KU LEUVEN

FACULTEIT INDUSTRIËLE
INGENIEURSWETENSCHAPPEN

CAMPUS GROEP T LEUVEN

MMSIN: A Hybrid No-Reference Point Cloud Quality Assessment Framework

Bridging Natural Scene Statistics and Image Features

Michiel CREEMERS

Jens COOSEMANS

Promotor(en): Maria Torres Vega
Co-promotor(en): Jit Chatterjee

Masterproef ingediend tot het behalen van de graad van Master of Science in de industriële wetenschappen: electronica-ICT

Academiejaar 2023 - 2024

KU LEUVEN

Faculteit industriële ingenieurswetenschappen Campus GROEP T Leuven — Masterproef ingediend tot het behalen van de graad van Master of Science in de industriële wetenschappen: electronica-ICT
Academiejaar 2023 - 2024

MMSIN: A Hybrid No-Reference Point Cloud Quality Assessment Framework

Bridging Natural Scene Statistics and Image Features

Coosemans Jens, Creemers Michiel

Master in Electronics and ICT , Faculteit industriële ingenieurswetenschappen, Campus GROEP T Leuven, Andreas Vesaliusstraat 13, 3000 Leuven, België

Promotor(en): Maria Torres Vega

Electronics and ICT , Faculteit industriële ingenieurswetenschappen, Campus GROEP T Leuven, Andreas Vesaliusstraat 13, 3000 Leuven, België, <maria.torresvega@kuleuven.be>

Co-promotor(en): Jit Chatterjee

Electronics and ICT , Faculty of Engineering Technology, Campus GROUP T Leuven, Andreas Vesaliusstraat 13, 3000 Leuven, Belgium, , <jit.chatterjee@kuleuven.be>

SAMENVATTING

Door het toenemende aantal toepassingen voor 3D puntenwolken in diverse technologische velden, zoals extended reality, robotica en volumetrische video, is er nood aan robuuste kwaliteitsnormen. De traditionele methoden vertrouwen vaak op modellen waarbij de gehele puntenwolk gekend is, maar dit is in vele gevallen niet praktisch door de vereiste bandbreedte voor het sturen van puntenwolken. De bestaande referentieloze methoden vergen daarnaast veel middelen waardoor ze niet bruikbaar zijn op apparaten voor consumenten. Deze thesis introduceert MMSIN, een nieuw hybride multimodaal deep learning model om zonder referentie de kwaliteit van een puntenwolk te bepalen. Onze aanpak maakt gebruik van de twee modaliteiten waarin we driedimensionele data bekijken, namelijk 2D en 3D. Het model wordt geëvalueerd op drie verschillende datasets, die elk verschillende referentiepuntenwolken en bijbehorende distorties bevatten, die mogelijke imperfecties simuleren. De resultaten tonen betere nauwkeurigheid in vergelijking met de allernieuwste modellen en zijn daarnaast ook sneller en hebben minder middelen nodig. Het voorgestelde model heeft potentieel om de kwaliteitsbepaling van puntenwolken te verbeteren bij afwezigheid van referentiedata. Hiernaast voorziet ons framework een simpele open source tool voor andere onderzoekers om de kwaliteit van een willekeurige puntenwolk te bepalen met een model naar keuze.

ABSTRACT

The increasing application of 3D point clouds, in various technological fields like extended reality, robotics and volumetric video streaming, require robust quality assessment methods. Traditional approaches often rely on full reference models, which may not be available in many practical scenarios due to the high bandwidth requirements of point cloud data. There exist accurate no-reference models, in which case the reference isn't available. These are memory intensive and it is not feasible to train these models on consumer hardware. This thesis introduces a novel hybrid multi-modal deep learning framework designed for no-reference quality assessment of point clouds, aiming to address this gap. Our approach leverages features extracted from the two main modalities of how we humans can experience 3D data, this being both 2D and 3D. MMSIN uses these features to train a deep learning model capable of predicting quality metrics. The model was evaluated using three point cloud datasets with synthetically introduced distortions to simulate real-world imperfections. Results show better accuracy (SRCC of 0.921 for SJTU) compared to the state of the art, while also greatly reducing memory requirements. We believe that the proposed method could significantly enhance the automation, accuracy and speed of point cloud quality assessment in the absence of references. Moreover, our framework provides an easy-to-use, open source tool for other researchers to assess the quality of a point cloud based on their model.

1 INTRODUCTION

One of the technological fields that is shaping this decade is that of Extended Reality (XR). This field, which encompasses both Virtual-Reality (VR) Augmented reality (AR) and Mixed Reality (MR), allows people to interact with and explore technology like never before. Predictions show that the XR market will keep growing at a Compound Annual Growth Rate (CAGR) of about 30 percent, leading to a market size of 500 billion USD by 2030 [1].

One of the most prominent examples of immersive content is volumetric video, where objects and humans are captured using point clouds and represented in the virtual environments in the form of meshes. 3D Point Clouds are a simple way of storing 3D data by placing points on the surface of an object and storing xyz-coordinates and the colour information.

1.1 Problem Statement

One of the main concerns is to provide the best possible experience to the consumer of the XR content. This can range from lowering the risk of cybersickness, to making sure that the content itself has a high perceived quality. This perceived quality is an important metric, as it gives insight on the effect that possible distortions could have on the consumer's perception. These distortions can occur naturally, for example by poor capturing of the content. However, compression, delivery and rendering can also introduce these distortions into the object. Since it costs a lot of bandwidth to stream uncompressed volumetric video, it can be very interesting to see what compression has little effect on the perceived quality.

For traditional video content, Netflix's Video Multi-Method Assessment Fusion (VMAF) [2] provides a solution to this problem. It has become the standard tool to assess the quality of traditional video content and is also finding its way to volumetric video [3].

In many cases, the 3D content isn't turned into a video sequence. For point-clouds and 3D meshes there exists no equivalent standard tool to assess their quality. Mainly for point clouds, there is a lot left to explore. In general, there are two ways to assess the quality of a point cloud. Full Reference (FR) models can be used where the reference, non-distorted, point cloud is known. On the other hand there are no-reference models (NR), that don't have access to this reference. The state of the art FR models are optimised and due to their limitations in practical scenarios, these methods aren't as interesting. The NR models on the other hand still need to be improved to work towards accurate real time assessment.

1.2 Overview

In this paper, we propose a hybrid multi-modal no-reference model for point cloud assessment. We use data fusion to combine a statistical machine learning model with an image based deep learning model. We will start by exploring the earlier solutions in the related work section, as well as giving more info about what choices had to be made along the process. After this, in the method section, more details will be given about the model itself, and the features that are extracted from the point cloud. Later, the results are shown, as well as what datasets were used to train the model. At the end of this section, we will introduce a simple graphical user interface which makes it easy for an end user to assess the quality of a single point cloud. After that, these results will be discussed. And in the final section, a conclusion is made. The results can be found on the Github Repository `https://github.com/MichielCreemers/MMSIN`

2 RESEARCH OBJECTIVE

The goal of this thesis is to research whether it is possible to develop a NR model to assess the quality of a point cloud. We aim to design and develop a model that can do this on low computation hardware. Many of the existing models are trained on workstation GPU's. This hardware is only available for a select group of people. Recently, VR headsets have been turning into standalone devices with limited resources and power. Because of this, the assessment tool needs to handle these resources with care and should be as efficiently as possible. So the main application, like streaming volumetric video for example, isn't affected. A second goal of point cloud quality assessment is real time assessment. Currently, this is not possible.

For our approach, we took inspiration from the state of the art [4], where we modified its components to make it more efficient. We hypothesise that by using a hybrid approach, combined with deep learning and statistical machine learning, the required resources should drop significantly.

A side-problem is to tackle the lack of an assessment tool. Several models exist, but no easy way to use them exists. To fill in this gap, we make a tool with a graphical user interface that allows researchers and other users to select a pre-trained model that they then can use to assess the quality of their own point cloud content.

3 RELATED WORK

3.1 Point Clouds

Before the quality assessment of point clouds can be discussed, we need to look at what exactly point clouds are, and what their significance is in the field of computer graphics. The origin of point cloud data can be traced back all the way to the 19th century, where the technique of photogrammetry was first put into practice. The goal of photogrammetry is to gather spatial information from an object through images from several perspectives.

Later, the technology to create these point clouds kept improving and in 1985, Marc Levoy and Turner Whitted published their paper called " The Use of Points as a Display Primitive" [5]. In this paper the concept of point cloud data is introduced for the first time. The main motivation to switch over to point clouds, was to split the modelling from the rendering. By splitting this up, making small adjustments to the model becomes easy. Besides, this format made it easier to store the data from LiDAR, and other capturing methods. The proposed format is the following:

$$(x, y, z, r, g, b, \alpha) \tag{3.1}$$

It is a simple tuple that consists of seven attributes, three spatial attributes and four non-spatial attributes. The spatial attributes represents the three cartesian coordinates. The r, g and b represent the RGB colour space. Finaly, α represents the opacity of the point. This format was later standardized by Greg Turk [6] and was called the Polygon File Format or the Stanford Triangle Format. It is still the standard to store point cloud files as of today.

With this standard in place, many new doors were opened in the field of computer graphics. It lead to the creation of many datasets. Datasets like S3DIS [7] are used in robotics and total scene understanding where point clouds are used to represent rooms and spaces. However, they can also be used to represent individual objects, which can then be easily distorted, like the point clouds in the 8i Voxelized Full Bodies dataset [8] as shown in Figure 3.1

Figure 3.1: "soldier" and "loot" point clouds from the 8i Voxelized Full bodies dataset.

3.1.1 Subjective Quality assessment of Point clouds

To assess the quality of a point cloud, and the effect of distortions, there are several approaches to take. The most accurate results can be obtained by using subjective quality metrics. In general, test subjects see two versions of a point cloud. The base point cloud, and a distorted version next to it.

However, the amount of interaction a participant can have with a point cloud can differ. Some studies only allow for passive interaction, where the participant gets shown a prerecorded video containing rendered frames of the point cloud [9] [10]. On the other hand, the user can also be given the possibility to control the object with a mouse [9] [11]. In some cases, test subjects can also view the object through a head mounted display (HMD) [11]. This is also finding its way in the assessment of volumetric video [12]. Both methods are great, but research concluded that giving the person interacting with the point cloud this freedom, can often lead to more distractions, and thus a worse result [9].

All test subjects then have to give the distorted version a score from one to five. This is repeated in random order. The results are averaged to obtain a mean opinion score (MOS). As was mentioned before, this is an excellent way to assess the quality, but the downside of this approach is the overhead coming along with these tests. If a new dataset needs to be assessed, a new group of assessors has to be gathered, and they all have to go through this process again. So the need to predict this MOS based on objective features is undeniable. To calculate these pseudo mean opinion scores, there are once again several approaches.

3.2 Full- vs No-Reference quality assessment

To assess the quality, there are three main strategies, based on how much is known about the point cloud. There are full-reference, reduced-reference (RR) and no-reference metrics.

When choosing for an FR approach, the reference object is known, so it becomes easier to predict the quality. Some of the earliest attempts made use of this approach, and based their model on the geometry of the point cloud [13] [14]. Later, more advanced methods like PointSSIM [15],

PSNR-yuv [16] , PCQM [17], and GraphSIM [18]obtained increasingly better results, which for a long time weren't met by any NR or RR approach. One downside of the FR approach is that the entire reference point cloud needs to be known.

To resolve this problem, RR methods can be created. These methods only need statistical data about the reference object to assess the quality of the distorted object. This data can be purely based of statistical features, like the colour and geometry quantisation steps [19]. In another case, saliency projections provide the reference [20]. These models are great for volumetric video streaming, where the quality of the point cloud can be affected by compression, and it is known what is being sent to the recipient. However, these models don't suffice to predict the quality of objects for which there is no reference at all.

Finally to handle these point clouds without access to a ground truth, NR methods are needed. This can be helpful for example when researchers are coming up with cheaper or new ways to capture a point cloud and they want to assess whether this new method yields good results. The no reference methods have taken some time to come to par with even the worst FR models, but since the increasing capabilities of deep learning, it has become possible to match the state of the art FR methods. However, one of the downsides of these complex machine learning models is the required hardware to run them. So there is still a need to create powerful, yet efficient assessment methods.

3.3 Image based vs Geometry based features

To train a machine learning model, there is of course a need for features of the point cloud. For NR models the main two categories are image- and geometry based features. Some of the latest models however combine multiple modalities to come up with a multi-modal approach.

3.3.1 Geometry based models

The geometry based methods get their features directly from the point cloud itself. This can be in the form of Natural Scene Statics (NSS) [21], where statistical parameters of the point cloud are extracted and put into a machine learning model. Another method called ResSCNN, instead opts to generate the pseudo MOS by generating hierarchical features from the point cloud object before feeding them into a sparse convolutional neural network (CNN) [22].

3.3.2 Image based models

Another popular method finds its origin in the quality assessment of 2D images. For images, there have been done far more studies on the topic of quality assessment [23]. This idea is the basis for several point cloud quality assessment methods. A point cloud renders onto a plane, resulting in an image. Then special decoders generate the different features of the images [24] [25] [26] [10]. In some cases, the model is created through transfer learning. A model like ResNet [27] then continues optimising itself based on the projections of the point cloud [4].

3.3.3 Multimodal models

The state of the art model to perform point cloud quality assessment harnesses the benefits of both the image based features, and the geometry features [4] to create a multi-modal model. The first modality comes from image projections, from which the features are extracted by an implementation of ResNet [28]. The second modality gets its features directly from the point cloud itself, and once again uses transfer learning with PointNet as the base model [29]. Combining these modalities results overall in more features, and thus resulting in better results.

4 METHOD

In this chapter we introduce MMSIN, a novel hybrid multimodal fusion network designed for the NR quality assessment of point clouds. This approach leverages two feature modalities to predict quality metrics. The first being features from the 3D data itself, while the second modality contains features extracted by first projecting a point cloud to a 2D plane. The network is shown in Figure 4.1. Features are extracted from two modalities using both a statistical machine learning modal and a deep learning model. These features are then enhanced through mutual guidance using symmetric cross-modal attention, resulting in a final feature representation consisting both of the original and enhanced features. Ultimately, this feature representation is decoded into a single quality prediction through the quality regression model.

4.1 NSS Machine Learning Model

The features from the first modality are extracted from a statistical machine learning model based on the concept of Natural Scene Statistics (NSS). Natural Scene Statistics refer to the statistical properties and models that describe the characteristics of natural scenes. The application of NSS is based on the premise that natural images

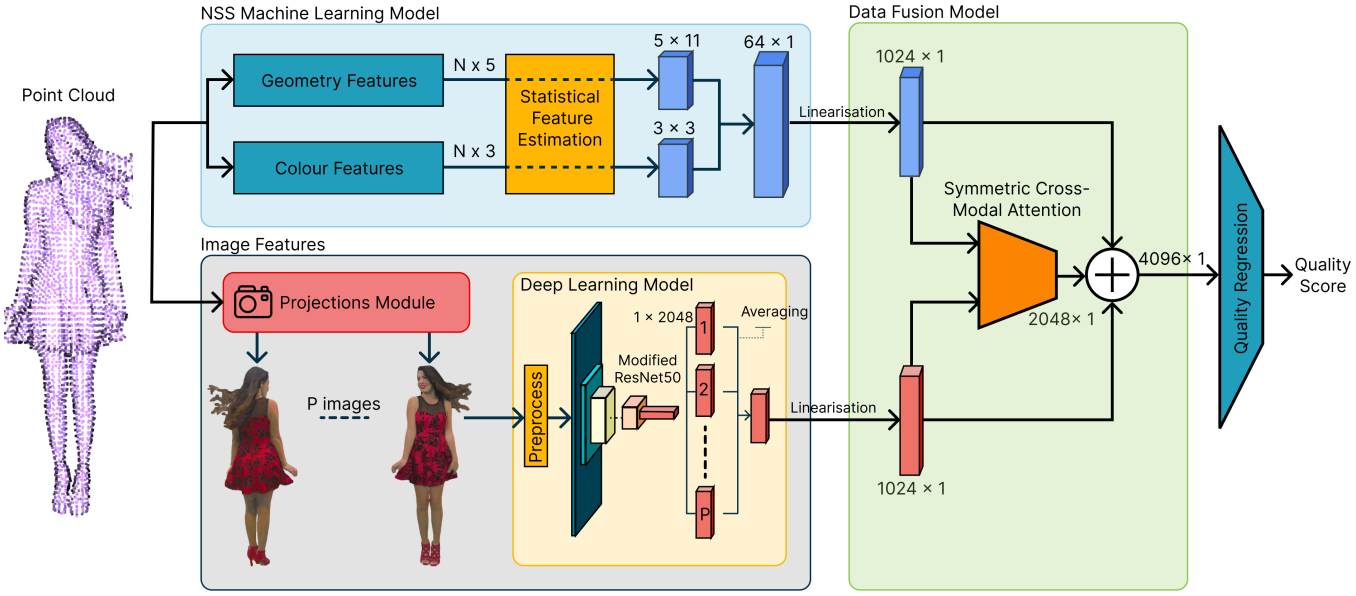

Figure 4.1: End-to-end pipeline of the model

exhibit certain statistical regularities that can be mathematically modelled and utilised for various purposes, one of them being image quality assessment (IQA) [30] [31]. This technique was then pushed further by [21]. Their method uses handcrafted features to estimate statistical parameters within certain NSS distribution to assess the quality of point clouds. It is this method that is used as the first modality. The method can be divided in three parts, a geometry feature projection, a colour feature projection, and a statistical parameter estimation.

4.1.1 Geometry Feature Projection

Since point clouds are inherently unstructured data forms, being a collection of points in space, each point represents a surface element of an object. Though without any explicit connection to its neighbours. The first step is to determine the local neighbourhood for each point in the point cloud. This is accomplished using the k-nearest neighbours algorithm, where the neighbourhood of a given point consists of the closest k points based on Euclidean distance. This step is crucial as it establishes the local context for each point, which is essential for subsequent calculations. The PyntCloud library [32] is used for this computation. Once the neighbourhoods are defined, the covariance matrix $\mathbf{C}_i$ is computed for each point p_i using the following formula:

$$\mathbf{C}_i = \frac{1}{K} \sum_{j=1}^{K} (\mathbf{p}_j - \hat{\mathbf{p}})(\mathbf{p}_j - \hat{\mathbf{p}})^T, \tag{4.1}$$

where $\mathbf{p}_j$ are the points in the neighbourhood of p_i, K is the number of points in the neighbourhood, and $\hat{\mathbf{p}}$ is the centroid of the neighbourhood. This matrix is a measure of the spatial distribution of points around p_i and captures the extent to which these points vary along different axes. The covariance matrix's eigenvalues and eigenvectors are then computed. The eigenvalues– $\lambda_1, \lambda_2, \lambda_3$ –indicate the variance of the points along the axes defined by their corresponding eigenvectors and are instrumental in deriving five key geometric features, as described in [21]:

- **Curvature**: Measured as $\frac{\lambda_3}{\lambda_1+\lambda_2+\lambda_3}$, indicating how much the surface deviates from being flat and can be used as a unit to describe roughness or smoothness.
- **Anisotropy**: $\frac{\lambda_1-\lambda_3}{\lambda_1}$, describing the variance in geometry along different directions.
- **Linearity**: $\frac{\lambda_1-\lambda_2}{\lambda_1}$, assessing the alignment of points to a straight line.
- **Planarity**: $\frac{\lambda_2-\lambda_3}{\lambda_1}$, evaluating how points fit to a plane.
- **Sphericity**: $\frac{\lambda_3}{\lambda_1}$, showing how closely the points form a spherical shape.

Again, the PyntCloud library is used to calculate these features at the point level, thus capturing the local distribution patterns within the point cloud.

4.1.2 Colour Feature Projection

Colour attributes play an important role in the assessment of visual quality for 3D models. In coloured point clouds, each point is characterised by colour information typically encoded in RGB colour space. It is however known for

a long time that the RGB colour space does not correlate well with human colour perception [33] [34]. To address this, the RGB colour values are converted to the CIELAB (LAB) colour space, which better aligns with human visual sensitivity to colour because it was designed to achieve near uniform spacing between perceived colour differences. The LAB colour space expresses colour as three values:

- $L*$ – Lightness from black (0) to white (100).
- $a*$ – Position between red and green, technically an unbounded value but typically kept between an integer range.
- $b*$ – Position between yellow and blue, technically unbounded but kept between an integer range.

This transformation involves first converting RGB values to the XYZ colour space, which serves as a bridge to LAB. The conversion from XYZ to LAB is designed to mimic the nonlinear response of the eye, emphasising the scaling of colour differences based on their perceptual importance. A full mathematical explanation of the transformation can be found in [21]. In our implementation the popular python image processing package scikit-image [35] is used to make the conversion.

4.1.3 Statistical Parameters

Once these eight feature domains are derived, statistical parameter estimation is applied on them. The estimation of these parameters is critical as they allow for a comprehensive assessment of the visual quality based on variations caused by different types of distortions. According to [21] and previous work, the following parameters of NSS models are significantly affected by distortions and are therefore suitable for quality evaluation.

1) Basic Statistical Parameters
First, some basic statistical parameters are computed, namely the mean$(\cdot)$ and standard deviation std$(\cdot)$. Also, it has been shown in [21] that the entropy is highly correlated with quantisation errors that come with the typical compression algorithms. The lower the quantisation levels, the more sparse the distribution becomes. Therefore, the use of entropy can help identifying alterations in the feature distribution. This is clearly perceptible by for example plotting the colour feature domains in histograms as can be seen in Figure 4.2. The specific steps in calculating entropy as a statistical parameter are outlined as follows:
Given a set of values from the feature domains, the probability distribution is estimated by dividing the data into a fixed number of bins. Each bin represents an interval, and the probability p_i for each bin is calculated based on the frequency of data points falling into the respective bin.

$$p_i = \frac{n_i}{N}, \tag{4.2}$$

where n_i is the number of points in the i-th bin and N is the total number of data points. The entropy H of the estimated probability distribution is then calculated using the formula:

$$H = -\sum_{i=1}^{k} p_i \log(p_i), \tag{4.3}$$

where p_i are the probabilities of each bin and k is the total number of bins. The entropy is calculated using `scipy.stats.entropy`.

2) GGD Parameters
Unlike the colour feature domains, where the sparseness of a Gaussian distribution is mainly affected, other types of distortions have more impact on the overall shape of the Gaussian distribution. For example the tail and peak behaviour is strongly affected for the linearity feature when different types of distortion are introduced Figure 4.3(i, j, k, l). Another typical behaviour can be seen when geometric noise is added. One can notice the distributions become smoother as seen in Figure 4.3. The Generalised Gaussian Distribution (GGD) is employed to model this behaviour. The GGD is defined as follows:

$$f(x;\alpha,\beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} \exp\left(-\left(\frac{|x|}{\alpha}\right)^{\beta}\right), \tag{4.4}$$

where $\alpha > 0$ is the scale parameter, $\beta > 0$ is the shape parameter, and the Gamma function $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt, x > 0$. Since the shape parameter β dictates the tail heaviness and peak sharpness, it is the main parameter of interest. The standard deviation is related to both the shape β and scale α as:

$$\sigma = \sqrt{\frac{\alpha^2\Gamma\left(\frac{3}{\beta}\right)}{\Gamma\left(\frac{1}{\beta}\right)}}, \tag{4.5}$$

and is therefore also chosen as a parameter. Estimating the parameters (β,σ) of the GGD is done using the moment-matching based approach proposed in [36], which finds the optimum shape parameter in one step, given the mean, variance, the mean of the absolute values $E[|\mathbf{x}|]$, and the shape parameter γ of a zero-mean generalised Gaussian probability density function because of the following relation:

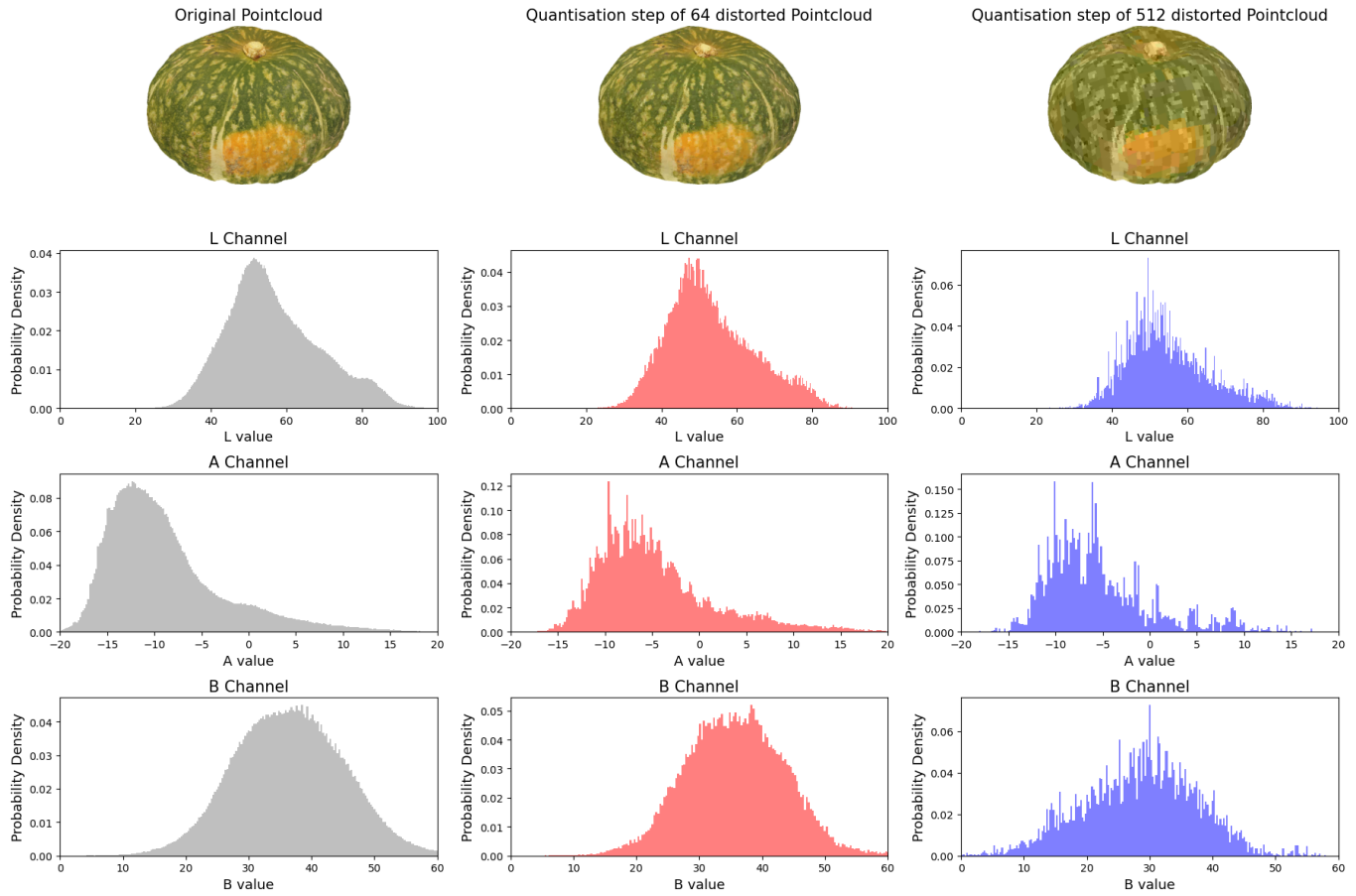

Figure 4.2: The probability density graphs of the LAB colour channels for three point clouds from the WPC dataset. The first is the reference point cloud. The other two are distorted with increasing levels of type $G-PCC(T)/S-PCC$ distortion with quantisation steps of 64 and 512 respectively.

$$r(\gamma) = \frac{\hat{\mu}_{\mathbf{x}}}{E^2[|\mathbf{x}|]} = \frac{\Gamma\left(\frac{1}{\gamma}\right)\Gamma\left(\frac{3}{\gamma}\right)}{\Gamma\left(\frac{2}{\gamma}\right)^2}, \tag{4.6}$$

where $\mathbf{x} \in \{F_{geo}\}$ and $r(\gamma)$ is called the *generalised Gaussian ratio function* as explained in [36].

First, an estimate for the mean $\hat{\mu}_x$ and variance $\hat{\sigma}_x^2$ is determined with:

$$\hat{\mu}_{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad \textit{and} \quad \hat{\sigma}_{\mathbf{x}}^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{\mu}_{\mathbf{x}})^2, \tag{4.7}$$

where N is the total number of data points.

Second, the modified mean of the absolute values is calculated: $\hat{E}[|\mathbf{x}|] = (1/N)\sum_{i=1}^{N}|x_i - \hat{\mu}_{\mathbf{x}}|$.

Then last, the ratio $\rho = \frac{\sigma^2}{E^2[|\mathbf{x}|]}$ is calculated. With all these parameters calculated, the equation $\hat{\gamma} = r^{-1}(\rho)$ can be solved by minimising the difference $|\rho - r(\gamma)|$, resulting in an optimal shape parameter estimation $\hat{\gamma}$.

3) AGGD Parameters

For feature domains that exhibit an asymmetrical distribution due to distortions, for instance in Figure 4.3(b, f, h, l, p, r, t), the Asymmetric Generalised Gaussian Distribution (AGGD) is utilised. It provides a more detailed parameterisation, especially useful for describing feature domains that show different extents of spread in two directions. More specifically, the AGGD with zero mode is used:

$$f(\mathbf{x};\beta,\sigma_l^2,\sigma_r^2) = \begin{cases} \frac{\beta}{(\alpha_l+\alpha_r)\Gamma(1/\beta)}\exp\left(-\left(\frac{-\mathbf{x}}{\alpha_l}\right)^{\beta}\right) & x < 0 \\ \frac{\beta}{(\alpha_l+\alpha_r)\Gamma(1/\beta)}\exp\left(-\left(\frac{\mathbf{x}}{\alpha_r}\right)^{\beta}\right) & x \geq 0, \end{cases} \tag{4.8}$$

where

$$\alpha_l = \sigma_l\sqrt{\frac{\Gamma\left(\frac{1}{\beta}\right)}{\Gamma\left(\frac{3}{\beta}\right)}} \quad \textit{and} \quad \alpha_r = \sigma_r\sqrt{\frac{\Gamma\left(\frac{1}{\beta}\right)}{\Gamma\left(\frac{3}{\beta}\right)}} \tag{4.9}$$

β is the shape parameter that controls the shape while σ_l^2 and σ_r^2 are the scale parameters that control the spread on each side of the distribution. The parameters that best fit the AGGD are $(\eta,\beta,\sigma_l^2,\sigma_r^2)$ where η is a measure of asymmetry defined as:

$$\eta = (\alpha_r - \alpha_l)\frac{\Gamma(2/\beta)}{\Gamma(1/\beta)} \tag{4.10}$$

Table 4.1: Overview of the parameters that are calculated or estimated in the NSS machine learning model

Feature Domain	Feature	Feature Description
F_{col}	μ, σ	Mean and standard deviation
	H	Entropy
F_{geo}	μ, σ	Mean and standard deviation
	H	Entropy
$\hat{F}_{geo}$	$GGD(\beta,\sigma)$	Shape parameter and standard deviation of GGD
	$AGGD(\eta,\beta,\sigma_l^2,\sigma_r^2)$	Asymmetry coefficient, shape and scale parameters of AGGD
	$Gamma(\alpha,\beta)$	Shape and scale parameters of Gamma shape

The parameters described above are then estimated using the moment-matching based approach proposed in [37], where the goal is to estimate γ and β from a theoretical distribution that models the empirical data:

$$r = \frac{(\gamma^2+1)^2}{(\gamma^3+1)(\gamma+1)} \times \rho(\beta) \tag{4.11}$$

where $\gamma = \alpha_l/\alpha_r$ and $\rho(\beta) = \frac{\Gamma(2/\beta)^2}{\Gamma(1/\beta)\Gamma(3/\alpha)}$ is the *generalised Gaussian ratio function*. First, since the AGGD distribution with zero mode is used, the feature domains should be normalised using:

$$\hat{F} = \frac{F - mean(F)}{std(F) - 1} \quad with \quad F \in \{F_{geo}\} \tag{4.12}$$

Then, a theoretical ratio $r(\gamma)$ is defined that models the expected relation between the moments and absolute moments of the distribution:

$$r(\gamma) = \frac{\Gamma\left(\frac{1}{\gamma}\right)\Gamma\left(\frac{3}{\gamma}\right)}{\Gamma\left(\frac{2}{\gamma}\right)^2} \tag{4.13}$$

Second, γ is estimated by dividing the estimates of the left and right standard deviations, $\hat{\sigma}_l$ and $\hat{\sigma}_r$ respectively:

$$\hat{\gamma} = \frac{\hat{\sigma}_l}{\hat{\sigma}_r} = \frac{\sqrt{\frac{1}{M_l-1}\sum_{\substack{k=1 \\ x_k<0}}^{M_l} x_k^2}}{\sqrt{\frac{1}{M_r-1}\sum_{\substack{k=1 \\ x_k\geq 0}}^{M_r} x_k^2}} \tag{4.14}$$

where M_l are the number of samples on the left side and M_r the number of samples on the right side. $\hat{\gamma}$ then provides an initial estimate of the distribution's asymmetry. Third, an unbiased estimate for r is:

$$\hat{r} = \frac{\left[\sum_{k=1}^{M} |x_k|\right]^2}{\sum_{k=1}^{M} x_k^2} \tag{4.15}$$

Finally, with all these required estimations calculated, the AGGD parameters are estimated using the algorithms from [37]:

1-calculate $\hat{R}$ using the $\hat{\gamma}$ and $\hat{r}$ estimations from equations 4.14 and 4.15:

$$\hat{R} = \hat{r} \times \frac{(\hat{\gamma}^3+1)(\hat{\gamma}+1)}{(\hat{\gamma}^2+1)^2} \tag{4.16}$$

2-According to the $\hat{R}$ value, β is estimated using the approximation of the inverse generalised Gaussian ratio [38]:

$$\hat{\beta} = \hat{\rho}^{-1}(\hat{R}) \tag{4.17}$$

by minimising the squared difference between $r(\gamma)$ and $\hat{R}$.
3-finally the scale parameter η is estimated:

$$\eta = (\hat{\sigma}_r^2 - \hat{\sigma}_l^2) \times \frac{\Gamma(2/\hat{\beta})}{\Gamma(1/\hat{\beta})} \times \sqrt{\frac{\Gamma(3/\hat{\beta})}{\Gamma(1/\hat{\beta})}} \tag{4.18}$$

4) Gamma Parameters
Some features resemble a Gamma shape, primarily seen in curvature and anisotropy distributions, but for some types of distortion also in sphericity [21]. This is shown in for example Figure 4.3(c, g, s). To model this, the parameters (α,β) are estimated in a shape-rate parameterised Gamma probability density function:

$$f(x;\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \tag{4.19}$$

$$with \quad \mathbf{x} > 0 \quad and \quad x \in \{\hat{F}_{geo}\},$$

where α is the shape parameter and β the rate parameter. The parameters are estimated using the *Method of Moments* method by first calculating the sample mean and sample variance using for example equation 4.7. Then, the theoretical moments $\bar{X} = \frac{\alpha}{\beta}$ and $S^2 = \frac{\alpha}{\beta^2}$ can be solved for $\hat{\alpha}$ and $\hat{\beta}$, resulting in:

$$\hat{\alpha} = \frac{\bar{X}^2}{S^2} \quad and \quad \hat{\beta} = \frac{\bar{X}}{S^2} \tag{4.20}$$

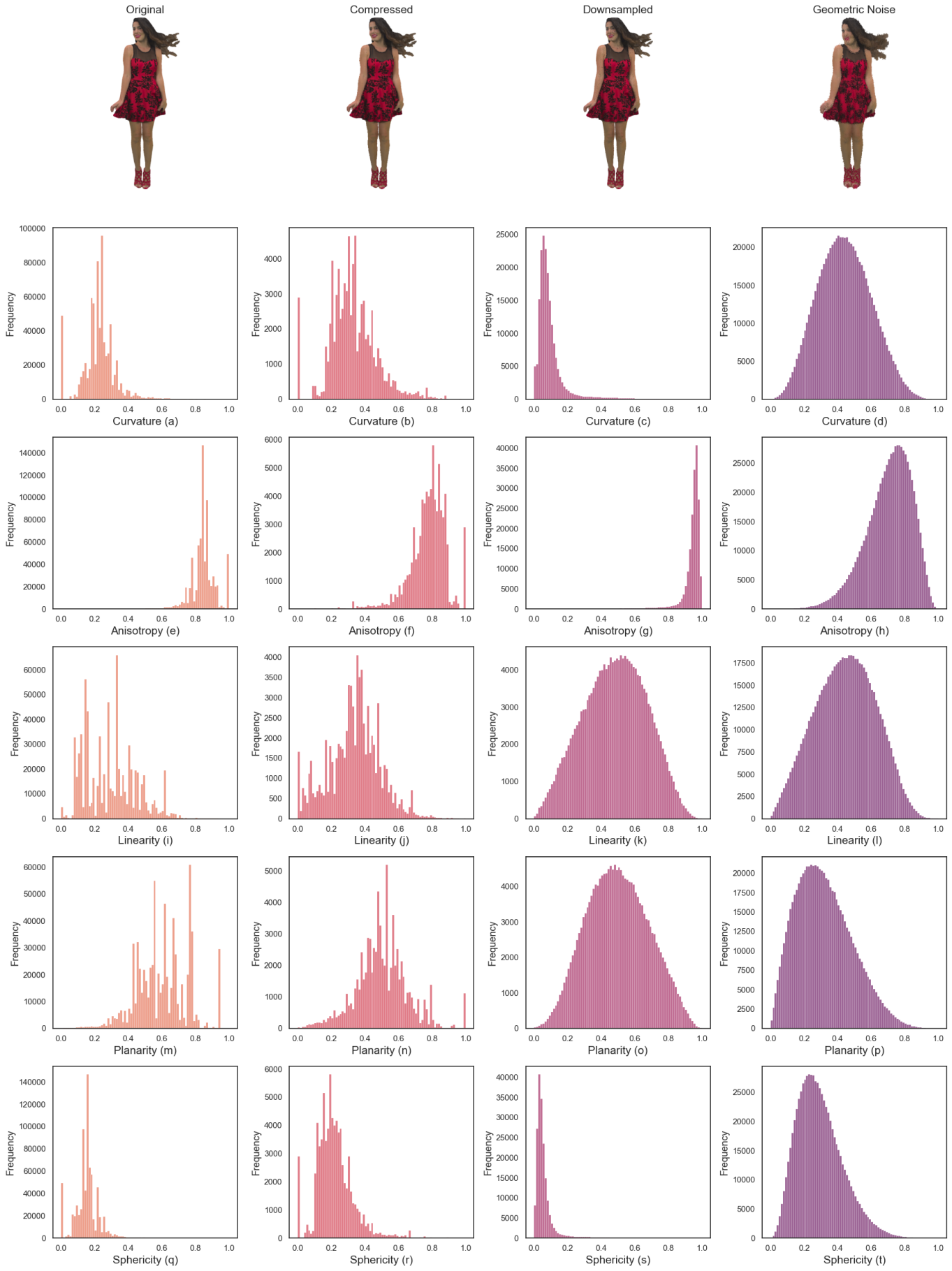

Figure 4.3: Probability distributions of the features from the geometry feature domain F_{geo} for four distorted versions of a point cloud from the SJTU database. A reference point cloud, one with octree-based compression, one downsampled, and a point cloud with geometry Gaussian noise applied.

Using the algorithms explained above, for each point cloud a total of 64 features are calculated. A total of 55 features (5×11) for the features *(Curvature, Anisotropy, Linearity, Planarity, Sphericity)* in the geometry domain, and 9 features (3×3) for the features *(L, a, b)* in the colour domain. An overview is shown in Table 4.1. Finally, a Min-Max scaler based on all the NSS features in the dataset is applied as normalisation, resulting in a single feature vector for a given point cloud:

$$F_{NSS} \in \mathbb{R}^{64\times 1} \tag{4.21}$$

4.2 Image Features

As explained in section 3.3, and due to the current state-of-the art IQA methods, taking a 2D approach has many benefits. The 2D data flow in our model consists of two main blocks, a projections module to transform a 3D point cloud into a series of 2D images, and a deep learning block that extracts features from these images.

4.2.1 Projection Module

For the generation of projections, the open-source library `Open3D` [39] is utilised. A custom function was made to enable the definition of a flexible path around a point cloud with a fixed viewing distance, facilitating the generation of P projections from various angles and viewpoints. This function allows for precise control, but also gives the freedom to experiment with different settings such as the number of projections P, different viewpoints, angles etc. A visualisation of the process is shown in Figure 4.4.

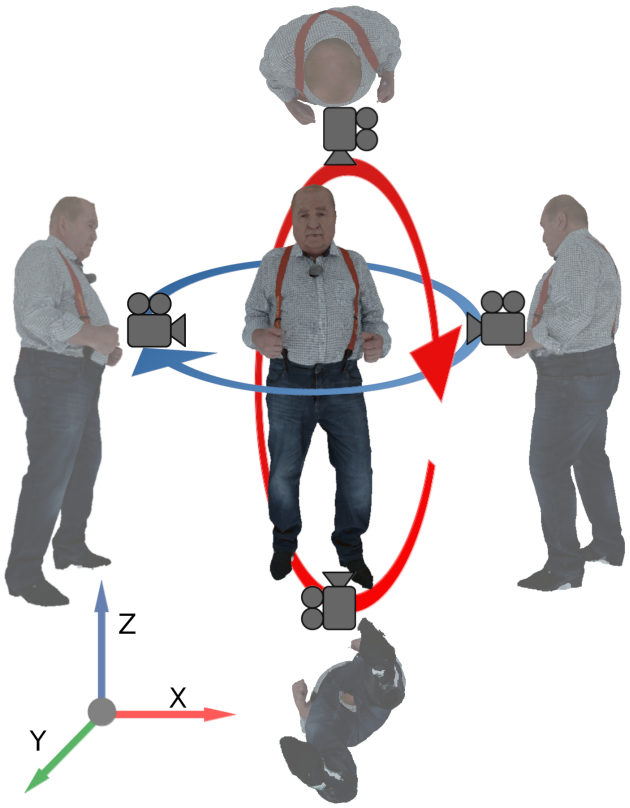

Figure 4.4: Example of the projection generation. A total of four projections are made, two along the z-axis, and two along the x-axis.

4.2.2 Deep Learning Model

To extract features from the P projections, the famous deep learning model ResNet50 [40] for image recognition is used. Although the model is mainly used for image recognition tasks, it can be used for feature extraction. In standard ResNet architectures used for classification, the final layer is a fully connected layer that maps the extracted features to the number of target classes. By omitting this layer, the model directly outputs the global feature representation extracted from the average pool layer, which can be used as a high-level representation of the input images, and thereby making it a valuable feature set for further analysis. For a specific point cloud, each of the P projections goes through the modified ResNet50 model, resulting in P feature vectors:

$$F_I^p \in \mathbb{R}^{C_I\times 1} \tag{4.22}$$

where $p \in 1, 2, ..., 9$ is the p-th projection and C_I the number of extracted features. Then lastly, the P image feature vectors are averaged resulting in a single feature vector F_I for a given point cloud with length C_I.

4.3 Data Fusion

To adequately assess the quality of a point cloud using the multi-modal feature vectors $F_{NSS} \in \mathbb{R}^{64\times 1}$ and $F_I \in \mathbb{R}^{C_I\times 1}$ extracted from prior models described in 4.1 and 4.2, a symmetric cross-modal attention mechanism is incorporated drawing on the methodology proposed in [4], [41].

First, the multi-modal features are linearly projected to a common dimensional space using learnable linear transformations:

$$\hat{F}_{NSS} = W_{NSS}F_{NSS} \quad and \quad \hat{F}_I = W_I F_I, \tag{4.23}$$

where W_{NSS} and W_I are the projection matrices and $\hat{F}_{NSS}, \hat{F}_I \in \mathbb{R}^{C'\times 1}$ represent the dimensionally adjusted features.

The next stage involves a multi-head attention mechanism, as illustrated in Figure 4.5, that enables the model to focus on informative parts of the data from both modalities

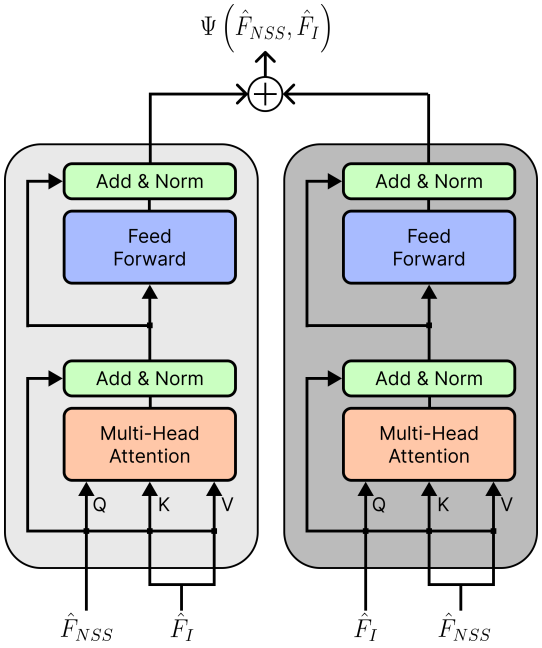

Figure 4.5: Illustration of the symmetric cross-modal attention (SMCA). The multi-modal feature vectors $\hat{F}_{NSS}$ and $\hat{F}_I$ are used to guide each other in the attention learning process.

simultaneously:

$$\begin{aligned} \Gamma(Q,K,V) &= (h_1 \oplus h_2 \oplus ... \oplus h_n)W, \\ h_\mu &= \beta(QW_\mu^Q, KW_\mu^K, VW_\mu^V) \quad \text{for} \quad \mu = 1,\ldots,n, \\ \beta(Q,K,V) &= \texttt{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \end{aligned} \tag{4.24}$$

where $\Gamma(\cdot)$ denotes the multi-head attention function, $\beta(\cdot)$ is the attention calculation, h_μ represents the μ-th attention head, and W, W^Q, W^K, W^V are learnable linear mappings for transforming the inputs into queries, keys, and values.

By leveraging symmetric cross-modal attention, the features from both modalities enhance each other through mutual guidance, resulting in a final feature vector that is derived by concatenating the original feature vectors $\hat{F}_{NSS}$ and $\hat{F}_I$ with the multi-modal feature vector obtained from the attention part:

$$\hat{F}_Q = \hat{F}_{NSS} \oplus \hat{F}_I \oplus \Psi\left(\hat{F}_{NSS}, \hat{F}_I\right), \tag{4.25}$$

where $\oplus(\cdot)$ represents concatenation, and $\Psi(\cdot)$ is the operation performed by the symmetric cross-modal attention module, resulting in the attention-augmented feature representation $\hat{F}_Q \in \mathbb{R}^{4C' \times 1}$, encapsulating enriched information across modalities.

4.4 Loss function

Finally, a loss function is needed. The loss function gives an indication of how well the model fits the reality by calculating the error between what is outputted, and what the output should be. There exist several different loss functions, each having their own benefits. In this model, a combination of both the L2 loss and a ranking loss is used.

The L2 loss, also called the Mean Squared Error loss, is calculated first. For all samples in the training dataset, the actual MOS is subtracted from the predicted MOS and this result is then squared. This is repeated for all entries in the training set, and the results are averaged. This results in the following formula

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \tag{4.26}$$

Where n is the number of samples in the training dataset, x_i is the actual MOS and $\hat{x}_i$ is the predicted MOS.

The ranking loss is based off a model proposed by Sun et al [42]. It uses pairwise comparison to see if two different test samples are ranked the same for both the predicted and actual MOS. If this is not the case, the loss increases. The model also has a threshold built in, this makes sure that very small deviations aren't penalised.

Finally, the two losses are added together to obtain one value. The weight of each of the losses can also be adjusted.

5 RESULTS

5.1 Dataset Description

To assess the performance of the point cloud quality assessment model, a dataset with reference point clouds and their distorted versions is needed together with their MOS. An important metric is the effectiveness of the quality assessment on a dataset the model wasn't trained on. This might seem trivial, but contrary to 2D images, there are several possible ways to construct a point cloud. The most traditional ones are LiDAR and photogrammetry, but also the use of generative AI will probably become an option in the near future [43]. The dataset constructed by Liu et al. [22] currently has the biggest collection of reference point clouds containing 104 reference point clouds and a total of 22000 distorted versions. However, due to its size this dataset becomes rather unpractical to run on consumer level hardware. Luckily, there exist several other datasets having a good balance between the amount of variation, and total size of the dataset. These datasets have been used in many research papers, so the proposed model can be easily compared based on the results on these datasets.

5.1.1 SJTU-PCQA

Before creating the LS-PCQA dataset, the same researchers created a smaller, yet widely used dataset called the SJTU-PCQA dataset [26]. This dataset consists of ten reference point clouds, of which one is used to train the participants for the evaluation. The dataset consists of five point clouds clouds representing people, and four inanimate objects. These references originate from different datasets. Six different transformations are applied, with each having seven levels of distortion. This results in 378 distorted point clouds. To link these point clouds to a MOS, an interactive assessment method is used where the assessors see the reference point-cloud next to the distorted version and can interact with the rotation of the objects. The six applied distortions are the following:

- **Octree-based compression:** The point cloud is compressed by using octree pruning. This technique splits the point cloud in leaf nodes. The pruning algorithm then starts removing these leaf nodes resulting in a compressed model.
- **Color Noise:** A noise value of scaling proportion is added to the RGB values of a set of the points in the point cloud.
- **Downscaling:** A random set of points get dropped from the point cloud.
- **Geometry Gaussian noise:** Gaussian noise is added to the point cloud which shifts the location to each point slightly.
- **Downscaling and color noise:** Combination of the downscaling of the number of points, and the noise added to the RGB values.
- **Downscaling and Geometry Gaussian noise:** Combination of downscaling of the number of points, and addition of Gaussian noise to the location of the points/
- **Color noise and Geometry noise** Addition to noise, both on the RGB values and the xyz-coordinates of the points.

5.1.2 WPC

Another well known dataset for point cloud quality assessment is the Waterloo Point Cloud dataset (WPC)[9]. The dataset also depicts inanimate objects like foods, and other objects. Compared to the SJTU-PCQA dataset, the objects are smaller, but they are all created under the same conditions. A camera array takes several images

Figure 5.1: Example of a point cloud from the WPC dataset with V-PCC distortion

of the object from different perspectives in a controlled environment. Then, image alignment, sparse point cloud reconstruction, dense point cloud reconstruction, and point cloud merging are used to generate the final point cloud. The point clouds gets normalised, and several distortions make the dataset ready for quality assessment. This is done to twenty different objects, and each object has 37 distortions, resulting in a dataset of 740 point clouds.

Two of the distortions are similar to the SJTU-PCQA 5.1.1 dataset. The first one is octree-based downsampling. In this dataset, there are only 3 distortion levels, created with CloudCompare [44]. The other similar technique is the Gaussian noise. This is added to both the geometry and texture elements. The levels of noise for the geometry are $\{0,2,4\}$, for the texture elements the levels are $\{8,16,32\}$ [45]. The other distortions are based on standardised point cloud distortions approved by MPEG. The distortions are G-PCC[46], L-PCC [47] and V-PCC[48]. An example distorted version of a PC is shown in Figure 5.1.

This dataset is also different from the SJTU-PCQA because of how the MOS is computed. Instead of going for interactive assessment, the test subjects only see a short ten second video of both the reference and distorted point cloud rotating which are placed next to each other on a calibrated monitor. This passive assessment shows more consistent results [9].

5.1.3 WPC2.0

The last PCQA dataset is the WPC2.0 dataset. This dataset was part of a research paper proposing a RR point cloud quality assessment model [19]. This dataset takes sixteen reference point clouds from 5.1.2. However, the distortions are different. All different distortions are now

based on the V-PCC test model v7 [49]. There are five levels of geometry distortions, and 5 levels of color distortions. These are all combined with each other to get a total of 25 different distorted versions per reference point cloud, resulting in a dataset containing a total of 400 point clouds. The MOS are generated in the exact same way as for the WPC dataset.

5.2 Evaluation Criteria

The performance is determined based on four criteria: Pearson's Linear Correlation Coefficient (PLCC), Spearman's Rank Correlation Coefficient (SRCC), Kendall's Rank Correlation Coefficient (KRCC) and the Root Mean Square Error (RMSE). These are all commonly used metrics. The first three tests are non-parametric hypothesis tests. This means that they try to limit the amount of assumption about the distribution of the data.

5.2.1 Pearsons's Linear Correlation Coefficient

The first metric is Pearson's Linear Correlation Coefficient [50], often referred to as Pearson's R. It measures the linear correlation between two sets of data. To get this coefficient, the covariance of the parameters is calculated, which is then divided by the product of their standard deviations. The formula is the following:

$$PLCC = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \tag{5.1}$$

Where x_i and y_i represent the individual samples, and $\bar{x_i}$ and $\bar{y_i}$ represent the mean values. Its value always lies between minus and plus one and approaches one in the case of a perfect correlation.

5.2.2 Spearman's Rank Correlation Coefficient

Second, there is Spearman's Rank Correlation Coefficient [51] often depicted with the Greek letter ρ. It is related to the PLCC, but instead it focuses on the monoticity of the results by ranking all individual samples. The SRCC can be calculated using the same formula as the PLCC, but instead of the sample value, it uses the ranking of the data data. This comes down to the following formula:

$$\rho = \frac{\sum(R(x_i) - R(\bar{x}))(R(y_i) - R(\bar{y}))}{\sqrt{\sum(R(y_i) - R(\bar{x}))^2 \sum(R(y_i) - R(\bar{y}))^2}} \tag{5.2}$$

Where $R(x_i)$ and $R(y_i)$ represent the rank of the values to compare, $R(\bar{x})$ and $R(\bar{y})$ represent the mean of the actual MOS and the mean of the calculated MOS.

5.2.3 Kendall's Rank Correlation Coefficient

Next, there is Kendall's Rank Correlation Coefficient [52], also called the Kendall's τ coefficient. For this test, the b variant is used. This method compares two datapoints with each other. Then it assesses whether they are ranked correctly against each other. If they are in the same order, they are called concordant pairs, if not, they are called discordant. The overall τ score then gets calculated with this formula:

$$\tau_b = \frac{C - D}{\sqrt{(C + D + T_x)(C + D + T_y)}} \tag{5.3}$$

Where C is the amount of Concordant pairs, D is is the number of discordant pairs. T_x and T_y represent the number of ties, the number of values that are the exact same. A value of one represents complete similarity, a value of minus one complete dissimilarity and a value of 0 represents no association.

5.2.4 Root Mean Squared Error

Finally, there is the Root Mean Square error, which is commonly used in various scientific fields to measure the difference between real and predicted values. It is very similar to the MSE used in the loss function 4.4. It calculates the difference for all data points, and squares them. These values are then averaged and added together. Then, the square root of this value is calculated. This results in the following formula:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5.4}$$

Where n is the number of validation samples, y_i represents the actual quality score, and $\hat{y}_i$ represents the predicted quality score. To have a perfect model, this value would have to be 0.

5.3 Implementation Details

Because of the difference in datasets, different models with slightly varying hyperparameters are trained. The optimiser is a standard Adam optimizer with a learning rate and decay rate of 0.001 and 0.0001 respectively. Every eight epochs the learning rate is lowered by ten percent. The ranking loss function also stays the same for all datasets, and it is the L2 ranking loss as described in

Table 5.1: Results of the quality prediction for several NR and FR PCQA methods. The best FR results are highlighted in green, the best NR results are highlighted in Red and the second best NR results are highlighted in blue

Type	Modality	Model	SJTU-PCQA				WPC				WPC2.0			
			SRCC	PLCC	KRCC	RMSE	SRCC	PLCC	KRCC	RMSE	SRCC	PLCC	KRCC	RMSE
FR	G	HD-p2pl	0.6277	0.5940	0.4825	2.2815	0.3281	0.2695	0.2249	22.8226	0.4136	0.4104	0.2965	21.0400
	G	HD-p2po	0.7157	0.7753	0.5447	1.4475	0.2786	0.3972	0.1943	20.8990	0.3587	0.4561	0.2641	18.8976
	G	PointSSIM	0.6867	0.7136	0.4964	1.7001	0.4542	0.4667	0.3278	20.2733	0.4810	0.4705	0.2978	19.3917
	G	PSNR-yuv	0.7950	0.8170	0.6196	1.3151	0.4493	0.5304	0.3198	19.3119	0.3732	0.3557	0.2277	20.14665
	G	PCQM	0.8644	0.8853	0.7086	1.0862	0.7434	0.7499	0.5601	15.1639	0.6825	0.6923	0.4929	15.6314
NR	I	BRISQUE	0.3975	0.4214	0.2966	2.0937	0.2614	0.3155	0.2088	21.1736	0.0820	0.3353	0.0487	21.6679
	I	IT-PCQA	0.63	0.58	-	-	0.54	0.55	-	-	-	-	-	-
	G	ResSCNN	0.86	0.81	-	-	-	-	-	-	0.75	0.72	-	-
	G	3D-NSS	0.7144	0.7382	0.5174	1.7686	0.6479	0.6514	0.4417	16.5716	0.5077	0.5699	0.3638	17.7219
	I	PQA-net	0.8372	0.8586	0.6304	1.0719	0.7026	0.7122	0.4939	15.0812	0.6191	0.6426	0.4606	16.9756
	G+I	MM-PCQA	0.9103	0.9226	0.7838	0.7716	0.8414	0.8556	0.6513	12.3506	0.8023	0.8024	0.6202	13.4289
	G+I	**MMSIN**	**0.9210**	**0.9401**	**0.7678**	**0.8196**	**0.9281**	**0.9270**	**0.7699**	**8.54**	**0.9288**	**0.9360**	**0.7634**	**7.6304**

4.4. The models use five-fold cross validation to ensure that there is a split of 80% train data and 20% validation data. As a standard, six projections of the point cloud are taken. The SJTU and WPC2.0 dataset use a batch size of five and are trained for 100 epochs. The WPC dataset also uses a batch size of five but it trains for a total of 600 epochs. For the cross modal attention, 2048 image features and 64 NSS features get mapped to 1024 features each. The feedforward network consists of two linear layers, with a ReLU activation function and a dropout of 0.1. For the image features, ResNet50 is used as a backbone.

The model is trained on a laptop with an Nvidia GeForce RTX 4060 Mobile GPU which has a total of 8GB of GDDR6 VRAM.

5.4 Competitor Analysis

A great place to start when assessing the performance of a new model is to see how well it fares against the state of the art, and other similar models. The model gets compared against five FR and six NR approaches. The FR models are HD-p2pl [13], HD-p2po [14], PointSSIM [15], PSNR-yuv [16] and PCQM [17]. These FR metrics are all based on the geometry modality, which means that they assess the quality based on the points themselves. The first NR method it gets compared to is BRISQUE [23]. Even though this model initially wasn't meant for point cloud quality assessment, this model forms the basis for the image encoding used in later models. To assess the quality, the quality of the projections is assessed. The other models however are dedicated towards PCQA and are, IT-PCQA [24], ResSCNN [22], 3D-NSS [21], PQA-net [25] and MM-PCQA. These models are based on image features, geometry features, or both. The models are evaluated using the criteria as described in section 5.2 and this is repeated for the three main datasets that were introduced in 5.1. An overview is given in Table 5.1. The results themselves are based on the results described in the MM-PCQA paper [4].

In the table, some results have a different colour. The Best FR model is highlighted in green. The second best No reference model is represented in blue, and the best results for a NR model are highlighted in red. Finally, the model represented in this paper is also displayed in bold.

From Table 5.1, it becomes clear that overall, the quality assessment for FR models perform better than the NR models, even though they were proposed several years earlier. The best FR model has an SRCC of 0.8644 for the SJTU-PCQA dataset and has only recently been surpassed by MM-PCQA, with a SRCC of 0.9103. This model has had the best results. However, with the introduction of this new hybrid multi-modal model, The SROCC was able to improve to 0.9210. This is also true across other datasets and for other criteria where this new model slightly outperforms other methods. Only for the SJTU-PCQA dataset, the MM-PCQA dataset has a slightly better performance in terms of KRCC and RMSE. For several earlier models, it is clearly visible that there is a big difference in performance on the WPC and WPC2.0 dataset compared to the SJTU-PCQA dataset. There is a difference of 0.3 for some of the earliest FR and NR models. Even the MM-PCQA model has a difference of 0.1 between the SJTU-PCQA dataset and the WPC2.0 dataset. However, for the MMSIN model, this discrepancy is hardly noticeable.

5.5 Cross Dataset Evaluation

It can also be interesting to assess the performance of the model on a different dataset. For this test, there are two variations. One test sees how well a model, trained on the WPC dataset performs on the SJTU-PCQA dataset, while the other looks at the performance on the WPC2.0 dataset. The results are shown in table 5.2

Table 5.2: Cross dataset evaluation. Shows how well the model performs on a dataset that is not related to the one it was trained on. The best results are displayed in red, the second best in blue.

Model	WPC → SJTU		WPC →WPC2.0	
	SRCC	PLCC	SRCC	PLCC
PQA-net	0.5411	0.6102	0.6006	0.6377
3D-NSS	0.1817	0.2344	0.4933	0.5613
MM-PCQA	0.7693	0.7779	0.7607	0.7753
MMSIN	**0.4933**	**0.5121**	**0.7803**	**0.7876**

The results for the assessment of the SJTU dataset with a model trained on the WPC dataset are worse compared to the results of MM-PCQA, but MMSIN outperforms the 3D-NSS model. This is not the case when evaluating the WPC2.0 dataset with the same trained model. For this test, the MMSIN model outperforms the MM-PCQA model slightly.

5.6 Ablation Studies

An important task when creating an AI system, are ablation studies. These studies try to find the significance of the different components. For the proposed model, the influence of the number of projections is studied.

Several papers [4] [26] already discuss the influence of the number of projections. In general, a higher number of projections results in an increased score. However, the required resources also scale with the number of projections given at the input. For real time assessment, it should be optimal if a good accuracy can be obtained using as few projections as possible.

All training cycles consist of a batch size of eight, a training duration of 50 epochs for the SJTU-PCQA dataset and 100 for the WPC dataset, compared to the 100 and 600 used to obtain the result in table 5.1. This is done to reduce the time needed to train the models, but it should show comparable discrepancies. The learning rate is 0.001 and the decay rate is 0.0001 . The loss function is the l2rank loss. The number of projections varied from just one up to eight as this was the highest number that could be used while not decreasing the batch size. The results of the test are shown in 5.3

Table 5.3: Influence of the number of projection on the accuracy of the model for the SJTU-PCQA dataset and the WPC dataset. The best result is highlighted in red. The second best in

proj	SJTU-PCQA				WPC			
	SRCC	PLCC	KRCC	RMSE	SRCC	PLCC	KRCC	RMSE
1	0.9026	0.9263	0.7394	0.9103	0.7943	0.8028	0.6058	13.62
2	0.9034	0.9276	0.7447	0.9003	0.8168	0.8212	0.6281	13.05
4	0.9046	0.9233	0.7410	0.9196	0.8332	0.8379	0.6447	12.48
6	0.9145	0.9285	0.7525	0.8922	0.8207	0.8227	0.6299	13.01
8	0.9040	0.9237	0.7320	0.9189	0.8437	0.8458	0.6540	12.20

The results for the SJTU-PCQA stay quite constant. The results for only one projection are only slightly worse compared to those with a total of eight projections. What is more, when going to the highest number of projections, the results become worse. Across the different tests for the SJTU-PCQA dataset, there is only a difference of around 0.01.

The results for the WPC dataset are different. For this dataset, an increased amount of projections also results in an increase of accuracy for the predictions. The results range from an SRCC of 0.7943 for one projection, to a SRCC of 0.8437 for eight projections. A similar difference of around 0.05 is also observed for the other parameters.

5.7 Influence of point size

One parameter that has often been overlooked is the point size when rendering and projecting the point clouds. This is an important parameter. A point size that is too small can result in the background becoming visible. Points that should be blocked by other points could also show through. A point size that is too big on the other hand can make it more difficult to spot smaller distortions. Figure 5.2 shows a comparison between two of the same objects, rendered with the same distortions, but with a different point size.

To see if there is any relation between the point size and the accuracy, a test is performed on the WPC dataset where the point size varies from one to eight. The model trains for 100 epochs, and used four fold cross validation with a batch size of eight. Six projections of the object are used. The other parameters stay constant. The results can be seen in Table 5.4

The best results occur when using a point size of two, followed by a point size of four. A point size of one is too small, and from a point size bigger than four, the accuracy starts degrading as well.

Figure 5.2: Influence of the point size for the same object, with the same distortions. The left image is rendered with a point size of eight (left) and the right one with a point size of 1

Table 5.4: Results of different point sizes. The best results are highlighted in Red, the second best in Blue.

point size	WPC			
	SRCC	PLCC	KRCC	RMSE
1	0.7669	0.7748	0.5786	14.34
2	0.8275	0.8299	0.6384	12.75
3	0.7843	0.7888	0.5983	13.90
4	0.8185	0.8226	0.6234	13.02
6	0.8040	0.8061	0.6080	13.54
8	0.7755	0.7742	0.5828	14.49

5.8 Timing analysis & Utilized Resource

One of the goals of point cloud quality assessment is real time evaluation. This is currently being held back by the data requirements for volumetric video streaming, but also the speed of the quality assessment in itself. Real time assessment seems to be a quite insolvable problem as of today. However, even the smallest time gains can be substantial.

Besides this, the utilised resources are also important. Standalone Head Mounted displays often require a battery and need to stay as light as possible. This means that they have to be optimised and power efficient. So it wouldn't help if the quality assessment model, which might be running in the background, asks for a lot of resources. On the other hand, if an end-user would like to train a model based on their own dataset, they can't be expected to have the latest hardware designed for machine learning. As a result, the model needs to be lightweight.

5.8.1 Single Point cloud assessment

It is important to know how long it takes for the quality of one point cloud to be calculated. In a real life scenario, the different features aren't extracted yet. The speed for this also plays a role. For the MM-PCQA, this test is also conducted, and in the paper a total time of around ten seconds is estimated. To see how long it takes to calculate the perceived quality of one point cloud, the process is split up in the three main blocks, the NSS feature extraction, the image projection and the inference. For one hundred point clouds, the time it takes for each of these steps is calculated and the worst case, best case and average are saved. The results are shown in Figure 5.3.

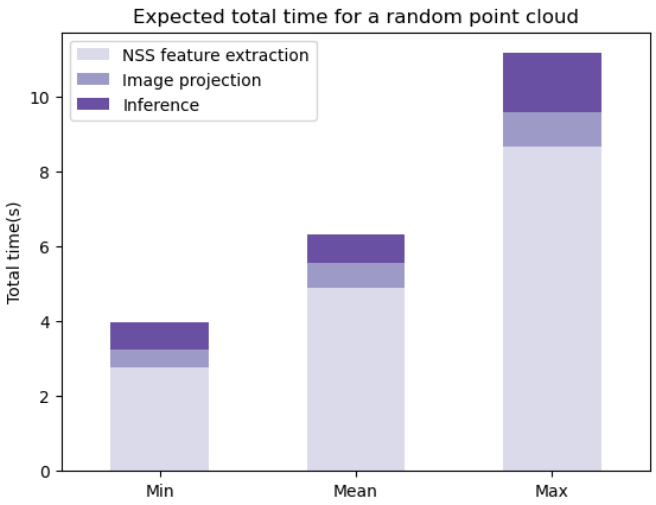

Figure 5.3: Minimum, average and maximum time for the different steps in the point cloud assessment

On average, the entire process takes 6.31 seconds, and in the worst case 11.17 seconds. The figure shows that the NSS feature extraction has the highest influence on the speed. The NSS feature extraction takes on average 4.91 seconds, which is almost 75% of the total time.

5.8.2 Model training

The speed and required memory for the model training can vary a lot based on the number of input features. For the MMSIN model, the results are shown when the number of projections are altered. For the MM-PCQA model, there is a difference in both the number of projections and the number of patches. One important difference is that the MMSIN model was able to train using a batch size of 8 for all tests, while a batch size of 2 had to be used for MM-PCQA model. The results are shown in 5.5

There is a clear difference between the two models. The MMSIN trains around five times faster, while using the same amount of resources.

Table 5.5: Influence of the number of projections (proj) and number of patches (patches) on the training time per epoch (Time), and the required VRAM for both the MMSIN model and the MM-PCQA model.

proj	MMSIN - SJTU		proj +	MM-PCQA - SJTU	
	Time (s)	VRAM (GB)	patches	Time (s)	VRAM(GB)
1	2.8	2.13	1+1	31.2	2.21
2	4.7	2.82	2+2	38.2	3.20
4	7.8	4.01	2+4	53.4	5.11
6	12.1	5.84	4+4	56.8	5.08
8	15.8	7.45	4+6	71.2	7.31

5.9 Quality Assessment Tool

It is hard to easily assess the quality of a point cloud as an end user. Because of this, a simple graphical user interface (GUI) was created. This functions as a proof of concept for a standardised quality assessment tool. An end user uploads a `.ply` file, chooses one of the provided models, and then all the necessary steps happen. The tool starts by generating all the projections and the NSS parameters. The data preprocessing is applied and these are fed to a pre-trained model. Then, the inference step starts, where the model tries to assess the quality. This quality will then be displayed in the GUI, together with a projection of the model.

This GUI is built with the CustomTkinter library. It offers all the functionality that is required, but it lacks a bit in customisation. Other more modern graphical user interfaces like Qt were also considered. These libraries make use of OpenGL, just like the open3d library that generates the projections. Even though these other libraries look nicer, the overhead that comes with them wasn't ideal. A screenshot of the tool is visible in Figure 5.4

The tool is designed to allow the selection of several models. There is the option to choose between different models introduced in this paper. We provide both a lightweight model, trained on only two projections which is designed to assess the quality as fast as possible. This could be useful in the case where a lot of data needs to be assessed. The other model makes six projections and has been trained for a longer period of time, resulting in a higher compute time, but also a higher accuracy.

6 DISCUSSION

In this chapter, the results of the MMSIN model will be discussed. This discussion ranges from how the model compares to other models, as well as what model would fit the best in different scenarios. Finally, possible improvements or new things to try are discussed.

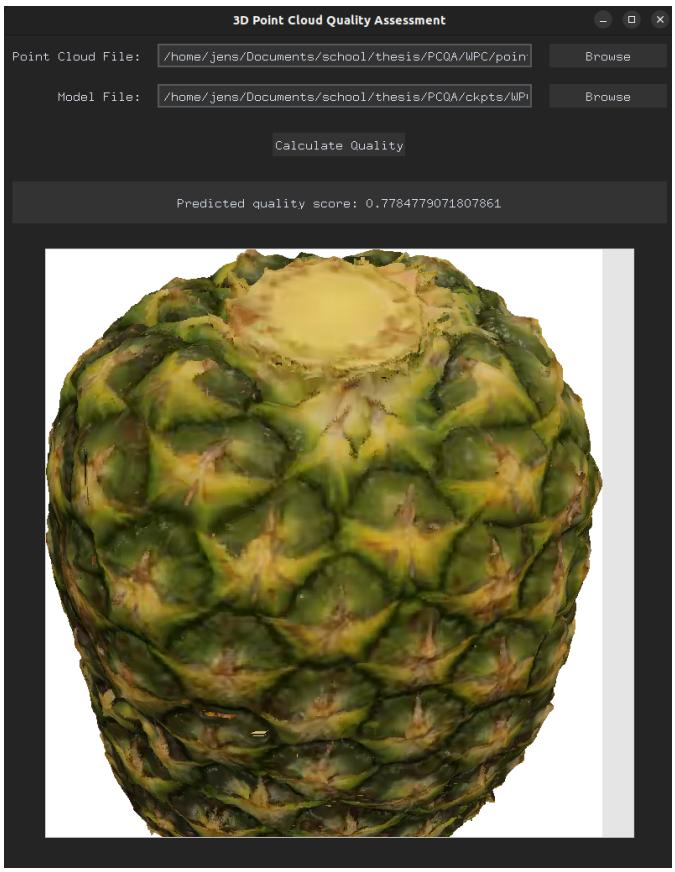

Figure 5.4: Screenshot of the point cloud quality assessment tool. It shows two file selection boxes, and a calculate quality button, after the quality is calculated, a projection of the pineapple is shown in a renderer together with the quality score.

6.1 Accuracy

In general the model outperforms the state of the art. It shows great accuracy after training. For the WPC and WPC2.0 dataset the increase in performance is substantial. It shows that the model is capable of handling multiple datasets. It can do it for datasets like the SJTU-PCQA dataset, which has a lot of different distortions and data, but on the other hand, it is also able to handle datasets like the WPC2.0 dataset, where all distortions are of the same type, but the gravity of the distortions only differ.

The effects of the number of projections is also interesting. From the test results, it looks like there is no real benefit when increasing the number of projections. Not only does the score barely change, the training time and needed resources also increase drastically. However, In the different datasets, the distortions were uniformly distributed over the entire dataset. If however, in a real life scenario , only a part of the model is distorted, there is a risk that these distortions aren't visible because they are blocked by other points. Because of this, at least six projections would be recommended as this covers all possible views. But if a small and fast model needs to be trained

when it is known that possible distortions are across the entire model, it could suffice to only use two projections.

When this is compared to the MM-PCQA model, it is clearly visible that the use of a hybrid model drastically reduced the complexity of the model. By doing this, the VRAM required by the model is reduced significantly, which allows training with an increased batch size. This increase, together with the reduced complexity, means that also the training time is lowered by a lot. Compared to the MM-PCQA model, lowering the number of projections almost has no impact on the performance.

The point size doesn't affect the training time, or required resources. However, it does affect the performance of the model. As expected, a point size that is too small has worse results. The points that lie behind it are showing through, influencing the results. The same is true for a point size that is too big. The distortions aren't distinguishable anymore, as well as any details in the image. As a result, a point size of two or four is recommended.

In terms of speed, the model also does better than the one that is described on the Github page of [4]. On average the assessment is a couple of seconds faster, while also having a better accuracy. Figure 5.3 shows that the main culprit is the NSS feature extraction. If this extraction could be sped up, then the total time needed to assess the point clouds could be reduced to only three to four seconds.

6.2 Cross Dataset evaluation

The results of the Cross dataset evaluation are also interesting. It shows some of the strengths, but also some of the weaknesses with Deep learning. On one hand there are great results when the WPC2.0 dataset is assessed with a model that has been trained on the WPC dataset. But on the other hand, there are the underwhelming results for the the SJTU-PCQA dataset using the same model. This can be explained by a possible domain shift. As was discussed in 5.1, there are multiple ways to construct a point cloud. This can result in a different distribution of data. For the models that use transfer learning like MM-PCQA, this effect can largely be disregarded since they are trained on huge datasets. However, for the NSS features, this is not the case. This can explain why the cross dataset evaluation for the WPC2.0, which has the same reference point clouds, is way better than the evaluation for the SJTU dataset that has point clouds that are created in different conditions. This is a known issue with deep learning in general.

6.3 Future work

This thesis introduces a novel method to assess the point cloud quality that is more efficient compared to other attempts. However, there still remains a lot of room for improvement, mainly towards real time assessment. There are also some parameters of the model that could be researched as well. These options are described below.

6.3.1 Patch-up strategies

One option that was shortly explored is using patch-up strategies as used in [4]. The idea is that the point cloud gets split up in n patches, and that the features are calculated for each of these patches. This is done so that all, or just a subset of the points are in a patch. In this model, this could be done for the calculation of the NSS features. The test results of a model with six patches, containing all points, six projections and a batch size of twelve is shown in Table 6.1

Table 6.1: First test results using the patch-up strategies, the best result is highlighted in red.

patch-up	WPC			
	SRCC	PLCC	KRCC	RMSE
no	0.7818	0.7889	0.5908	14.01
yes	0.7299	0.7327	0.5431	15.50

The results show that the patch-up strategy decreased the accuracy of the model a lot for the same number of training cycles. Besides, generating the NSS features for all patches also requires extra time. Because of this no further tests were conducted, and the patch-up strategy was not explored any further. However, this strategy has proven beneficial in several papers [4] [53]. The increased number of input features probably meant that the training time also had to be increased. A different approach could be to only select a small amount of points, and calculate the features on a subset of the points. This could also improve the speed to calculate the NSS.

6.3.2 NSS features

The influence of the NSS features on the performance can also be interesting to look at in future work. This step is currently a bottleneck as the calculation of the different statistical parameters scales with the number of points. Also, the calculation isn't optimised at all. Multithreading could be an option to significantly speed up the process. On the other hand, it is known that python isn't the fastest language, but it comes with the benefit of having already existing libraries to take care of the calculations. What is

more, it could also be researched whether all features and advanced statistical values are needed.

6.3.3 Hyperparameter tuning

For the current model, the architecture of the model isn't changed that much. All feautures are mapped to a dimension of 1024 before being passing them to the cross-modal attention layers. Also other hyperparameters for both feature extraction have been kept constant, unless mentioned in the paper. For now, also the type of loss function has been kept fairly simple, but these could also still be changed.

6.3.4 Point Cloud Quality Assessment Tool

Something else that also needs improvement is the PCQA tool. In its current state, as a proof of concepts, it gives a good indication of what the tool could be. However, it lacks the ability to interact with the point cloud. This can be a great extra addition for an end user, as it gives more freedom, and also the ability to spot potential distortions that cause the quality problems. A different issue with this tool, is that no quality of a sequence of data can be calculated. However, to be able to do this, a different application altogether would be recommended.

6.3.5 Point cloud assessment dataset

Finally, in the future, the performance could be assessed on different, larger, and more distorted datasets. Also a combination of some of the different datasets could be interesting. Right now, the accuracy stays tightly connected to the dataset. A combination of different datasets, each having their own distortions can result in greater robustness.

7 CONCLUSION

In this work, a novel hybrid multi-modal No-Reference model is proposed to assess the quality of a point cloud. It outperforms currently existing No Reference models. Besides, our approach leverages the strengths of both deep learning and machine learning. This achieves faster training times while requiring fewer resources, making it more usable for a wider public. Its performance is also increased by making use of transformers, enhancing the model's capability to deliver precise and efficient quality assessments.

Looking forward, the model presents numerous avenues for further research. Future studies could for example explore the scalability of the model across other datasets. Also, since the NSS features have the biggest impact on the inference time, we think this is the most important part for further analysis. For example research could be done to make the parameter estimations faster. Another option could be to derive these same parameters with less points since the computational operations scale with the number of points.

We hope that our work is able to contribute to the field of point cloud quality assessment in general. So that one day, it will become possible to assess the quality, both accurately and in real time.

ACKNOWLEDGEMENTS

This thesis would not have been possible without some important people. First and foremost we would like to thank our promotor prof. dr. ir. Maria Torres Vega and co-promotor drs. ir. Jit Chatterjee. We would like to thank them for introducing us to this topic, as well as for guiding us through this challenging, yet fulfilling journey with their support and technical knowledge. Special thanks go to our parents who have always fully supported us in our studies and made us to the people we are today. Finally, we would like to thank our friends, which were there for us in the most difficult moments.

LIST OF SYMBOLS

Acronyms

AGGD	Asymmetric Generalised Gaussian distribution
AR	Augmented Reality
CNN	Convolutional Neural Network
FR	Full Reference
IQA	Image Quality Assessment
GGD	Generalised Gaussian distribution
GUI	Graphical User Interface
KRCC	Kendall's Rank Correlation Coefficient
MOS	Mean Opinion Score
MR	Mixed Reality
NR	No Reference
NSS	Natural Scene Statistics
PCQA	Point Cloud Quality Assessment
PLCC	Pearson's Linear Correlation Coefficient
RMSE	Root Mean Square Error
RR	Reduced Reference
SRCC	Spearman's Rank Correlation Coefficient
VMAF	Video Multi-Method Assessment Fusion.
VR	Virtual Reality

BIBLIOGRAPHY

[1] Market.US, "Global extended reality market by component (hardware, software, and service), by technology (ar, vr, and mr), by end-user (online and offline), by end-user (gaming, retail, healthcare, manufacturing, media & entertainment, education, aerospace & defense, and other end-users), by region and companies - industry segment outlook, market assessment, competition scenario, trends and forecast 2023-2032," February 2024.

[2] Z. Li, C. Bampis, J. Novak, A. Aaron, K. Swanson, A. Moorthy, and J. D. Cock, "Vmaf the journey continues," *Netflix Technology Blog*, October 2018.

[3] S. Van Damme, M. T. Vega, and F. De Turck, "A full- and no-reference metrics accuracy analysis for volumetric media streaming," in *2021 13th International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 225–230, 2021.

[4] Z. Zhang, W. Sun, X. Min, Q. Zhou, J. He, Q. Wang, and G. Zhai, "Mm-pcqa: Multi-modal learning for no-reference point cloud quality assessment," *IJCAI*, 2023.

[5] M. Levoy and T. Whitted, "The use of points as a display primitive," 2000.

[6] G. Turk, "Re-arranging polygons with screw motions," in *Proceedings of the 20th annual conference on Computer graphics and interactive techniques (SIGGRAPH '92)*, pp. 55–64, ACM Press/Addison-Wesley Publishing Co., 1992.

[7] I. Armeni, A. Sax, A. R. Zamir, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1534–1543, 2016.

[8] E. d'Eon, B. Harrison, T. Myers, and P. A. Chou, "8i voxelized full bodies - a voxelized point cloud dataset," Tech. Rep. WG11M40059/WG1M74006, ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG), Geneva, Jan. 2017.

[9] Q. Liu, H. Su, Z. Duanmu, W. Liu, and Z. Wang, "Perceptual quality assessment of colored 3d point clouds," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2022.

[10] H. Su, Z. Duanmu, W. Liu, Q. Liu, and Z. Wang, "Perceptual quality assessment of 3d point clouds," in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 3182–3186, IEEE, 2019.

[11] E. Alexiou, E. Upenik, and T. Ebrahimi, "Towards subjective quality assessment of point cloud imaging in augmented reality," in *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, Oct 2017.

[12] S. Vats, M. Nguyen, S. Van Damme, J. van der Hooft, M. T. Vega, T. Wauters, C. Timmerer, and H. Hellwagner, "A platform for subjective quality assessment in mixed reality environments," in *2023 15th International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 131–134, 2023.

[13] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro, "Geometric distortion metrics for point cloud compression," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3460–3464, 2017.

[14] R. Mekuria, K. Blom, and P. Cesar, "Design, implementation, and evaluation of a point cloud codec for tele-immersive video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 828–842, 2017.

[15] E. Alexiou and T. Ebrahimi, "Towards a point cloud structural similarity metric," in *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1–6, 2020.

[16] E. M. Torlig, E. Alexiou, T. A. Fonseca, R. L. de Queiroz, and T. Ebrahimi, "A novel methodology for quality assessment of voxelized point clouds," in *Optical Engineering + Applications*, 2018.

[17] G. Meynet, Y. NehmÃ©, J. Digne, and G. LavouÃ©, "Pcqm: A full-reference quality metric for colored 3d point clouds," in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–6, 2020.

[18] Q. Yang, Z. Ma, Y. Xu, Z. Li, and J. Sun, "Inferring point cloud quality via graph similarity," 2020.

[19] Q. Liu, H. Yuan, R. Hamzaoui, H. Su, J. Hou, and H. Yang, "Reduced reference perceptual quality model with application to rate control for video-based point cloud compression," *IEEE Transactions on Image Processing*, 2021.

[20] W. Zhou, G. Yue, R. Zhang, Y. Qin, and H. Liu, "Reduced-reference quality assessment of point clouds via content-oriented saliency projection," *IEEE Signal Processing Letters*, vol. 30, pp. 354–358, 2023.

[21] Z. Zhang, W. Sun, X. Min, T. Wang, W. Lu, and G. Zhai, "No-reference quality assessment for 3d colored point cloud and mesh models," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2022.

[22] Y. Liu, Q. Yang, Y. Xu, and L. Yang, "Point cloud quality assessment: Dataset construction and learning-based no-reference metric," 2022.

[23] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, 2013.

[24] Q. Yang, Y. Liu, S. Chen, Y. Xu, and J. Sun, "No-reference point cloud quality assessment via domain adaptation," 2022.

[25] Q. Liu, H. Yuan, H. Su, H. Liu, Y. Wang, H. Yang, and J. Hou, "Pqa-net: Deep no reference point cloud quality assessment via multi-view projection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4645–4660, 2021.

[26] Q. Yang, H. Chen, Z. Ma, Y. Xu, R. Tang, and J. Sun, "Predicting the perceptual quality of point cloud: A 3d-to-2d projection-based exploration," *IEEE Transactions on Multimedia*, vol. 23, pp. 3877–3891, 2021.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[28] H. Jun, B. Ko, Y. Kim, I. Kim, and J. Kim, "Combination of multiple global descriptors for image retrieval," 2020.

[29] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," 2017.

[30] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012.

[31] A. K. Moorthy and A. C. Bovik, "Blind image quality assessment: From natural scene statistics to perceptual quality," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3350–3364, 2011.

[32] D. de la Iglesia Castro, K. Mader, Hanchen, B. Sullivan, pandu rao, Sebastian, J. Cole, M. Wallbaum, N. Mitchell, M. Movva, P. McCartney, S. Jadhav, Y. B. de Miguel, A. Milan, B. Mitzkus, D. Bazazian, G. Gandenberger, J. Buchanan, M. D. Trevisani, M. Quach, N-McA, R. Joshi, R. Tweedie, brett koonce, fatih, iindovina, J. Hadfield, joskaaaa, and nokonoko1203, "daavoo/pyntcloud: v0.3.1," July 2022.

[33] F. E. Correa-Tome, R. E. Sanchez-Yanez, and V. Ayala-Ramirez, "Comparison of perceptual color spaces for natural image segmentation tasks," *Optical Engineering*, vol. 50, no. 11, p. 117203, 2011.

[34] M. Tkalcic and J. Tasic, "Colour spaces: perceptual, historical and applicational background," in *The IEEE Region 8 EUROCON 2003. Computer as a Tool.*, vol. 1, pp. 304–308 vol.1, 2003.

[35] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014.

[36] K. Sharifi and A. Leon-Garcia, "Estimation of shape parameter for generalized gaussian distributions in subband decompositions of video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 1, pp. 52–56, 1995.

[37] N.-E. Lasmar, Y. Stitou, and Y. Berthoumieu, "Multiscale skewed heavy tailed model for texture analysis," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, pp. 2281–2284, 2009.

[38] J. A. Dominguez Molina, G. GonzÃ¡lez FarÃas, and R. RodrÃguez-Dagnino, "A practical procedure to estimate the shape parameter in the generalized gaussian distribution," 01 2003.

[39] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[42] W. Sun, X. Min, W. Lu, and G. Zhai, "A deep learning based no-reference quality assessment model for ugc videos," in *Proceedings of the 30th ACM International Conference on Multimedia*, ACM, Oct. 2022.

[43] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen, "Point-e: A system for generating 3d point clouds from complex prompts," 2022.

[44] CloudCompare, "Cloudcompare-3d point cloud and mesh pro-cessing software." `https://www.danielgm.net/cc/`.

[45] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "MeshLab: an Open-Source Mesh Processing Tool," in *Eurographics Italian Chapter Conference* (V. Scarano, R. D. Chiara, and U. Erra, eds.), The Eurographics Association, 2008.

[46] K. Mammou and P. Chou, "Pcc test model category 13 v2," in *ISO/IEC JTC1/SC29/WG11 MPEG, N17519*, 2018.

[47] "Pcc test model category 2 v0," 2017.

[48] K. Mammou, "Pcc test model category 3 v1," in *ISO/IEC JTC1/SC29/WG11 MPEG N17349*, 2018.

[49] MPEG, "Mpeg-i: Visual volumetric video-based coding (v3c) and video-based point cloud compression (v-pcc)." `https://www.mpeg.org/standards/MPEG-I/5/`.

[50] K. Pearson, "On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 50, pp. 157–175, 1900.

[51] C. Spearman, "The proof and measurement of association between two things," *The American Journal of Psychology*, vol. 15, no. 1, pp. 72–101, 1904.

[52] M. G. Kendall, "The treatment of ties in ranking problems," *Biometrika*, vol. 33, no. 3, pp. 239–251, 1945.

[53] M. Tliba, A. Chetouani, G. Valenzise, and F. Dufaux, "Representation learning optimization for 3d point cloud quality assessment without reference," in *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 3702–3706, 2022.