with MOS of a distortion subset. Point cloud objects are individually labeled by marker shape and color. Distortions shown are OT&PCL from SJTU and IRPC, CN, DS and GGN from SJTU, G-PCC TriSoup-lifting and -RAHT from from ICIP and M-PCC, G-PCC Octree-lifting and -RAHT from ICIP and M-PCCD, V-PCC from ICIP, M-PCCD and IRPC, and G-PCC from IRPC

If each variant of a point cloud object follows a monotonic downward or upward trend, it is suitable for point-based full-reference metrics, as these metrics base the quality score on the difference of the values rather than the absolute value. A trend over all data points with small variance shows the applicability of the feature in no-reference metrics. All features show an upwards or downwards trend, but some with more variance than others.

The best correlation coefficient from Figure 5.16 is clear in the plot of GGN MOS values and mean of Anisotropy. Both the general trend and for each object(marker shape and color), the feature value increased with the MOS value with similar values. We see the same behavior with more variance in the Color noise and Down-sampling distortion plots.

In the plot of OT and PCL, an object feature value is staying within a small range when the MOS value changes. We do see a global trend of objects with low value for the Gamma shape parameter for the Curvature distribution do have a higher quality score, which is useful information in our context.

In the Octree-based G-PCC plot, we see the feature values of below 1.2 indicate low quality and the values above 1.5 are generally higher. Within these sections there is no clear trend, and a lot of objects fall within the range of no correlation.

From the TriSoup-based G-PCC and V-PCC plots we can see their feature as a higher range where most object only have a high MOS rating.

Feature elimination based on distortion As an additional method of finding features that have high potential for no-reference quality assessment, we experiment with feature ranking with models designed for estimating quality of specific distortions. We fit linear SVR models per single distortion type, and run feature ranking with recursive feature elimination until the 5 most contributing features are left.

Distortion type	Result of feature elimination until 5
OT	Density$\{\sigma_r^2\}$, Linearity$\{Sp\}$, Sphericity$\{a\}$, Variation$\{\nu, c\}$
CN	Chroma$\{\beta\}$, EPES-CO$\{\nu, d, Sp\}$, L$\{Sp\}$
CN, DS+CN, CN+GGN	a*$\{En, H\}$, Curvature$\{a\}$, Variation$\{c, Sp\}$
DS	Anisotropy$\{a\}$, Omnivariance$\{\eta\}$, Planarity$\{\nu\}$, Variation$\{H, Sp\}$
DS, DS+CN, DS+GGN	Anisotropy$\{d\}$, Planarity$\{En, \mu\}$, Variation$\{H, Sp\}$
GGN	Eigenval Sum$\{\sigma_r^2\}$, EPES-CO$\{\nu\}$, Sphericity$\{\sigma_r^2, Sp\}$, Variation$\{\nu\}$
GGN, DS+GGN, CN+GGN	CV-PED$\{\eta\}$, Density$\{\nu\}$, Sphericity$\{a\}$, Variation$\{H, Sp\}$
V-PCC	Anisotropy$\{c\}$, CV-PED$\{\nu\}$, Eigenval Entropy$\{\nu, \beta\}$, Eigenval Sum$\{\nu\}$
Octree	CFGD$\{\beta, \mu, Md\}$, Sphericity$\{\nu, Sp\}$
Trisoup	CFGD$\{H\}$, Density$\{\eta\}$, Linearity$\{\sigma_r^2, Mo\}$, PED$\{H\}$
G-PCC	Anisotropy$\{\nu\}$, Eigenval Entropy$\{\gamma\}$, Linearity$\{a\}$, PED$\{\gamma\}$, Sphericity$\{\nu\}$
G-PCC variants	CFGD$\{Md\}$, Curvature$\{b\}$, Density$\{a\}$, Sphericity$\{En\}$, Variation$\{\nu\}$
PCL	Chroma$\{d\}$, CFGD$\{\beta, \sigma\}$, Curvature$\{a\}$, Eigenval Entropy$\{c\}$

Table 5.3: Best ranked features for each distortion type

We can see the Octree-based compression, Downscaling, Geometry Gaussian Noise and G-PCC distortions are predicted by various geometric features. We can hypothesize some reasons for this. It can be the case that the distortion is easily detected by a lot of different features, and the gathered top 5 is just a random grab of all efficient options. It can also be the case that the distortion is hard to detect by all features under investigation in this experiment, and the top 5 is a random subset of all features. We do see a pattern of the features Variation and Sphericity performing well, so we can say these have a proper predicting power. For the Color noise distortion, we can make a surprising observation. Where we observed in section 5.1 that the behavior of EPES-CO on color distortion was non-correlated with quality, here it is together with the Chroma and Luminance attributes most contribution to a model fit on only color distorted data points. With the added information of Chroma and Luminance, a fitting with the quality score can be modeled. The a* attribute seems to contribute the most individually in terms of detecting color noise when selecting 5 features to find the combined distortion quality values (with Downscaling and Geometry Gaussian Noise), in combination with geometry features.

Some distortions, Octree-lifting and the octree-based approach PCL, rely more on the energy feature CFGD. Octree-lifting removes a lot of points and details, like a color blurring, so the color differences between neighbors changes when the quality is reduced this way. This is detected by the color fluctuation over geometric distance feature.

Final feature sets We have a proposed feature set based on individual performance per feature composed in Section 5.1. Based on the distortion analysis of this section, we make another feature selection. In Table 5.3, we found 5 features for each distortion that together contribute to the best performance in a prediction model. We select the features from distortions OT, CN, DS, GGN, V-PCC, G-PCC, G-PCC variants, and PCL.

The two feature sets to be evaluated in next experiments can be found in Appendix B.1. To summarize the contents of each set, the following histograms report the quantity of each feature and global statistic per the two sets.

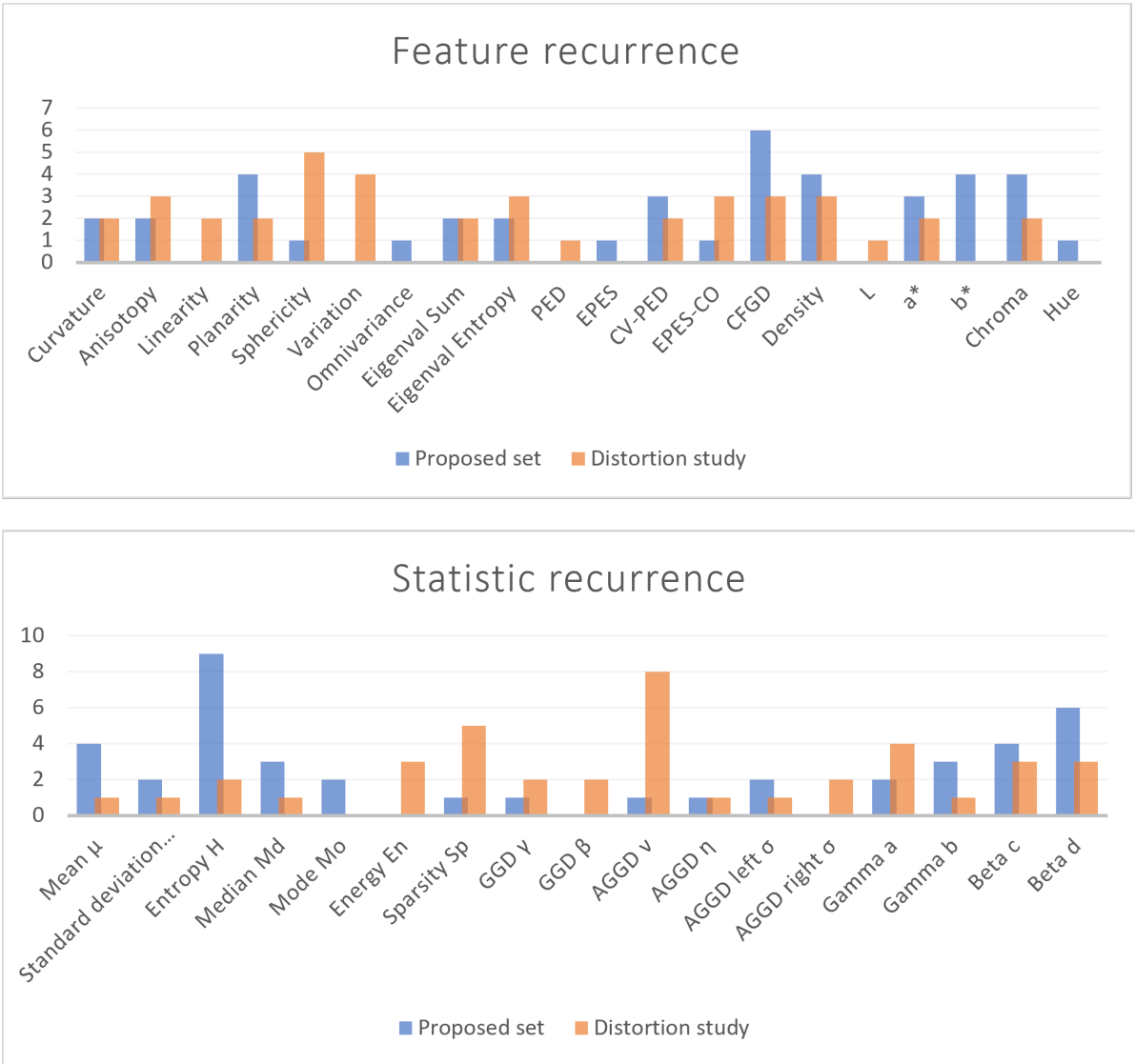


Figure 5.18: Frequency of the features in the final sets.

5.3. Model results

With the observations per distortion and feature, we can try to make a prediction model with the best performance in predicting the quality of a point cloud. To optimize the model, the input features can be selected, and the settings/parameters of the machine learning model can be adjusted.

The state of the art in PCQA evaluation reports the performance per dataset. This way the controlled experiment can be clearly compared to previous work. Eventually, the goal of the project is to create a prediction model for real-time quality assessment, this means the model should work for as much different data as possible, not just within one dataset. To compare with previous work, the experiment will be fitted per dataset, and next to this a global model is presented for all point cloud objects from each gathered dataset.

5.3.1. Dataset models

Most of the state-of-the-art are full-reference methods, so to compare these with our experiment is not ideal. Some datasets have results on deep-learning PCQA models, this is a no-reference method that performs very well thanks to the complexity of such a model.

The experimental setup does not let well to easy adjustment to test full-reference models. The evaluation of machine learning models is different, as we need training data and test data. In full-reference experiments, the weight/fitting of a model is defined for the whole dataset, and it is then reported how each point cloud differs from this fit using the evaluation methods(PLCC, SRCC, RMSE). Recreating the full-reference evaluation setup would only add another table of results that slightly differs from the other tables in every paper on this topic that also slightly differ from each other. The evaluation scores from these metrics are resourced from the most reliable paper available, like whether it is peer-reviewed or if it has a code base to reproduce it. The evaluation of no-reference metric NR-3DQA[57] can be reproduced, since we have evaluated the relevance of their features and the experimental setup is only slightly different from ours. In this section we will use the same SVR settings as well as these seem most effective without exhaustive study (kernel='rbf', ϵ=0.1).

Table 5.4 shows the performance of the datasets, the experiments were done per dataset, so both training and testing data are samples from the same database. The best performance of the experiment section (bottom table) is in **bold**. The "All features" row is a model including all PCA, Energy, and Density features, and the LAB+C color features. The "Proposed set" model includes all features from the feature analysis (41 features), and the "Set without color" model excludes the color features (29 features). The "Distortion study set" model includes the top 5 ranked features from the distortions OT, CN, DS, GGN, V-PCC, G-PCC, G-PCC variants, and PCL as seen in Table 5.3 (40 features).

Table 5.4: Performance evaluation of state-of-the-art quality metrics and proposed features.

Metric	SJTU-PCQA			ICIP2020			M-PCCD			IRPC		
	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE
Point2point MSE [15]	0,237	0,463	2,361	0.946	0.934	0.368	0.845	0.868	0.728	0.643	-	-
Plane2plane MSE [7]	0,796	0,736	1,470	0.925	0.912	0.432	0.624	0.477	1.066	0.246	-	-
PointSSIM [2]	0,828	0,762	1,383	0.869	0.865		0.929	0.936	-	-	-	-
PCQM [32]	0,864	0,854	1,225	0.796	0.832	-	-	-	-	0.873	0.807	-
PCM RR [44]	0,590	0,553	1,974	-	-	-	0.798	0.826	-	-	-	-
PointPCA [4]	0,880	0,853	1,154	-	-	-	-	-	-	-	-	-
BitDance [10]	0,730	0,714		0.936	0.932	0.544	0,819	0,839	1.068	-	-	-
IW-SSIMp[42]	0.794	0.783	1.422	-	-	-	-	-	-	-	-	-
GraphSim[56]	0.856	0.841	1.253	-	-	-	-	-	-	-	-	-
MPED[52]	0.92	0.91	1.00	-	-	-	-	-	-	-	-	-
GQI-VGG16[8]	0.923	0.907	-	0.956	0.966	-	-	-	-	-	-	-
GQI-VGG19[8]	0.925	0.912	-	0.952	0.966	-	-	-	-	-	-	-
NR-3DQA [57]	0.781	0.760	1.644	0.741	0.704	0.825	0.533	0.553	1.220	0.560	0.572	0.885
All features	**0.879**	**0.866**	1.350	0.883	0.884	0.667	0.849	**0.861**	0.871	**0.770**	**0.689**	0.761
Proposed set	0.864	0.843	**1.279**	0.799	0.810	0.823	0.728	0.749	1.016	0.628	0.537	0.846
Set without color	0.811	0.775	1.476	0.842	0.848	0.706	0.724	0.739	0.998	0.640	0.564	0.803
Distortion study set	0.859	0.833	1.428	**0.915**	**0.915**	**0.601**	**0.857**	0.858	**0.847**	0.692	0.599	**0.760**

Performance for SJTU-PCQA

Table 5.4 shows the performance evaluation scores on the SJTU data set. We see the performance of a model using all available features, which for 19 properties and 17 statistical means 323 features, has the highest performance score in this no-reference context. Reducing the feature amount by feature selection based on the top 5 features of the distortions, reduces performance. The handcrafted feature set based on the feature analysis of this chapter is a better feature selection, but does not reach the performance of using all features using the correlation evaluation methods, the root mean squared error is actually reporting the proposed set to be most accurate.

Figure 5.19 shows the predicted quality value average for all point clouds in this data set, with standard deviation bars, from each run of the experiment. To compare, the same experiment is run with the feature set of NR-3DQR[57]. The trend line in both models is close to $x = y$, however the variation does span a wide area. For our proposed model, we can see in the correlation and the plot, the feature set is a little more accurate than the NR-3DQA model.

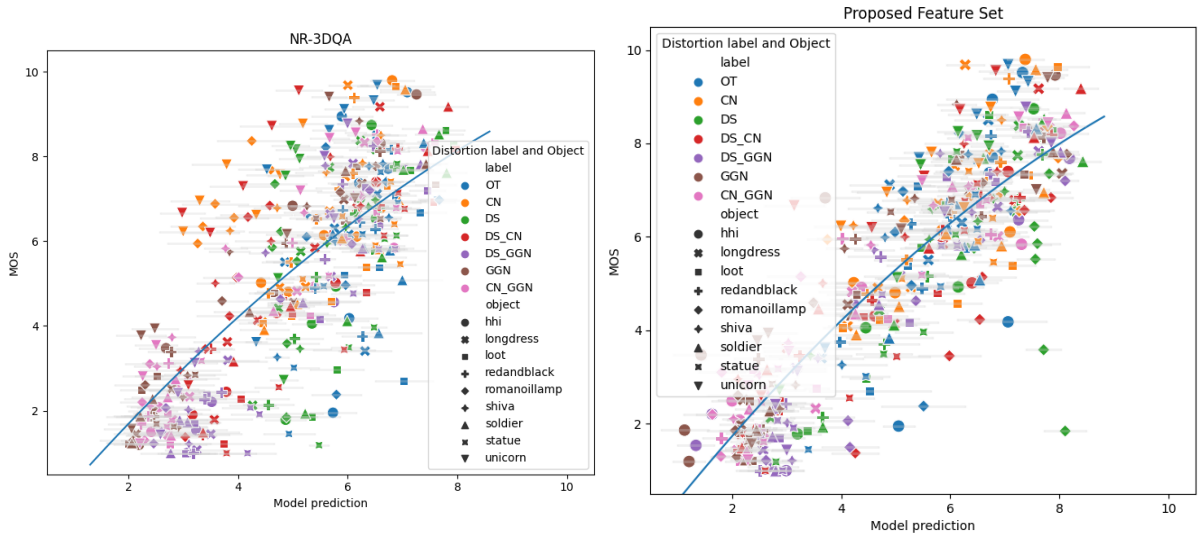


Figure 5.19: Average prediction per SJTU point cloud objects against MOS value, shaded bars are STD value of this average. Quadratic regression estimated trend line fitted to these value pairs. Prediction based on SVR(kernal = rbf) models as described in experiment.

Performance for ICIP2020

For the experiments run on the ICIP2020 dataset, using the feature set of the distortion analysis gives the best performance in terms of Pearson, Spearman and Root Mean Squared Error evaluation. From the proposed handcrafted sets, excluding color features increase the performance compared to the complete set.

The performance of the distortion set is very good, ICIP has distortions VPCC and lifting-based G-PCC. Comparing the feature set of Proposed feature set and Distortion set, we see the overlap is quite small. Both methods of feature selection produced good results, but the Distortion set is reaching full-reference level predictions.

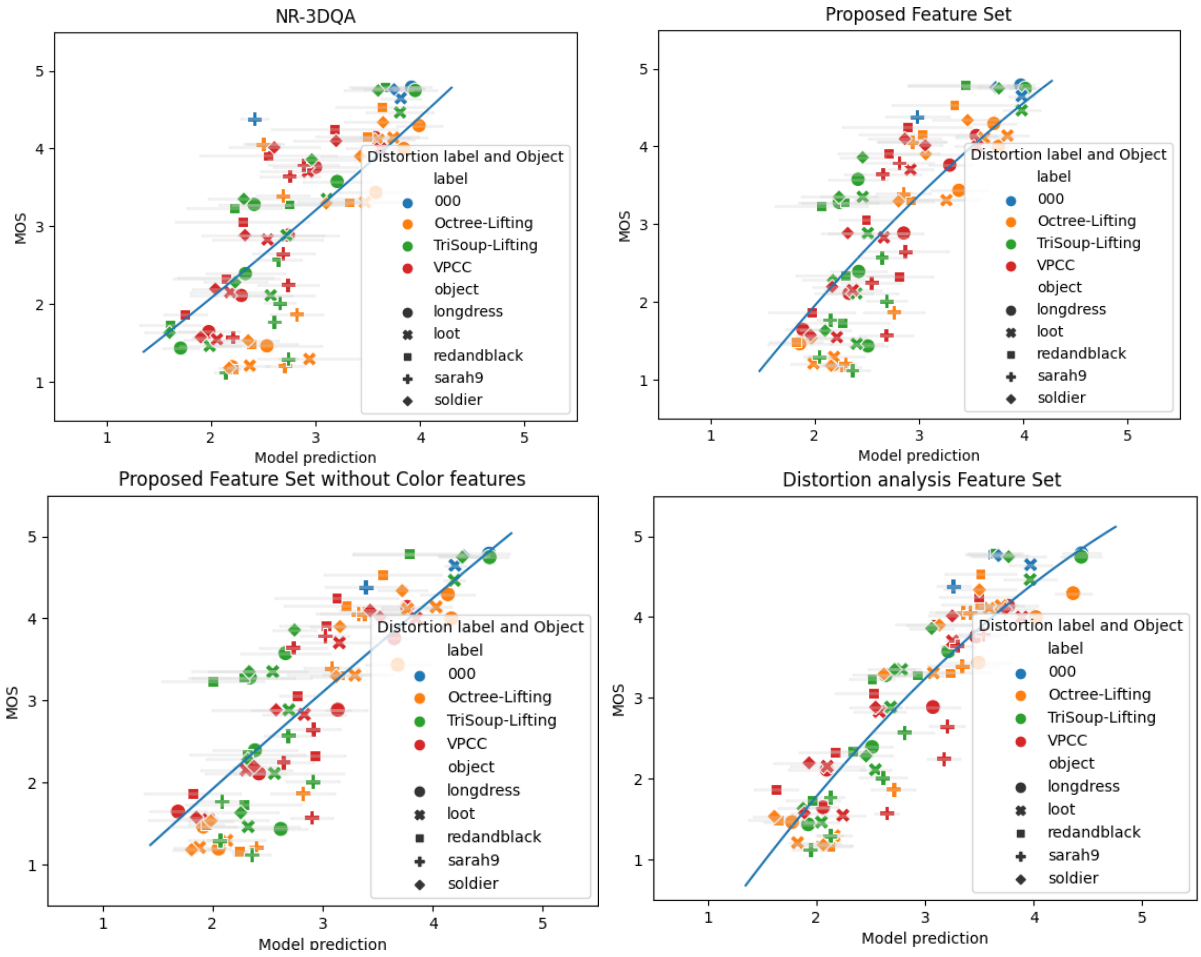


Figure 5.20: Average prediction per ICIP point cloud objects against MOS value, shaded bars are STD value of this average. Quadratic regression estimated trend line fitted to these value pairs. Prediction based on SVR(kernal = rbf) models as described in experiment. 000 distortion label is an object without distortion.

Performance for M-PCCD

On the larger M-PCCD dataset, the large feature set and the distortion set similarly outperform the handcrafted one. The NR-3DQA feature set was not designed for this dataset, as we can see in the reduced performance as well as in the scatter plot. Our proposed feature set is marginally better. In the scatter plot of Figure 5.21 we see the trend is better, but most points have a big range of predictions, making the possible metric not very reliable.

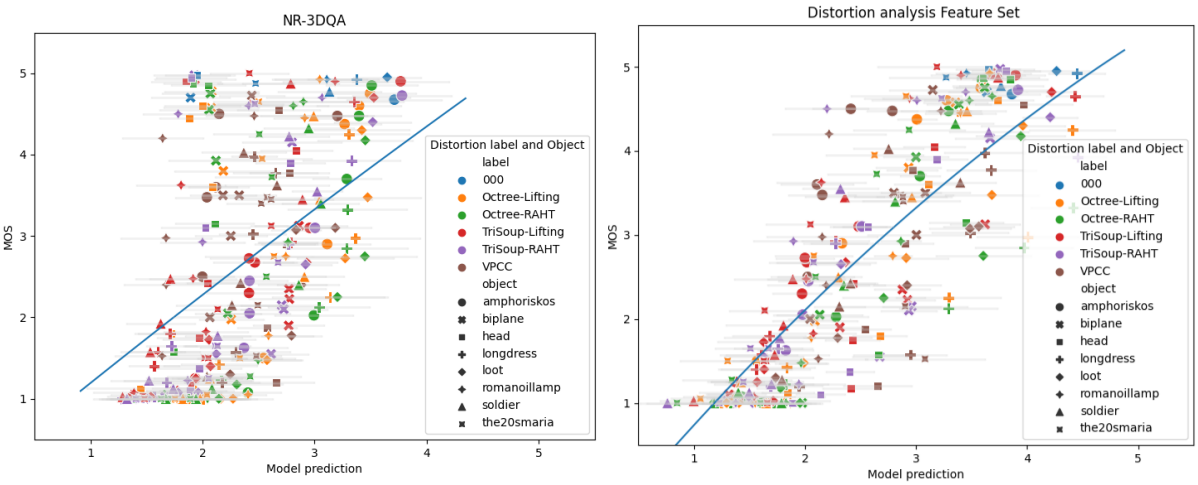


Figure 5.21: Average prediction per M-PCCD point cloud objects against MOS value, shaded bars are STD value of this average. Quadratic regression estimated trend line fitted to these value pairs. Prediction based on SVR(kernal = rbf) models as described in experiment. 000 distortion label is an object without distortion.

Performance for IRPC

This small dataset has the best performance when more features than data points are used to fit this model. This dataset might be too small to use for such a no-reference PCQA task. The practice of selecting 2 point cloud objects to use as test set means for this dataset that the training sets contain 36 data points and the test set is of size 18. The ICIP2020 dataset had the 4:2 ratio as well in its training:testing sets, but the 60:30 data points might be a big enough training pool to have better performance.

In the figures 5.22, the predicted scores all range between 3 and 4, instead of the MOS range of 1-5. We can see the most point cloud objects have a MOS value of 3-5, making a very skewed dataset. Even though this dataset has mostly the same distortions as the previous set, this skewed dataset combined with this PCQA method seems unsuitable in this setting.

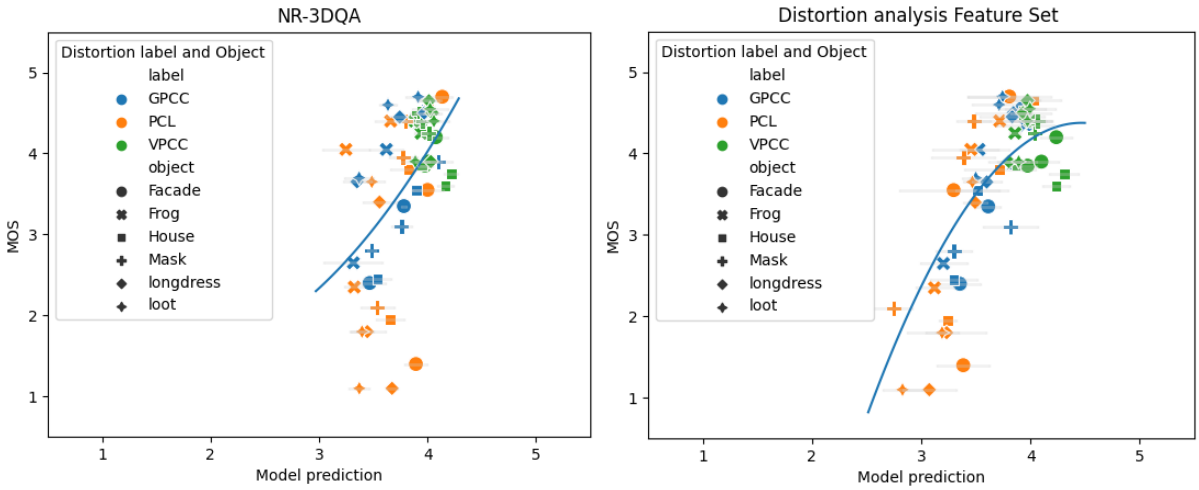


Figure 5.22: Average prediction per IRPC point cloud objects against MOS value, shaded bars are STD value of this average. Quadratic regression estimated trend line fitted to these value pairs. Prediction based on SVR(kernal = rbf) models as described in experiment.

5.3.2. Cross datasets

We see in the tests per dataset, the feature set with most potential is distortion set over the feature analysis set. Since the fitting in these experiments is smaller and can be specific for each dataset, this is not an abnormal observation. We will now test the robustness of the set with cross dataset experiments. First we'll combine all data, this setting is most like the real implementation if a metric would be used in real applications. The second in cross-datasets, this shows us the robustness of the metric when tested on unknown distortions or different implementations of them.

Combining datasets With the same method of training the model on all except 2 point cloud objects, and testing on these two, we run all combinations and average the score for a final performance of the model. The performance scores of these combined models are lower as there is more variation of the distortion algorithms. The feature correlations found in the individual datasets might be overfitting the specific data and correlate less with external data.

The features set which we concluded from the linear correlation of features with the quality scores, performs best in this linear setting. However, the more precise approach, using an RBF kernel in the SVR with all available information, shows a significantly better performance in this experiment.

Table 5.5: Performance evaluation of state-of-the-art quality metrics on combined datasets.

Metric	PLCC	SRCC	RMSE
NR-3DQA RBF [57]	0.602	0.598	2.250
All features RBF	**0.743**	**0.732**	**1.921**
Distortion set RBF	0.734	0.730	1.947
Proposed Set RBF	0.664	0.650	2.091
NR-3DQA Polynomial	0.409	0.453	2.727
All features Polynomial	**0.600**	**0.699**	**2.305**
Distortion set Polynomial	0.529	0.574	2.498
Proposed Set Polynomial	0.450	0.535	2.626
NR-3DQA Linear	0.443	0.450	3.642
All features Linear	0.573	0.585	3.585
Distortion set Linear	0.559	0.543	2.644
Proposed Set Linear	**0.607**	**0.595**	**2.832**

Cross datasets In Table 5.6a, we find the results of models fitted with an RBF-kernel SVR to one dataset, all features, and its performance tested on another. Table 5.6b and 5.6c show experiment results from fitting the model with just the Proposed feature set, and the Distortion study set, respectively. The performance is reported as PLCC/SRCC/RMSE, of which the Pearson and Spearman score should be close to 1, and Root Mean Square Error should be close to 0. The results in **bold** show which of these 3 models performed best on this dataset combination.

Due to overfitting on situations unique to the datasets, the scores are expected to be lower than in previous experiments. Datasets with similar distortions like M-PCCD and ICIP match better than datasets with completely different distortions like SJTU. A larger training set of M-PCCD performs better in this machine learning setting then smaller datasets of ICIP2020 and IRPC.

When using the handcrafted feature set, we see the performance of the experiments with the SJTU dataset go up and the rest go down. This could indicate that this selection is more robust when predicting quality of unknown distortions. The distortion feature set performs on the other datasets best or almost the same as using all features.

If the quality of point clouds of which the distortion type is known is to be predicted, using a feature set created with this distortion type is recommended. If the distortion type is unknown, a more robust feature set based on correlation with all distortion types has more potential to be a recommended metric.

Table 5.6: Performance evaluation of cross dataset models reported as PLCC/SRCC/RMSE

(a) All features

	M-PCCD test	ICIP test	IRPC test	SJTU test
M-PCCD train		**0.89**/**0.89**/**1.64**	0.57/0.46/**3.29**	0.08/0.05/2.78
ICIP train	0.68/0.72/2.45		0.46/0.38/2.83	0.15/0.10/2.46
IRPC train	0.38/0.33/4.03	0.48/0.36/3.08		-0.10/-0.14/3.25
SJTU train	0.10/0.11/3.42	0.35/0.30/2.51	**0.36**/**0.19**/**2.63**	

(b) Proposed feature set

	M-PCCD test	ICIP test	IRPC test	SJTU test
M-PCCD train		0.78/0.81/1.96	0.35/0.27/3.54	**0.29**/**0.28**/**2.75**
ICIP train	0.63/0.65/2.58		0.25/0.14/2.97	**0.43**/**0.44**/**2.23**
IRPC train	0.28/0.26/4.03	0.17/0.03/3.15		**0.30**/**0.33**/**3.09**
SJTU train	**0.16**/**0.16**/**3.29**	**0.42**/**0.39**/**2.44**	0.23/0.08/3.00	

(c) Distortion feature set

	M-PCCD test	ICIP test	IRPC test	SJTU test
M-PCCD train		0.87/0.87/1.68	**0.63**/**0.54**/3.32	-0.11/-0.14/2.98
ICIP train	**0.72**/**0.75**/**2.40**		**0.58**/**0.56**/**2.46**	-0.26/-0.20/2.92
IRPC train	**0.57**/**0.62**/**3.91**	**0.76**/**0.77**/**2.97**		-0.01/-0.04/3.24
SJTU train	0.08/0.10/3.53	0.25/0.19/2.70	0.31/0.12/2.78	

6

Discussion and Conclusion

First, some discussion points are brought up in Section 6.1. Then, we will make a conclusion based on the research questions from Chapter 1 in Section 6.2. We end on possible future directions of this research in Section 6.3.

6.1. Discussion

Before we come back to the research questions to make a conclusion, we first extend on the discussion reported in the results chapter.

6.1.1. Feature relevance ranking

In Section 5.1 and 5.2 we concluded two sets of high potential features to use in the task of no-reference quality assessment. No features were found that individually have enough predicting power to completely fit with the user given quality score. In Figure 5.16, we do see features can have high correlation with individual distortion cases. In Figure 5.18 we observe the recurrence of features and global statistics in the feature sets, where the biggest distance between these two sets in the global statistics used.

In the handcrafted feature set from Section 5.1, the features that correlate across all distortion data were gathered, we see the mean value, the entropy of the values and the Beta shape and scale parameters occur frequently. In the set generated with split data on distortions from Section 5.2, the statistical values of energy, sparsity, AGGD shape parameter, and Gamma shape parameter are found to fit best to quality the specific distortion cases. In both sets, all feature types are represented, the Distortion study set selected more PCA features replacing color features, which is to be expected with the ratio of color distortions in the dataset.

Per distortion type, we can summarize a selection of features effective in that context. Examples are the PCA features being very effective in predicting Generalized Gaussian Noise, and Energy features performing well on point clouds effected by Octree-based G-PCC. Figures like Figure 5.16 can be a good base for creating prediction models per distortion.

6.1.2. Full-Reference to No-Reference Quality Assessment

The results of the experiments show potential in this approach to find a reliable no-reference quality metric for point clouds. The evaluation method of reporting correlation and error are comparable with full-reference metrics, in terms of both evaluation metrics and values. However, the evaluation experiment does differ in terms of set up, so we did not recreate full-reference evaluation methods, and instead sourced other experiment results for performance values. To properly compare and generate comparable performance results, it would have been better to implement a full-reference testing environment similar to our experiment. This environment could use the same testing and training data split as this experiment.

The cross-dataset experiment shows much room for improvement. The current features under review do not correlate with all distortions, so when fitting to one, the other will not be able to be accurately predicted.

6.1.3. Computational resources

Of the experiment pipeline, the most time-consuming step is the extraction of features for all point clouds, which takes multiple seconds per point cloud. The implementation is currently in Python which is not the fastest programming language for this computational task. Calculating the energy features also takes a lot of memory to compute, with the current Python implementation. After all features are extracted, fitting the model and predicting quality is a very fast procedure. To see this kind of quality assessment in real time, would mean a speed-up of the feature extraction process is needed. If we are estimating the quality of a single point cloud, the process now takes between 5 and 30 seconds, depending on the point cloud size. Implementing this method in C++ will probably improve the potential of using this method in real time. As mentioned earlier, the experimental setup of testing no-reference metrics is different from full-reference. We did not run other metrics on the same datasets to compare computational performance. Previous work that is comparable to our experiment[57] did a comparison in Computational Efficiency for other model-based methods GraphSIM, PointSSIM, PCQM, and PCMRR, and reported this method achieves relatively fast processing times.

6.1.4. Incompatible features

More experiments on other features then discussed in Chapter 5 were started. These experiments were unsuccessful, mostly due to the time-consuming processing of these features to properly analyze, but also due to showing little potential to answer the research questions. It is not the conclusion of this thesis that these features are not suitable for no-reference quality assessment, further research with a different approach to test these is recommended.

Voxel patterns The current implementation of extracting Local Luminance Patterns does not result in a feature vector with useful information to extract statistical features

from that can represent the point cloud for quality prediction.

The best performance of this feature is reached with mean, std and entropy descriptors (SRCC=0.24), where the performance of predicting color distortion (Downscaling and Color noise (SRCC=0.41), Color noise and Geometry Gaussian noise (SRCC=0.49) is okay at best.

In terms of computational time, this feature took a lot of time to process. Due to both showing little potential, and taking a long time to execute an experiment, the decision was made to remove these features from the analysis pool.

Point Feature Histograms Another feature under investigation from the literature study was Point Feature Histograms. More research on the feasibility of testing this feature in this experiment proved this feature to be incompatible with the pipeline as well, due to the computational time. The experiment was halted due to taking too long, and no data has been generated to make observations about.

6.2. Conclusion

Whether the goal **estimate perceptual quality of a point cloud when no high-quality reference point cloud is available** is already within reach after this research depends on the specific context this task is applied to. We tried to find point cloud properties to use in a model to predict the user given quality score. The results on parts of the process is certainly promising, like certain known distortions on unknown objects are properly predictable using fitted weights on this distortion. However, a single defining quality distributor is not yet found.

Can (existing) local features of point clouds be adapted to global descriptors?

To answer this question, we selected multiple local properties to transform into global descriptors.

We took multiple statistical attributes of the data on all the points to combine into scalars. This method can be applied to any local feature. These attributes seem to still convey enough unique information to work with in the setting of quality assessment. We looked into Summery statistics and distribution parameters of (Asymmetric) Generalized Gaussian, Beta, and Gamma distribution. For each analyzed local feature, different global attribute sets were most effective in this context. For some the mean value already has useful correlation (planarity, energy), but most features are best adapted to a distribution parameter to correlate with quality.

Do any global descriptors correlate with perceptual quality?

We analyzed the created global descriptors to find linear correlations with the quality score.

We found many features that somewhat linearly correlate with some distortion types. We found features that have a 0.94 Pearson and Spearman correlation coefficient with Geometry Gaussian Noise. For the V-PCC distortion, the Pearson correlation coefficient caps at 0.5. In the analyzed feature set, there is not an all-embracing

feature that has good correlation with all reviewed distortions. A new combined descriptor based on the best performing features and weighted per distortion could be created in the future to better correlate with a universal quality value.

Can global descriptors be used to create a model to estimate the perceptual quality of a point cloud?
With the selected global descriptors, we used Support Vector Regression to find the fitting of the features to the quality score. Varying performances are achieved by training the models on different data subsets, and different fitting kernel functions.

Even though the selected global descriptors seem like they already correlate linearly with the quality scores, using a radial basis function is needed to reach prediction performances that compare to the state-of-the-art of full-reference methods. The performance of the linear models is lower, but since this setting of no-reference quality assessment is a real challenge, the achieved performance is impressive.

How can the perceptual quality of a point cloud be estimated when a high-quality reference point cloud is not available?
From the analysis of existing point cloud features in the context of no-reference quality assessment, we proposed estimating the perceptual quality of a point cloud by extracting most relevant features and calculate a predicted quality using a pretrained model.

The results show potential, but our model is not robust and fast enough to perform a reliable assessment of quality in a real-time environment. We are just at the beginning of exploring no-reference quality assessment, and while the current methods are not yet appliable in real-time scenarios, future work can fill many knowledge gaps to reach this goal.

6.3. Future work

The performance reached in this thesis is not yet suitable to translate to real application. Extending this work directly would mean testing more local features to see if their global translation has a correlation with quality. A start would be the voxel features that did not work in this experiment. We did not create new features to better imitate the quality behavior of the point clouds. Combining quality-correlating features could create a high performing feature for this task.

Continuing this work in a programming environment more suitable for big data (C++ instead of Python) could already improve the feasibility of using this approach in real-time scenarios.

To take into account the knowledge that different distortions can be detected properly by different methods, this could lead to a different approach to quality assessment by using an additional distortion classification step before assessing quality. Adjusting the codebase of this thesis slightly already shows promise for this approach. Using the calculated features and fitting the distortion label instead of the quality score to these, we produce the following confusion matrix of distortion type prediction:

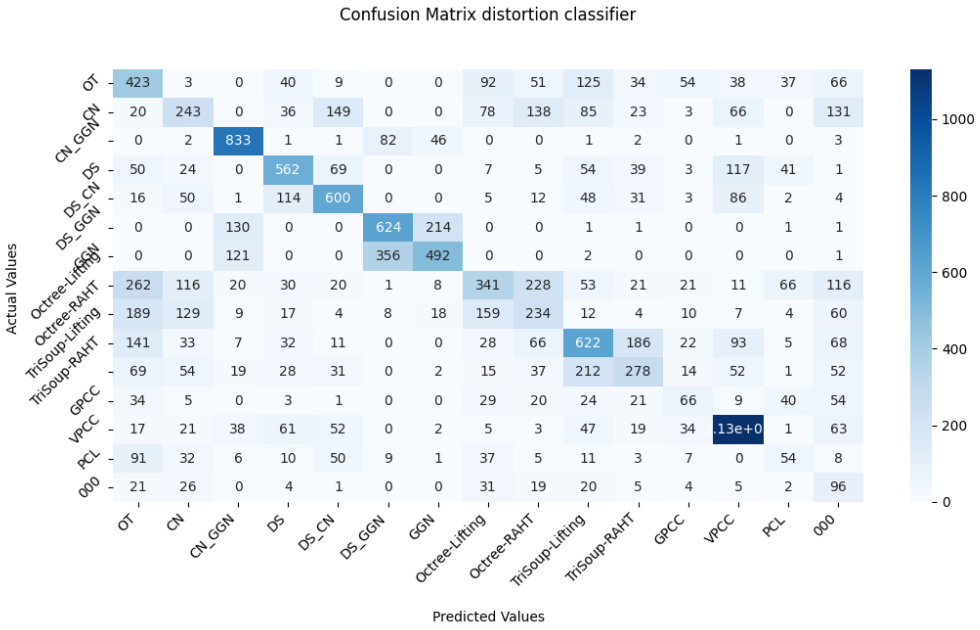


Figure 6.1: Confusion matrix of classifying distortion type, all available datasets. Values are the sum of all classifications of the cases.

References

[1] Ilyass Abouelaziz, Mohammed El Hassouni, and Hocine Cherifi. “No-Reference 3D Mesh Quality Assessment Based on Dihedral Angles Model and Support Vector Regression”. en. In: *Image and Signal Processing*. Ed. by Alamin Mansouri et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 369–377. ISBN: 978-3-319-33618-3. DOI: 10.1007/978-3-319-33618-3_37.

[2] Evangelos Alexiou and Touradj Ebrahimi. “Towards a Point Cloud Structural Similarity Metric”. In: *2020 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*. July 2020, pp. 1–6. DOI: 10.1109/ICMEW46912.2020.9106005.

[3] Evangelos Alexiou, Evgeniy Upenik, and Touradj Ebrahimi. “Towards subjective quality assessment of point cloud imaging in augmented reality”. In: *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*. ISSN: 2473-3628. Oct. 2017, pp. 1–6. DOI: 10.1109/MMSP.2017.8122237.

[4] Evangelos Alexiou, Irene Viola, and Pablo Cesar. “PointPCA: Point Cloud Objective Quality Assessment Using PCA-Based Descriptors”. In: *arXiv:2111.12663 [cs]* (Nov. 2021). arXiv: 2111.12663.

[5] Evangelos Alexiou, Nanyang Yang, and Touradj Ebrahimi. “PointXR: A Toolbox for Visualization and Subjective Evaluation of Point Clouds in Virtual Reality”. In: *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. ISSN: 2472-7814. May 2020, pp. 1–6. DOI: 10.1109/QoMEX48832.2020.9123121.

[6] Evangelos Alexiou et al. “A comprehensive study of the rate-distortion performance in MPEG point cloud compression”. en. In: *APSIPA Transactions on Signal and Information Processing* 8 (2019), e27. ISSN: 2048-7703. DOI: 10.1017/ATSIP.2019.20.

[7] Evangelos Alexiou et al. “Point Cloud Subjective Evaluation Methodology based on 2D Rendering”. In: *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. ISSN: 2472-7814. May 2018, pp. 1–6. DOI: 10.1109/QoMEX.2018.8463406.

[8] Aladine Chetouani et al. “Deep Learning-Based Quality Assessment Of 3d Point Clouds Without Reference”. In: *2021 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*. July 2021, pp. 1–6. DOI: 10.1109/ICMEW53276.2021.9455967.

[9] P. Cignoni, C. Rocchini, and R. Scopigno. “Metro: Measuring Error on Simplified Surfaces”. en. In: *Computer Graphics Forum* 17.2 (1998). _eprint: https://onlinelibrary.wiley.com 8659.00236, pp. 167–174. ISSN: 1467-8659. DOI: 10.1111/1467-8659.00236.

[10] Rafael Diniz, Pedro Garcia Freitas, and Mylène C. Q. Farias. “Color and Geometry Texture Descriptors for Point-Cloud Quality Assessment”. In: *IEEE Signal Processing Letters* 28 (2021). Conference Name: IEEE Signal Processing Letters, pp. 1150–1154. ISSN: 1558-2361. DOI: 10.1109/LSP.2021.3088059.

[11] Rafael Diniz, Pedro Garcia Freitas, and Mylène C. Q. Farias. “Towards a Point Cloud Quality Assessment Model using Local Binary Patterns”. In: *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. ISSN: 2472-7814. May 2020, pp. 1–6. DOI: 10.1109/QoMEX48832.2020.9123076.

[12] Rafael Diniz, Pedro Garcia Freitas, and Mylène C.Q. Farias. “Local Luminance Patterns for Point Cloud Quality Assessment”. In: *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*. ISSN: 2473-3628. Sept. 2020, pp. 1–6. DOI: 10.1109/MMSP48831.2020.9287154.

[13] Yu Fan et al. *A No-reference Quality Assessment Metric for Point Cloud Based on Captured Video Sequences*. arXiv:2206.05054 [cs, eess]. June 2022.

[14] Jason Geng. “Structured-light 3D surface imaging: a tutorial”. EN. In: *Advances in Optics and Photonics* 3.2 (June 2011). Publisher: Optica Publishing Group, pp. 128–160. ISSN: 1943-8206. DOI: 10.1364/AOP.3.000128.

[15] D. Girardeau-Montaut et al. *Change Detection on Points Cloud Data Acquired with a Ground Laser Scanner*. 2005.

[16] D. Graziosi et al. “An overview of ongoing point cloud compression standardization activities: video-based (V-PCC) and geometry-based (G-PCC)”. en. In: *APSIPA Transactions on Signal and Information Processing* 9 (2020), e13. ISSN: 2048-7703. DOI: 10.1017/ATSIP.2020.12.

[17] Xian-Feng Han et al. “3D Point Cloud Descriptors in Hand-crafted and Deep Learning Age: State-of-the-Art”. In: *arXiv:1802.02297 [cs]* (July 2020). arXiv: 1802.02297.

[18] Miles Hansard et al. *Time-of-Flight Cameras: Principles, Methods and Applications*. en. SpringerBriefs in Computer Science. London: Springer London, 2013. ISBN: 978-1-4471-4657-5 978-1-4471-4658-2. DOI: 10.1007/978-1-4471-4658-2.

[19] Lei Hua et al. “CPC-GSCT: Visual quality assessment for coloured point cloud based on geometric segmentation and colour transformation”. en. In: *IET Image Processing* 16.4 (2022). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/ipr2.12211, pp. 1083–1095. ISSN: 1751-9667. DOI: 10.1049/ipr2.12211.

[20] ITU-T P.1401. *Methods, metrics and procedures for statistical evaluation, qualification and comparison of objective quality prediction models*. International Telecommunication Union. July 2012.

[21] Jack Jansen et al. “A pipeline for multiparty volumetric video conferencing: transmission of point clouds over low latency DASH”. en. In: *Proceedings of the 11th ACM Multimedia Systems Conference*. Istanbul Turkey: ACM, May 2020, pp. 341–344. ISBN: 978-1-4503-6845-2. DOI: 10.1145/3339825.3393578.

[22] Alireza Javaheri et al. “A Generalized Hausdorff Distance Based Quality Metric for Point Cloud Geometry”. In: *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. ISSN: 2472-7814. May 2020, pp. 1–6. DOI: 10.1109/QoMEX48832.2020.9123087.

[23] Alireza Javaheri et al. “Mahalanobis Based Point to Distribution Metric for Point Cloud Geometry Quality Evaluation”. In: *IEEE Signal Processing Letters* 27 (2020). Conference Name: IEEE Signal Processing Letters, pp. 1350–1354. ISSN: 1558-2361. DOI: 10.1109/LSP.2020.3010128.

[24] Alireza Javaheri et al. “Point Cloud Rendering after Coding: Impacts on Subjective and Objective Quality”. In: *IEEE Transactions on Multimedia* (2020). Conference Name: IEEE Transactions on Multimedia, pp. 1–1. ISSN: 1941-0077. DOI: 10.1109/TMM.2020.3037481.

[25] Michael Kazhdan and Hugues Hoppe. “Screened poisson surface reconstruction”. en. In: *ACM Transactions on Graphics* 32.3 (June 2013), pp. 1–13. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/2487228.2487237.

[26] Nour-Eddine Lasmar, Youssef Stitou, and Yannick Berthoumieu. “Multiscale skewed heavy tailed model for texture analysis”. en. In: *2009 16th IEEE International Conference on Image Processing (ICIP)*. Cairo, Egypt: IEEE, Nov. 2009, pp. 2281–2284. ISBN: 978-1-4244-5653-6. DOI: 10.1109/ICIP.2009.5414404.

[27] Qi Liu et al. “No-reference Bitstream-layer Model for Perceptual Quality Assessment of V-PCC Encoded Point Clouds”. In: *IEEE Transactions on Multimedia* (2022). Conference Name: IEEE Transactions on Multimedia, pp. 1–1. ISSN: 1941-0077. DOI: 10.1109/TMM.2022.3177926.

[28] Qi Liu et al. “PQA-Net: Deep No Reference Point Cloud Quality Assessment via Multi-view Projection”. In: *IEEE Transactions on Circuits and Systems for Video Technology* (2021). Conference Name: IEEE Transactions on Circuits and Systems for Video Technology, pp. 1–1. ISSN: 1558-2205. DOI: 10.1109/TCSVT.2021.3100282.

[29] Qi Liu et al. “Reduced Reference Perceptual Quality Model and Application to Rate Control for 3D Point Cloud Compression”. In: *IEEE Transactions on Image Processing* 30 (2021). arXiv: 2011.12688, pp. 6623–6636. ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2021.3096060.

[30] Yipeng Liu et al. “Point Cloud Quality Assessment: Dataset Construction and Learning-based No-Reference Approach”. In: *arXiv:2012.11895 [eess]* (July 2021). arXiv: 2012.11895.

[31] Donald Meagher. “Geometric Modeling Using Octree-Encoding”. In: *Computer Graphics and Image Processing* 19 (June 1982), pp. 129–147. DOI: 10.1016/0146-664X(82)90104-6.

[32] Gabriel Meynet et al. “PCQM: A Full-Reference Quality Metric for Colored 3D Point Clouds”. In: *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. ISSN: 2472-7814. May 2020, pp. 1–6. DOI: 10.1109/QoMEX48832.2020.9123147.

[33] Yana Nehmé et al. “Visual Quality of 3D Meshes With Diffuse Colors in Virtual Reality: Subjective and Objective Evaluation”. In: *IEEE Transactions on Visualization and Computer Graphics* 27.3 (Mar. 2021). Conference Name: IEEE Transactions on Visualization and Computer Graphics, pp. 2202–2219. ISSN: 1941-0506. DOI: 10.1109/TVCG.2020.3036153.

[34] M. Okutomi and T. Kanade. “A multiple-baseline stereo”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.4 (Apr. 1993). Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 353–363. ISSN: 1939-3539. DOI: 10.1109/34.206955.

[35] Stuart Perry et al. “Quality Evaluation Of Static Point Clouds Encoded Using MPEG Codecs”. In: *2020 IEEE International Conference on Image Processing (ICIP)*. ISSN: 2381-8549. Oct. 2020, pp. 3428–3432. DOI: 10.1109/ICIP40778.2020.9191308.

[36] Maurice Quach et al. “Survey on Deep Learning-based Point Cloud Compression”. In: *Frontiers in Signal Processing* 2 (2022). Publisher: Frontiers. DOI: 10.3389/frsip.2022.846972.

[37] Radu Bogdan Rusu and Steve Cousins. “3D is here: Point Cloud Library (PCL)”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China: IEEE, May 2011.

[38] Markus Schütz, Katharina Krösl, and Michael Wimmer. “Real-Time Continuous Level of Detail Rendering of Point Clouds”. In: *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. ISSN: 2642-5254. Mar. 2019, pp. 103–110. DOI: 10.1109/VR.2019.8798284.

[39] K. Sharifi and A. Leon-Garcia. “Estimation of shape parameter for generalized Gaussian distributions in subband decompositions of video”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 5.1 (Feb. 1995). Conference Name: IEEE Transactions on Circuits and Systems for Video Technology, pp. 52–56. ISSN: 1558-2205. DOI: 10.1109/76.350779.

[40] H.R. Sheikh and A.C. Bovik. “Image information and visual quality”. In: *IEEE Transactions on Image Processing* 15.2 (Feb. 2006). Conference Name: IEEE Transactions on Image Processing, pp. 430–444. ISSN: 1941-0042. DOI: 10.1109/TIP.2005.859378.

[41] Honglei Su et al. “Perceptual Quality Assessment of 3d Point Clouds”. In: *2019 IEEE International Conference on Image Processing (ICIP)*. ISSN: 2381-8549. Sept. 2019, pp. 3182–3186. DOI: 10.1109/ICIP.2019.8803298.

[42] Honglei Su et al. *Perceptual Quality Assessment of Colored 3D Point Clouds*. Number: arXiv:2111.05474 arXiv:2111.05474 [eess]. Nov. 2021.

[43] Dong Tian et al. “Geometric distortion metrics for point cloud compression”. In: *2017 IEEE International Conference on Image Processing (ICIP)*. ISSN: 2381-8549. Sept. 2017, pp. 3460–3464. DOI: 10.1109/ICIP.2017.8296925.

[44] Irene Viola and Pablo Cesar. “A Reduced Reference Metric for Visual Quality Evaluation of Point Cloud Contents”. In: *IEEE Signal Processing Letters* 27 (2020). Conference Name: IEEE Signal Processing Letters, pp. 1660–1664. ISSN: 1558-2361. DOI: 10.1109/LSP.2020.3024065.

[45] Irene Viola, Shishir Subramanyam, and Pablo Cesar. “A Color-Based Objective Quality Metric for Point Cloud Contents”. In: *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. ISSN: 2472-7814. May 2020, pp. 1–6. DOI: 10.1109/QoMEX48832.2020.9123089.

[46] Jianqiang Wang et al. “Learned Point Cloud Geometry Compression”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 31.12 (Dec. 2021). arXiv:1909.12037 [cs, eess], pp. 4909–4923. ISSN: 1051-8215, 1558-2205. DOI: 10.1109/TCSVT.2021.3051377.

[47] Martin Weinmann, Boris Jutzi, and Clément Mallet. “Feature relevance assessment for the semantic interpretation of 3D point cloud data”. In: *ISPRS Workshop Laser Scanning 2013. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. II-5/W2* (Nov. 2013), pp. 313–318. DOI: 10.5194/isprsannals-II-5-W2-313-2013.

[48] Xinju Wu et al. “Subjective Quality Database and Objective Study of Compressed Point Clouds With 6DoF Head-Mounted Display”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 31.12 (Dec. 2021). Conference Name: IEEE Transactions on Circuits and Systems for Video Technology, pp. 4630–4644. ISSN: 1558-2205. DOI: 10.1109/TCSVT.2021.3101484.

[49] Yiling Xu et al. “EPES: Point Cloud Quality Modeling Using Elastic Potential Energy Similarity”. In: *IEEE Transactions on Broadcasting* 68.1 (Mar. 2022). Conference Name: IEEE Transactions on Broadcasting, pp. 33–42. ISSN: 1557-9611. DOI: 10.1109/TBC.2021.3114510.

[50] Qi Yang et al. *Inferring Point Cloud Quality via Graph Similarity*. arXiv:2006.00497 [cs, eess]. Dec. 2020.

[51] Qi Yang et al. “No-Reference Point Cloud Quality Assessment via Domain Adaptation”. en. In: 2022, pp. 21179–21188.

[52] Qi Yang et al. “Point Cloud Distortion Quantification based on Potential Energy for Human and Machine Perception”. In: *arXiv:2103.02850 [cs, eess]* (July 2021). arXiv: 2103.02850.

[53] Qi Yang et al. “Predicting the Perceptual Quality of Point Cloud: A 3D-to-2D Projection-Based Exploration”. In: *IEEE Transactions on Multimedia* 23 (2021). Conference Name: IEEE Transactions on Multimedia, pp. 3877–3891. ISSN: 1941-0077. DOI: 10.1109/TMM.2020.3033117.

[54] Emin Zerman et al. “Subjective and Objective Quality Assessment for Volumetric Video Compression”. In: vol. 2019. Jan. 2019. DOI: 10.2352/ISSN.2470-1173.2019.10.IQSP-323.

[55] Emin Zerman et al. “Textured Mesh vs Coloured Point Cloud: A Subjective Study for Volumetric Video Compression”. In: *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. ISSN: 2472-7814. May 2020, pp. 1–6. DOI: 10.1109/QoMEX48832.2020.9123137.

[56] Yujie Zhang, Qi Yang, and Yiling Xu. “MS-GraphSIM: Inferring Point Cloud Quality via Multiscale Graph Similarity”. en. In: *Proceedings of the 29th ACM International Conference on Multimedia*. Virtual Event China: ACM, Oct. 2021, pp. 1230–1238. ISBN: 978-1-4503-8651-7. DOI: 10.1145/3474085.3475294.

[57] Zicheng Zhang et al. “No-Reference Quality Assessment for 3D Colored Point Cloud and Mesh Models”. In: *IEEE Transactions on Circuits and Systems for Video Technology* (2022). arXiv:2107.02041 [cs, eess], pp. 1–1. ISSN: 1051-8215, 1558-2205. DOI: 10.1109/TCSVT.2022.3186894.

A

Distortion correlation

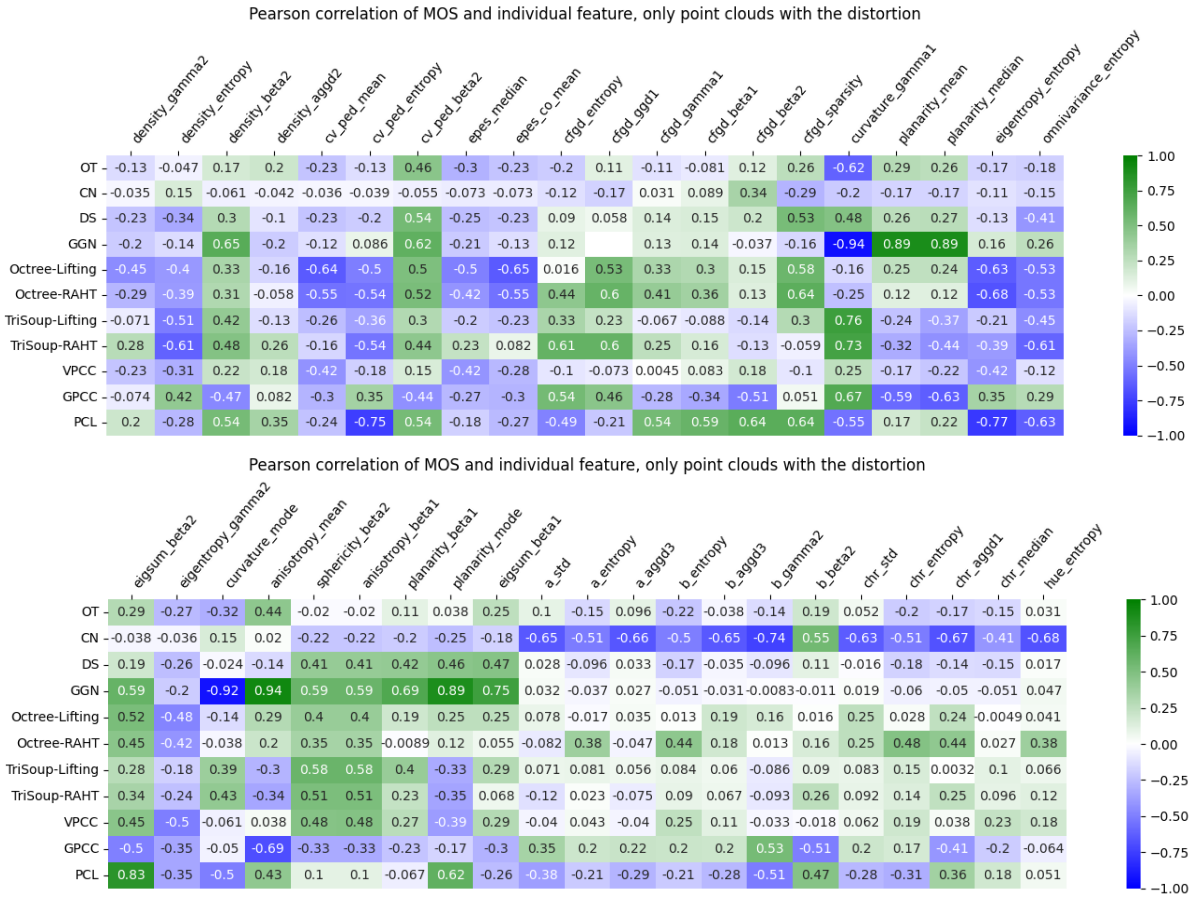


Figure A.1: Heatmap of the Pearson correlation of each feature (x-axis) for a subset with distortion label (y-axis) with the MOS value.

B

Final feature sets used in experiment

Proposed feature set	(Section 5.1)	**Distortion study set**	(Section 5.2)
anisotropy-beta1		anisotropy-aggd1	anisotropy-beta1
anisotropy-mean		anisotropy-beta2	
curvature-gamma1	curvature-mode	curvature-gamma1	curvature-gamma2
eigentropy-entropy		eigentropy-aggd1	eigentropy-beta1
eigentropy-gamma2		eigentropy-ggd1	
eigsum-beta1	eigsum-beta2	eigsum-aggd1	eigsum-aggd4
planarity-beta1	planarity-mean	planarity-energy	planarity-mean
planarity-median	planarity-mode		
sphericity-beta2		sphericity-aggd1	sphericity-aggd3
		sphericity-energy	sphericity-gamma1
		sphericity-sparsity	
omnivariance-entropy		linearity-gamma1	linearity-sparsity
		variation-aggd1	variation-beta1
		variation-entropy	variation-sparsity
a-aggd3	a-entropy	a-energy	a-entropy
a-std			
b-aggd3	b-beta2		
b-entropy	b-gamma2		
chr-aggd1	chr-entropy	chr-beta2	chr-ggd2
chr-median	chr-std		
hue-entropy		l-sparsity	
cfgd-beta1	cfgd-beta2	cfgd-ggd2	cfgd-median
cfgd-entropy	cfgd-gamma1	cfgd-std	
cfgd-ggd1	cfgd-sparsity		
cv-ped-beta2	cv-ped-entropy	cv-ped-aggd1	cv-ped-aggd2
cv-ped-mean			
epes-co-mean		epes-co-aggd1	epes-co-beta2
		epes-co-sparsity	
epes-median		ped-ggd1	
density-aggd2	density-beta2	density-aggd1	density-aggd4
density-entropy	density-gamma2	density-gamma1	

Table B.1: Feature sets. Ordered to easily compare. For reproduction purposes, the feature sets used in the experiments of section 5.3 are reported here.