University of Brasília

Exact Sciences Institute
Computer Science Department

3D Point-Cloud Quality Assessment Using Color and Geometry Texture Descriptors

Rafael Diniz

Thesis submitted in partial fullfilment of
the requirements to Doctoral Degree in Informatics

Advisor
Prof.a Dr.a Mylène Christine Queiroz de Farias

Co-advisor
Prof. Dr. Pedro Garcia Freitas

Brasília
2021

University of Brasília

Exact Sciences Institute
Computer Science Department

3D Point-Cloud Quality Assessment Using Color and Geometry Texture Descriptors

Rafael Diniz

Thesis submitted in partial fullfilment of
the requirements to Doctoral Degree in Informatics

Prof.a Dr.a Mylène Christine Queiroz de Farias (Advisor)
University of Brasília

Prof. Dr. Camilo Chang Dorea
University of Brasília

Prof. Dr. Guido Lemos
Federal University of Paraíba

Prof. Dr. Roberto Gerson De Albuquerque Azevedo
ETH Zürich

Prof.a Dr.a Genaína Nunes Rodirgues
Coordinator of Graduate Program in Informatics

Brasília, July 26, 2021

Dedicated to

This thesis is dedicated to my advisor Mylène Christine Queiroz Farias, to my parents Constantino Marques Diniz and Fabíola Graça do Amaral Diniz, to my wife Anna Orlova Diniz, and to my daughter Joanna Orlova Diniz.

I also dedicate this thesis to Luiz Fernando Gomes Soares (in memoriam), my advisor during my master's course, which without him I'd not arrive to this point of academic life.

Acknowledgements

I'm glad Pedro Garcia Freitas was my co-advisor. Thanks for the co-authoring of many publications, and for the research done together.

Thanks to Prof. Marcelo Menezes de Carvalho, for all the help during these years in Brasília, and my colleagues at Digital Signal Processing Group, with which I shared many good moments and relevant scientific discussion, namely, Mouhamad, Sana, Dário, Gustavo Sandri, Henrique, André, Priscila and all the others. Also thanks to Prof. Queiroz and the LISA laboratory of UnB for their work on point clouds and our helpful discussions.

Also I'm very glad for my friends in Brasília, with whom I was able to enjoy very good moments of my stay in the capital of the country, especially Rogério Basali and Mari, Lua, Jupagul, Adriana, and my friends from Vila Planalto, Gil, Chaguinha, Careca, Loti, Paulo, Chico, Diogenes and so on.

I'd like to thank the University of Brasília (UnB), specially the Computer Science Department, for all the support and motivation given to me during all my course.

I would like to acknowledge the financial support for this work provided by scholarships from the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Fundação de Apoio à Pesquisa do Distrito Federal (FAP-DF).

Abstract

Since the mid 20th century the use of digital formats for visual content allowed a great evolution on how the society communicates. The Internet and digital broadcast systems introduced in the decade of 90 to the wider public allowed an incredible expansion of multimedia consumption by the people, while the telecommunication networks and providers were pushed to their limits to address the growing multimedia content demand.

Older electronic imaging systems, notably TV broadcasting systems, were designed after long subjective quality analysis for the definition of parameters like number of lines of the video. But recent digital visual content services need faster and affordable ways of evaluating the human perceived quality of the always evolving multimedia systems.

To address the need of automatic quality assessment, in the past decades many visual quality models based on algorithms which run on digital computers have been proposed. Different types of metrics to access the quality of still images and video were developed and provide good correlation to the perception of quality by humans. While the current metrics are very advanced for 2D digital imagery, a new set of immersive media is dawning, with different data structures, to which the 2D methods are not applicable, and need new quality assessment metrics.

The new visual immersive media formats provide a 3D visual representation of real objects and scenes. In this new visual format, objects can be captured, compressed, transmitted and visualized in real-time not anymore as a flat 2D image, but as 3D content, allowing free view point selection by a consumer of such media. One of the most popular formats for immersive media is Point Cloud (PC), which is composed by points with 3 geometry coordinates plus color information, and sometimes, other information like reflectance and transparency.

In this work it is presented a research about quality assessment of 3D Point Clouds based on novel color and geometric texture statistics. Considering that distortions to both color and geometry attributes of 3D visual content affect the perceived visual quality, it is proposed in this work to use both color-based and geometry-based texture descriptors for PC to obtain the visual degradation through their statistics.

The proposed model for quality evaluation is a full-reference method, which means

it uses information from the reference PC and degraded version of the PC to obtain a quality estimation. This work introduces 4 novel PC texture descriptors, 3 of them color-based, while 1 is geometry based. Also, a new voxelization method is proposed, which converts points to voxels (volume elements), and improves the performance of the color-based texture descriptors. The performance of the proposed PC quality assessment method is among the best of the state-of-the-art PC quality assessment methods, while being flexible and extensible to adapt to different types of distortions.

Keywords: Point Cloud Quality Assessment, RBG-D, volumetric video, point cloud, mesh, virtual reality, mixed reality

Resumo

Desde meados do século 20 o uso de formatos digitais para conteúdo visual permitiu uma grande evolução sobre como a sociedade se comunica. A Internet e os sistemas de transmissão digital introduzidos na década de 90 para o público em geral permitiu uma expansão incrível do consumo de conteúdo multimídia pela população, ao mesmo tempo que as redes de telecomunicações e os provedores foram levados ao limite para lidar com a crescente demanda de conteúdo multimídia.

Sistemas de imagem eletrônicos mais antigos, principalmente sistemas de transmissão de TV, foram projetados após uma longa análise subjetiva de qualidade para a definição de parâmetros como número de linhas do vídeo. No entanto serviços com conteúdo visual digital mais recentes precisam de maneiras rápidas e acessíveis de avaliar a qualidade percebida por seres humanos dos sistemas multimídia em constante evolução.

Para atender à necessidade de avaliação automática da qualidade, nas últimas décadas, muitos modelos de qualidade visual baseados em algoritmos que funcionam em computadores digitais foram propostos. Tipos diferentes de métricas para acessar a qualidade de imagens estáticas e vídeo foram desenvolvidas e fornecem boa correlação com a percepção de qualidade por humanos. Enquanto as métricas atuais são muito avançadas para imagens digitais 2D, um novo conjunto de mídias imersivas está surgindo, com diferentes estruturas de dados, para as quais os métodos 2D não são aplicáveis e precisam de novas métricas de avaliação de qualidade.

Os novos formatos de mídia visual imersiva fornecem uma representação visual 3D de objetos e cenas reais. Neste novo formato visual, objetos podem ser capturados, comprimidos, transmitidos e visualizados em tempo real, não mais como uma imagem 2D, mas como conteúdo visual 3D, permitindo a livre seleção do ponto de vista por um consumidor de tal mídia. Um dos formatos mais populares para mídia imersiva é o Point Cloud (PC), que é composto por pontos com 3 coordenadas geométricas e informações de cores e, às vezes, outras informações como refletância e transparência.

Neste trabalho é apresentada uma pesquisa sobre avaliação da qualidade de Point Clouds 3D com base em estatísticas de texturas de cor e geometria inovadoras. Considerando que distorções em ambos os atributos de cor e geometria do conteúdo visual 3D

afetam a qualidade visual percebida, é proposto neste trabalho usar ambos descritores de textura baseados em cor e geometria para PC para se obter a degradação visual através de suas estatísticas.

O modelo proposto para avaliação da qualidade é um método de referência completa, o que significa que usa informações do PC de referência e da versão degradada do PC para obter uma estimativa de qualidade. Este trabalho apresenta 4 novos descritores de texturas para PC, 3 deles baseados em cores, enquanto 1 é baseado em geometria. Um novo método de voxelização é também proposto, que converte pontos em voxels (elementos de volume) e melhora o desempenho dos descritores de textura baseados em cores. O desempenho da proposta de avaliação da qualidade de PC está entre os melhores métodos do estado da arte para avaliação da qualidade de PC, sendo flexível e extensível para se adaptar a diferentes tipos de distorções.

Palavras-chave: Point Cloud Quality Assessment, RBG-D, volumetric video, point cloud, mesh, virtual reality, mixed reality

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, 3D imaging technologies have advanced at a fast pace, allowing more faithful visual representations of the real world, paving the way to a new visual media, not anymore a window to the world, but where a real object and its volumetric virtual image are almost indistinguishable. What was previously available for a glimpse only in science fiction movies and futuristic predictions, now with the ongoing research on volumetric capturing, coding and presentation, realistic mixed reality experiences are becoming a reality.

Advances on devices which capture and present 3D imagery content boosted the research and development of algorithms and techniques to capture, compress, transmit, present and assess the quality of volumetric content. These devices represent the visual data using an approximation of the plenoptic illumination function, which can describe visible objects in any position and point-of-view of the 3D space. Among the data representations for 3D imaging are holograms, light fields and point clouds (PCs).

Point clouds are the most popular volumetric media, being composed by elements with 3D geometric coordinates, color information, and sometimes other attributes like reflectance coefficient. Nevertheless, PCs require a large number of points to accurately represent a 3D scene, and hence an impractical bitrate. So, new codecs for PCs were developed and in 2021, ISO/MPEG published the first international standard for visual immersive media [5] coding, which comprises all compressing and decompressing steps of a 3D point cloud. Figure 1.1 shows a point cloud rendered in a way the points can be clearly seen.

This thesis is focused on the quality assessment of 3D point cloud content, more specifically the objective quality assessment of volumetric content, which predicts the average human quality perception through an automatic way, with the use of algorithms. The rest of this introduction contains the description of the problem, a summary of the contributions, and finally the overall organization of this thesis.

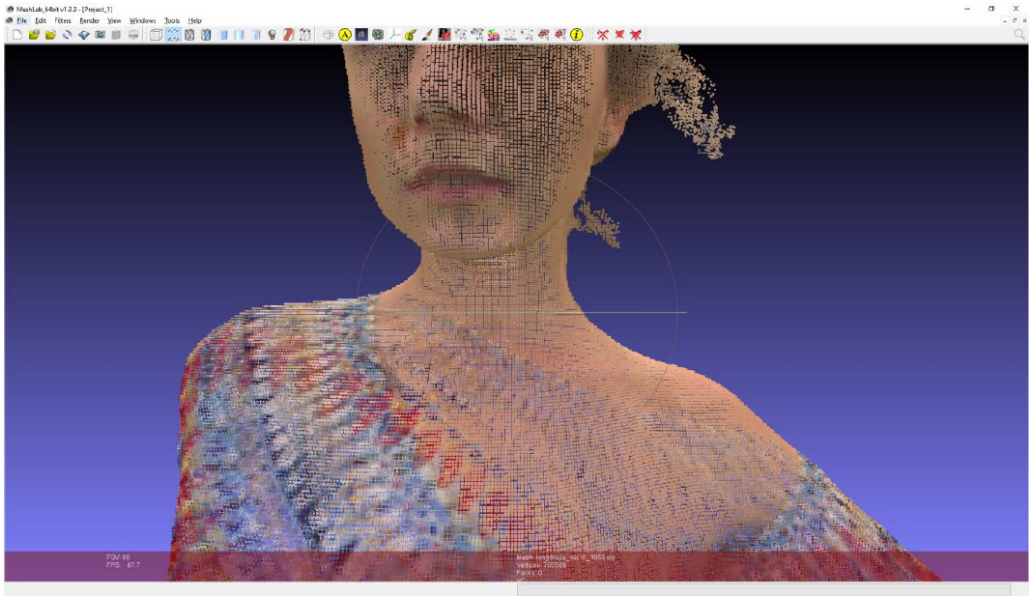


Figure 1.1: Visualization example point-cloud.

1.1 Problem Description

The last decade experienced a many fold increase of the consumption and production of digital multimedia content. In just one year, with the COVID-19 pandemic, the use of tele-presence systems more than doubled in a short period of time. According to Cisco [6], an average of 4.7 times more workers are working from home. It is expected, considering the current context, that 3D imaging systems will receive widespread adoption, and as 3D visual content need even more bandwidth than previous generation visual systems, tools that can optimize the bitrate while keeping the best possible quality of experience (QoE) is extremely relevant.

Users will typically consume 3D PC content with displays very close the eyes, through head-mounted displays, so PC visual impairments can easily degrade user's experience. While most of past visual quality assessment metrics were tailored for quality assessment of 2D and 3D stereoscopic visual medias (i.e. 2D still image or video), with the recent advances of 3D visual systems, also known as visual immersive media or volumetric media, new metrics adapted to this type of content need to be developed.

This thesis addresses the problem of the PC quality assessment, through the development of a method that provides an objective full-reference (FR) PC quality assessment method with a good performance to any kind of content distortion. An objective metric should predict the quality of a given content in an automatic manner, without human intervention, while being a full-reference metric means the proposed metric considers the

information of the original content and the distorted content to predict the quality, as close as possible to the human visual system perception.

1.2 Proposed Method

This work started with an initial research how 3D content is captured and displayed. The common techniques used by the emerging 3D systems were evaluated and real tests were performed, including capture and playback of point clouds.

After the initial research phase, two approaches were used to develop a new method for 3D point clouds quality assessment. First, we used ideas from metrics already existent for 2D images and adapted for 3D point clouds. This is the case of the Local Binary Patterns, which already presented a solid performance for 2D image quality assessement [4] and was adapted in this research to work with 3D point clouds. The second approach was the creation of completely new texture descriptors, which were developed to extract both color-based texture information and geometry-based texture information. Also, an innovative voxelization technique was developed, in order to emulate how the rendering system works, and improve the color-based texture descriptors quality assessment performance. The general idea of the proposed quality assessment method based on texture descriptors is presented in Figure 1.2. The diagram represents, from the left to right, the input reference and test contents, the pre-processing step (ie. voxelization), the texture descriptor application, the texture descriptor histograms distance calculation and the mathematical regression of the distances, which outputs the final quality score of the test content.

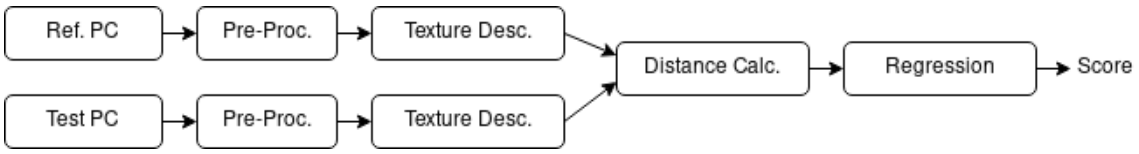


Figure 1.2: Diagram of the proposed PC quality assessment method based on texture descriptors.

After the development of the texture descriptors, a larger quality assessment framework was established. Statistical methods for the calculation of the distances between texture descriptor histograms were compared and discussed. Also, regression methods were compared and used in the proposed PC quality assessment framework. Finally, the proposed quality assessment method is compared to state-of-the-art metrics.

1.3 Summary Of Contributions

In the beginning of this research some methods for point cloud capture using just a single RGB-D camera were proposed [3]. The main goal of this initial development was simple setups for volumetric video capture which rely on a single off-the-shelf RGB-D capture device. The proposed method first captures the model of a human head, and then a live RGB-D feed from the same head gets reconstructed by the registration of the pre-captured model to each live RGB-D frame. Graphical results of this work are shown in Figures 2.13, 2.10, 2.10, 2.11 and 2.12. Chapter 2 exposes the outcome of this initial work, plus a comprehensive overview of visual immersive media.

Nevertheless the main contributions of this thesis are the development of a quality assessment framework for point clouds with any type of distortions. Chapter 3 contains all the research carried out, which was in part already published in important conferences and journals [3, 7, 8, 9, 10, 11], while the not published parts are in final stage of acceptance by journals. The contributions of this thesis are listed below:

- Parameterized voxelization method;
- 4 novel PC texture descriptors based on local PC neighborhoods;
- Statistical analysis of the proposed texture descriptors on different data-sets;
- A model for PC quality assessment based on texture descriptors.

Among the contributions is the proposed voxelization procedure, described in Chapter 3.1.1. The proposed voxelization method establishes the voxel size of point cloud points (initially with no volume information) based on the average distance of the nearest neighbors of each PC point. The proposed voxelization technique was discussed and published also in the articles [9] and [10]. The main contribution of this work are the proposal of 4 novel PC texture descriptors. One of the proposed texture descriptors is based on the Local Binary Patterns (LBP), which was initially conceived for 2D images. The LBP texture descriptor adapted to point clouds is presented in Chapter 3.1.2, and also presented article [7]. The other proposed texture descriptors are totally novel. The Local Luminance Patterns (LLP) introduces an innovative way to define the PC texture, based on the luminance of local neighborhoods, being described in Chapter 3.1.3, while an article discussing it was also published [9]. The Local CIEDE2000 Patterns (LCP) is another texture descriptor proposal, based of the CIELab CIEDE2000 distances between each PC point and a local neighborhood. The LCP is discussed in the Chapter 3.1.4, [11] and [10]. The last proposed descriptor is the geometry-based texture descriptor, presented in Chapter 3.1.5 and [10].

Another contribution of this research is the overall quality assessment framework, which includes, apart of the definition of the texture descriptors, the distance metrics to be used to evaluate the texture descriptor statistics (Chapter 3.2) and the final quality prediction modeling, discussed in Chapter 3.3. At last, our final contribution is the joint use of color-based texture descriptor and geometry-based texture descriptor to obtain a final quality assessment prediction for degraded PCs, presented in Chapter 4, and also discussed in the IEEE SPL published article [10]. The performance of the joint use of color-based and a geometry-based texture descriptor proposed by this research presents better correlation than other state-of-the-art PC quality assessment metrics.

1.4 Organization Of This Thesis

This thesis is organized in 5 chapters: this introduction; an overview of immersive media; the proposed PC quality assessment method; simulation results and comparisons; and conclusions. Chapter 2 contains an overview of the immersive media ecosystem and its quality assessment methods. Chapter 3 describes the 3D quality assessment contributions of this thesis, containing all the proposed texture descriptors and PC quality assessment methods. Chapter 4 contains the experimental setup, simulation results and comparisons of the proposed PC quality assessment methods and other state-of-the-art methods. Chapter 5 contains the conclusions of this work.

Chapter 2

Overview

Recent technology advancements have driven the production of devices that capture and display visual contents in a much more realistic way than 2D images. Among these technologies are the light-fields, holography and point-clouds. These new media differ from 2D image and video, and need new methods not only to capture and display, but also for compressing and to assess the quality of these compressed immersive formats. This chapter contains an overview of visual immersive media, with a more detailed discussion about subjective and objective quality assessment methods of 3D point clouds, the main topic of this thesis.

2.1 Visual Immersive Media

Recently, immersive image and video was the nomenclature adopted for the new generation of imaging formats. Devices and technologies supporting this new generation media represent the visual information using more dimensions of the plenoptic illumination function than previous formats. The plenoptic function describes every possible view, from every position, at every moment, and at every wavelenght [1]. Figure 2.1 exemplifies how light rays reach the observer's eyes. The 7D plenoptic function, $P(x, y, z, \theta, \phi, t, \lambda)$, represents the light observed from every position and direction in 3-dimensional space, where x, y, z represents any viewpoint, θ, ϕ represents any angular viewing direction, over time t and for each wavelenght λ, as illustrated by Figure 2.2.

Because of the high dimensionality of the plenoptic function, practical visual representations use an approximation of it. Examples of this approximation are holograms, light fields, or point clouds (PC) imaging formats. 2D image representations are also approximation of the plenoptic function, but at a reduced dimensionality when compared to the mentioned visual immersive formats.

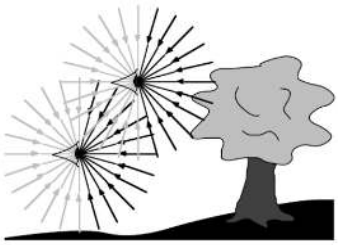


Figure 2.1: Light ray pattern towards the observer's eyes, by Adelson and Bergen [1].

Figure 2.2: Plenoptic function measures the intensity of light seen in all possible positions, viewing angles, over time and for each wavelength [1].

The hologram was introduced by Gabor [12] in 1947 and consists of a method for recording a light field, rather than an image formed by lens. Holograms encode the light field as an interference pattern of variations in the opacity, density, or surface profile of the photographic medium. To display, the hologram's interference pattern diffracts the light into an accurate reproduction of the original light field. In recent decades, with the advent of digital holography [13], the photographic physical medium was substituted by digital sensor arrays and the image rendering is now performed from digitized interferograms. Many methods for recording and processing digital holograms exist, but several challenges exists related to the optical capture and display of holograms, as seem in recent publications [14, 15, 16]. An example of a setup for holography capture and reconstruction is shown in Figure 2.3.

Light field [17] is an imaging technology which describes the distribution of light rays in empty space. Real world light-field implementations typically measure the distribution of light rays as a function of position and angle. In these light-fields, also called 4D light-fields, the light rays are defined in a coordinate system denoted by two planes, (u, v) for the first plane and (s, t) for the second plane, and can be represented as $L(u, v, s, t)$. An oriented light ray defined in the system first intersects the uv plane at coordinate (u, v)

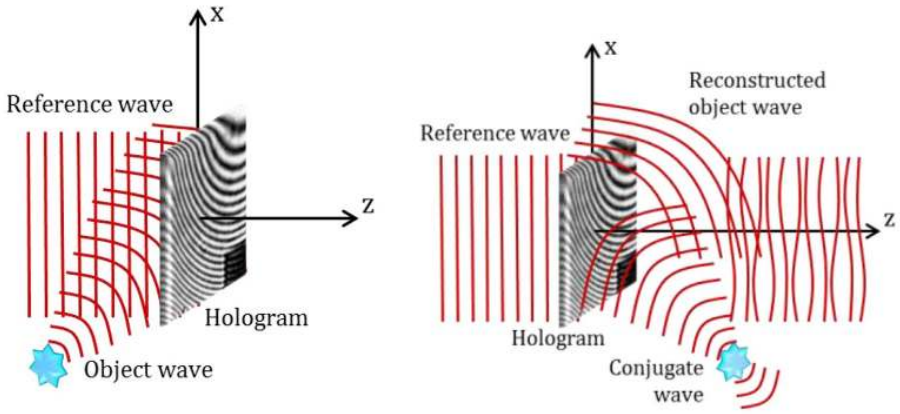


Figure 2.3: Recording and reconstruction of an object light wave through a hologram.

and, then, intersects the st plane at coordinate (s, t). In typical setups, the st plane is the camera array, and the uv plane the camera's focal plane. A 4D light-field can be represented by a 2D array of 2D images, as shown Figure 2.4.

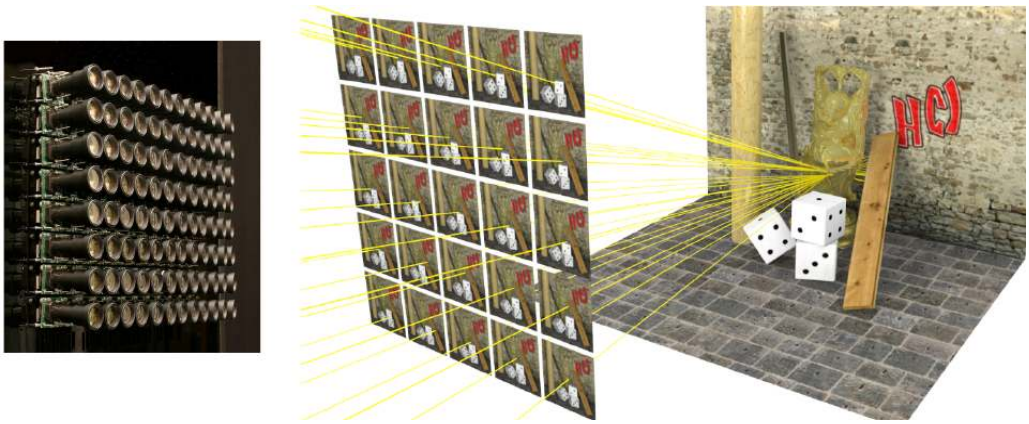


Figure 2.4: Light-field capture setup with an array of cameras which can capture light rays from different angles, provided by Instituto Superior Técnico of Lisbon (IST).

Finally, point-clouds, one of the visual representations which gained more acceptance recently for volumetric visual representation [18], consists of elements with 3D coordinates plus color channels and, in some cases, other attributes are also present like surface normal vector, reflectance and opacity. Point-clouds are typically captured using one or more cameras with a photographic sensor plus a depth sensor, also called RGB-D cameras. These cameras allow the capture of a 2D image frame plus an aligned 2D depth map, which contains the distance between the camera and an object for each pixel or group of them in the 2D image. This RGB-D data allows the creation of point-clouds,

by transforming the RGB-D camera-coordinate elements to point-cloud world-coordinate elements. When more than one camera is available for capturing, the many point-cloud patches produced by each camera are fused to create one point-cloud. Examples of a multi-camera setup for point cloud capture and an illustration of rendering differences between 2D and 3D imaging are shown in Figures 2.6 and 2.5 respectively.

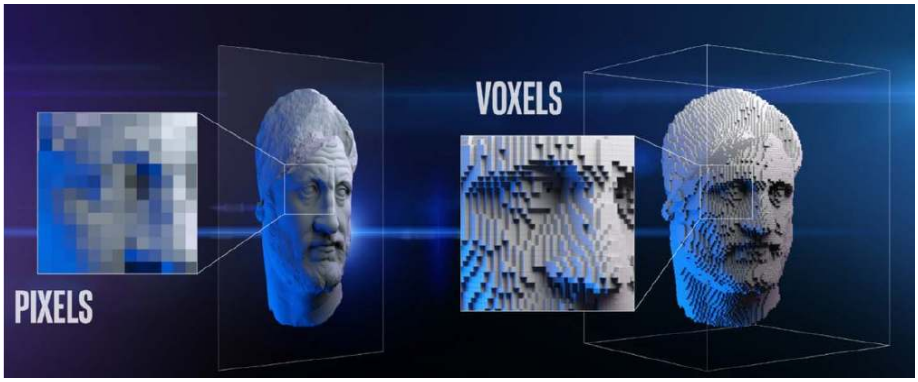


Figure 2.5: 2D image with picture elements (pixels) on the left, and 3D point-cloud with volume elements (voxels) on the right, provided by IST Lisbon.

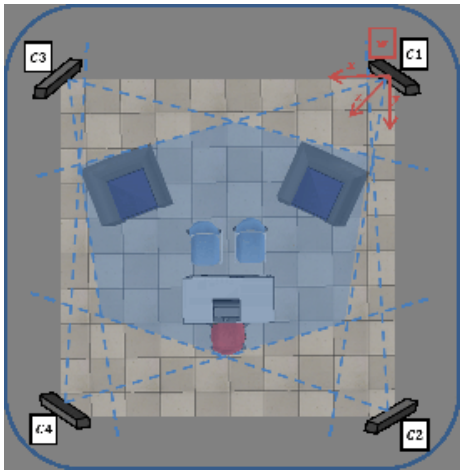


Figure 2.6: Point-cloud capture illustration with 4 cameras capture setup, by IST Lisbon.

Another common 3D representation are the meshes. 3D polygon meshes are being used for decades since the inception of the 3D Computer Graphics to represent 3D objects. Meshes are composed by vertices, edges and faces. Each face is typically triangle, but can have any shape, and the resulting representation is a polyhedral 3D shape, which is commonly used to represent solid objects. However, point-clouds are simpler to obtain than 3D polygon meshes, as surface reconstruction does not need be computed,

considering RGB-D capture setups provide no edge connectivity or surface properties. So point-clouds have a more compact and direct representation of the captured 3D visual content, so they are more computationally efficient - a relevant aspect for real-time immersive media systems.

While meshes can be displayed in a canonical way, the point-cloud elements have no intrinsic volume size, needing more semantics to be rendered for visualization. A process called voxelization converts infinitesimal point-cloud elements to volume elements, by attributing each point a volumetric dimension which optimally produces an output which can be rendered without holes. Indeed, voxelized point-clouds are being used by codecs being developed for volumetric video [19]. A voxelized point-cloud is a point-cloud where its 3D points are converted to voxels, which are typically small 3D cubes. A correctly voxelized point-cloud can be used to represent solid objects [20], but an incorrectly voxelized point-cloud might present empty voxels (holes) between each occupied voxel, allowing an user to see through an object, thus reducing the quality of experience. A mesh representation, obviously, does not suffer from this problem.

One common attribute of all the mentioned visual immersive media are the support for 6 degrees of freedom by the observer. Degrees of Freedom (DoF) in this context refer to the types of movement of a rigid body inside a 3D space, being in total 3 translations and 3 rotations. The 3 translations are typically named forward/backward, up/down and left/right, while the 3 rotations are named yaw, pitch and roll. Figure 2.7 illustrates the Degrees of Freedom from an observer point of view.

Figure 2.7: Previous 3 Degrees of Freedom and immersive 6 Degrees of Freedom illustration plus the names of the 6 types of movements on the right, by IST Lisbon.

Support for 6 DoF is a pre-requisite for visual immersive experiences. The visual immersive media can be assessed by a person in any Milgram's reality-virtuality continuum [2] context, apart of the full reality extreme. The reality-virtuality continuum contains in one end the real environment, and in the other end, the virtual environment.

Common names for different ranges of the reality-virtuality continuum, excluding the full reality, are the augmented reality (AR), virtual reality (VR) or mixed reality (MR) variations, also collectively called extended reality (XR), as shown in Figure 2.8.

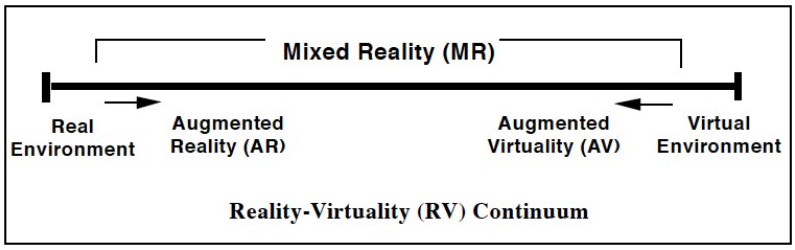


Figure 2.8: Reality-virtuality continuum by Milgram et al [2].

While point-clouds can accurately describe a 3D scene for immersive experiences, they require a large number of points and increased computation complexity, which limits their use in real applications [21]. As a consequence, new technologies and international standards were developed for compression of point-clouds. The MPEG Immersive Media standard (MPEG-I) series will contain at least two standards for immersive visual media coding. One of them, the already published MPEG ISO/IEC 23090-5 [5], relies on a 2D video codec plus volumetric mapping information for encoding 3D point clouds. A volumetric content is first split in patches, projected to a 2D frame, and side information is added with occupancy maps and the geometric information for later volumetric reconstruction by the decoder. Other standard for immersive media which is in late stages prior publication is the MPEG ISO/IEC 23090-9 [22], which is the geometry-based point cloud codec. The geometry-based point cloud codec encodes the point cloud's geometry and color information as voxels, meaning that geometry also becomes a first class data entity [19] to be later reconstructed by the decoder. While one of the standards is already published, the other is in the last stage of development. The 2D video-based codec is more appropriate for low bitrate coding of small scene and objects, like humans with background, while the geometry-based point-cloud coding is more appropriate for large scale point-clouds, like 3D cities scans and cultural heritage capture with very fine geometric precision. Reference implementation of these codecs exist and are widely used in academic research, as seem later in this chapter. Commercial volumetric video technologies are also available (eg. Interdigital [1]), and Brazil is considering the adoption of the ISO/IEC 23090-5 as national standard for the terrestrial DTV volumetric video de-

[1]Interdigital Immersive Lab: `https://www.interdigital.com/immersive-lab`

livery. The Brazilian SBTVD Forum evaluation about the Brazilian 3.0 TV technologies is occurring in 2021 (See SBTVD Forum TV 3.0 [2]).

A popular data structure to represent a point-cloud is the octree [23] [24]. Octree partitioning keep dividing a volume in small sub-volumes, while there is one or more points inside the volume, as presented in Figure 2.9. Other data structures exist, like spanning trees [25], binary trees [26] or based on a graph representation [27]. In MPEG geometry-based encoding, octrees are used together with a triangle soup rendering method [28].

Figure 2.9: Two variations of octree, in the left a traditional octree partitioning where cubes with points are divided until a target layer is reach, and in the right an octree approach where partitioning a cube is based on split decisions, for example, based on RDO (Rate-Distortion Optimization).

2.1.1 Capture

As mentioned earlier, capturing the world in 3D is not an easy task. Most of the currently available work suggest the use of at least 3 RGB-D sensors for a fair volumetric reconstruction [29]. The difficulty in assembling and calibrating such an array of RGB-D sensors limits their popularity in more mainstream communication applications. Previous to the availability of affordable RGB-D sensors, multi-view stereo based volumetric reconstruction was typically used. Stereo based volumetric image reconstruction relies on captured views at difference angles to extract the volumetric shape from an object [30]. This area is a pretty mature field of research, but has its limitations due to inherent absence of real captured depth data.

Volumetric image reconstruction received substantial attention recently, but the reconstruction of 3D objects from RGB-D frames still faces many challenges, like for example, noisy and missing data acquired from the sensors [31] and lighting differences between sensors. The reconstruction methods available to obtain of a volumetric content from multiple RGB-D capture devices is examined in depth by Berger [31], which first classifies reconstruction methods in relation to point-cloud artefacts, like missing data, misalignment and non-uniform sampling, and input requirements, like presence or not of surface normals. The mentioned techniques of surface reconstruction, as pointed by Berger, has

[2]Brazilian TV 3.0 project page: `forumsbtvd.org.br/tv3_0/`.

grown from methods that handle limited defects in point-clouds reconstructions, to methods that handle substantial artefacts, while he also discusses about a growing development of data-driven reconstruction algorithms, which use large point-cloud database and allows for a method to identify classes and properties of objects. As discussed, reconstruction can use the color and geometry information from sensors, but also use prior information. Firman et at. [32], for example, proposes a structure prediction of unobserved voxels from single depth view by employing classes of 3D mesh models used as reference for the reconstruction. Alexiadis et al. [33] also proposes a reconstruction method specific for human body reconstruction, while Boldi et al. [34] proposes a method specific for face reconstruction. On the specific case of 3D volumetric face reconstruction, other approaches uses just 2D color images as input, like in [35] and [36].

Concerning deep learning approaches for 3D shape generation and completion, 3D ShapeNets [37] uses deep learning to train a 3D convolutional network from a shape database and complete or repair shapes, including broken meshes [38]. Other works which use deep learning for object shape reconstruction include Rock et at. [39] and Song et al. [40], all with a similar approach, using deep learning knowledge acquired with volumetric objects datasets.

While multi-camera setups are desirable, it is possible to do a 3D scan using a single consumer grade RGB-D. Among these, are that work of Hernandez et al. that implements a 3D face scan using a single RGB-D device [41]. Also, Kinect Fusion work [42] and others, provide tools for performing a good quality 3D scanning using just one RGB-D sensor. Kinect Fusion's technique consists of rotating the camera or the object in order to capture all its faces. Farias and I proposed and implemented a 3D volumetric video capture system which reconstructs the human head from a single RGB-D capture device [3]. The head model of the user is first captured by registering multiple RGB-D frames captured from many angles, and then, the model is used for reconstructing the whole head for each live captured frame from the RGB-D camera. Examples of a head model, a live captured point-cloud created from a single RGB-D source, and the reconstructed head are shown in Figures 2.10, 2.11 and 2.12. As seen in the examples, some challenges do exist for RGB-D single camera capture, for example, caused by different illumination conditions between model capture and live frame capture and also different focus, exposure time and other intrinsic camera parameters which are automatically adjusted in low-end devices.

Current RGB-D camera devices provide at least two separate streams: color and depth, and sometimes also the captured infra-red (IR) frame is available. These color and depth streams come from different types of sensors inside the device. Naturally, each sensor has different accuracy and noise levels and, typically, there is no pixel alignment between the two streams. Concerning the depth sensor, two types are more commonly available:

Figure 2.10: Views of my captured head model using a single RGB-D capture device.

Figure 2.11: Views of a point-cloud created from a single RGB-D device.

Figure 2.12: Views of my reconstructed head using a system I developed [3].

structured light [43] and time-of-flight (ToF) based [44]. The Kinect 1 RGB-D camera, for example, is structured light based, while the sensor of the Kinect 2 is time-of-flight based. Figure 2.13 shows two models of Kinect cameras on a tripod.

It is worth mentioning that the Kinect for Xbox 360 (Kinect 1) was the first widely available RGB-D sensor. The Kinect 1 hardware, shown in Figure 2.14, comes with a RGB camera, an Infrared (IR) projector and an IR camera. The depth sensor is based on the structured light principle and is composed of an IR projector combined with an IR camera. The IR projector projects a known pattern of IR dots to a scene and the IR camera captures this projected pattern. By comparing the projected with the captured IR pattern, the sensor can obtain the depth information [45]. The output of the sensor is

Figure 2.13: Kinect 2 (left) and Kinect 1 (right) assembled on a tripod, in the configuration used for capture experiments.

transmitted over the USB 2 interface and is composed of a bayer-pattern chroma subsampled 8 bit/pixel RGB stream and a 11 bit/pixel depth stream. In the typical operating mode, both streams have 640x480 pixels at 30fps.

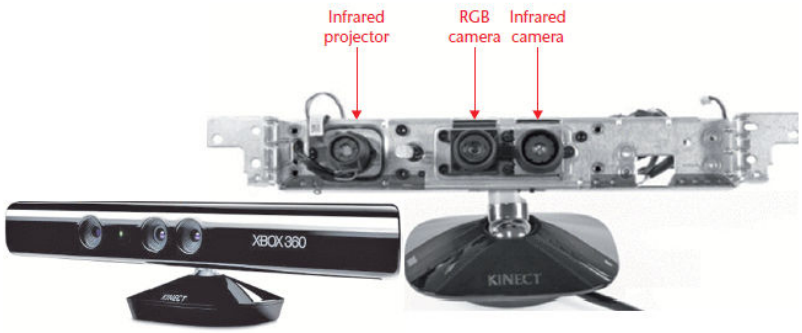


Figure 2.14: Kinect 1 hardware, with it's Infrared projector, RGB camera and Infrared camera.

Kinect for Xbox One, or just Kinect 2, is a time-of-flight (ToF) based device greatly used in the past decade for RGB-D capture. Kinect 2, shown in Figure 2.15, has a 1920x1080 HD RGB camera and uses a different depth sensing technology when compared to the Kinect 1. Kinect 2 has an IR light source which emits a modulated square wave,

and an IR receiver which captures the reflected wave. Through the phase analysis of the received signal the sensor can compute the depth. Kinect 2's time-of-flight depth sensor outputs a 512x424 depth frame, which together with the RGB information, is transmitted over the USB 3 bus to a host computer. The Kinect 2 depth sensor has better accuracy and less noisy output than the structured light based Kinect 1 [46].

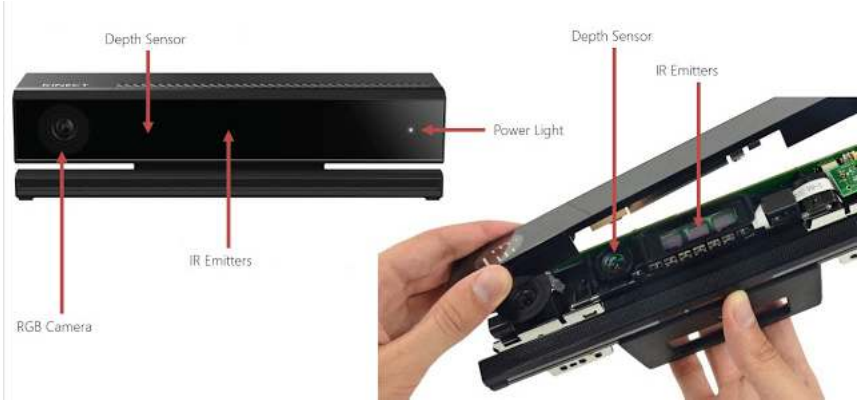


Figure 2.15: Kinect 2 hardware, with it's IR emitters, depth sensor and RGB camera.

Recent launches include the Azure Kinect, which evolves the Kinect 2 ToF camera with an improved ToF sensor [47] which can capture depth frames up to 1024x1024 of resolution, paired with a 3840x2160 "4k" RGB camera. The Azure Kinect has a working depth range of 0.25 m up to 5.46 m, selectable field-of-view of 75° by 65° (HxV) or 120° by 120°, and exposure ranging from 12.8 ms up to 20.3 ms, depending on the operating mode, with standard error deviation equal or less than 17 mm in the operating range.

Since 2019 all major smartphone manufacturers started to include ToF camera in their high-end phone and tablet models [48]. Figure 2.16 shows a partial view of a Huawei P40 Pro+ phone, which contains a ToF camera and five photographic cameras with different type of lenses and light spectrum capture ranges, manufactured by German company Leica Camera AG.

2.1.2 Display

With the availability of light field displays and head-mounted displays with ultra-high definition near-eye display [49], the presentation of visual immersive media became possible. Volumetric visual content can be experienced in head-mounted displays (HMD), 3D flat screens or through holographic projectors. While holographic and light-field displays do exist, they are much more expensive than HDMs, which are the most used apparatus for visual immersive experiences in current days. Typically, HMDs are classified in two types

Figure 2.16: Huawei phone with a ToF depth sensor and other photographic sensors developed by Leica Camera AG.

depending on the experience provided: Virtual Reality (VR) or Mixed Reality (MR). The VR type presents the volumetric graphics in a screen inside an opaque device, in which there is no blending of real and virtual images. The MR type, also named Augmented Reality (AR) in some contexts, allow the users to visualize both real world and computer generated content, as shown in Figure 2.18.

Figure 2.17: User with a Oculus Rift VR glass interacting with another user, while being captured by a setup of three (one hidden) RGB-D Microsoft Kinect sensors. A synthesized scene with the two users is shown in the TV in the back.

In order to create a mixed reality experience, interactivity is paramount, and a HMD needs to provide 6 degrees of freedom to the user. This requires additional sensors to track the head position and the eyes of the user. Also, it is necessary to have sensors to map the external world, which allows the HMD device to know where to project a

Figure 2.18: Demonstration by Microsoft of a user viewing a remote located kid through a MR device. Top left shows the real scene and bottom left the way the scene is viewed in the Microsoft's MR HMD, called HoloLens.

volumetric element in the field of view of a user, and in a sane real-world location. To present a smooth and realistic blend between real and virtual worlds, mixed reality HMD devices must account on user's localization, eyes and head tracking, and external world mapping, allowing a user to have infinite ways to visualize a scene, given that 6 degrees of freedom are allowed. An example of HMD with such features is illustrated in Figure 2.19

Figure 2.19: Magic Leap One mixed reality head-mounted device. In the image it's possible to see some of the many sensors the device has, including depth sensor.

Immersive video, also called volumetric video, has many use cases, for example a live multi-party volumetric video conference, where each person in the conference has its own volumetric video captured and transmitted, while receiving volumetric streams from other participants (eg. Microsoft's Holoportation [50]). It is worth pointing out that volumetric imagery allows a more accurate representation of the world, being extremely useful for industrial, medical, educational, gaming and recreational applications, among other purposes.

2.2 Point Cloud Quality Assessment Overview

Considering the increased popularity of the use of point clouds for immersive media and the development of high-compression encoders, the need to evaluate the quality of compressed point-clouds to a human viewer is important. The quality assessment methods are very important to guarantee a good quality of experience (QoE) of point-clouds and the acceptable levels of degradation imposed by the recently developed and future PC coding tools.

Metrics which assess the quality of a media can be divided in two main types: subjective or objective. Subjective metrics are the ones based on humans evaluating a content and giving scores based on the subjective quality perception. Subjective metrics are typically used when precise results are desired, for example, for decision making on the quality of lossy encoders. Also, subjective metrics provide a ground truth for the development and comparison of the other type of metric - the objective metric. Objective metrics are the ones which evaluate the quality of content in an automatic way, based on algorithms which optimally have the higher possible correlation with the subjective ground truth. This section is split in two, one for the subjective PC quality assessment overview, and another for the objective PC quality assessment methods overview.

2.2.1 Subjective Quality Assessment

The subjective PC quality assessment methods re-use many protocols used to 2D image and video quality analysis, as described in the ITU-R BT.500-14 [51]. Currently there is no specific standard or recommendation for PC subjective quality evaluation [52], so questions related to interactive PC visualization (active or passive) and different PC rendering methods available prior display still cause uncertainties for PC subjective quality assessment. Nevertheless, available publications in the literature provide sufficient subjective data to provide a better understanding of the human perception of PC color and geometry visual impairments. Point cloud quality assessment (PCQA) methods proposed up to date use standardized methodologies for subjective quality assessment, for example, the

Absolute Category Rating (ACR) and Double Stimulus Impairment Scale (DSIS) [53, 54]. In the ACR the subjects rate the quality of each content independently, being classified as a single-stimulus method. In DSIS, a double-stimulus method, the viewer sees the reference visual content and the impaired content, and then attributes a score to the impaired content. The scores of the viewers are typically mapped to the Mean Opinion Score (MOS) scale, ranging from 1 (very bad) to 5 (very good).

Several subjective quality assessment experiments were carried up to date. Zhang at al. [55] carried subjective quality tests with PCs using different levels of degradation of both color and geometry. The types of degradation applied were geometry down-sampling and uniform noise added independently for color and geometry. The work suggests that human perception is less tolerant to geometry degradation than color degradation in PCs. Also Torlig [56] performed subjective tests of PCs with color and geometry, with degradation to the color texture generated by an JPEG encoder and geometry affected by reducing the octree resolution.

Mekuria et al. [57] conducted PC subjective quality assessment experiments with real-world captured content and computer graphics generated content. The work evaluated two types of degradation, octree pruning for geometry, and JPEG for color, with the users were asked to evaluate the interactive immersive experience, including different quality analysis aspects, like realism, immersiveness and color quality.

Javaheri et al. [58] carried subjective and objective PC quality experiments with impulse noise and Gaussian noise at different intensities to geometry-only PCs, while also testing different techniques of denoising and surface reconstruction for rendering to 2D displays. In other work [59], Javaheri et al. performed another subjective and objective quality evaluation experiments with octree and graph-based PC codecs to create the test dataset. In both work, observers assessed the quality through 2D displays and the experiment used the Double-Stimulus Impairment Scale (DSIS) subjective quality assessment.

Alexiou et al. performed many studies. In [60], subjective quality analysis is carried with two types of degradation, octree pruning and Gaussian noise, applied to the geometry. Augmented reality googles were used by the subjects and just the geometry degradations were evaluated, with PCs without their original color texture. The traditional point-based objective metrics evaluated performed well for Gaussian noise but under performed for octree pruning compression artifacts. In other works [54, 61], Alexiou et al. used the same content of previous experiments, but with the observers experiencing the content through 2D display, also without color texture but without any interactivity. Also both Absolute Category Rating (ACR) and DSIS subjective methodologies were evaluated, with the DSIS methodology found to be better with lower confidence intervals. In [62], Alexiou continues to improve the subjective experiments by testing different reconstruction algorithms to

obtain a mesh representation prior to render.

In [63, 62], subjective tests were carried for comparing the perception of quality over different types of display, including 2D display, stereoscopic 3D displays and an AR headset. Only geometry artifacts were considered in the study, and it was found a high correlation between the human perception of distortions across different visualization devices. It was also noted that the rendering method of a PC may influence the subjective evaluation results. Finally, in [64], Alexiou et al. carried a quality assessment analysis in which the distortions were obtained using both MPEG codecs [5, 22] adjusted with different parameters.

Christaki at al. [65] performed subjective experiments in which observers used virtual reality head-mounted display to assess the content. 3D meshes degraded with different codecs for mesh compression were evaluated. He concludes that available objective mesh metrics have a low correlation to the subjective scores, while also pointing that the surface reconstruction type used influence the performance of the tested objective metrics. Dumic et al. [52] presented an evaluation of the PC subjective quality evaluation methods and the available PC objective metrics. Recently, Perry et al [66] presented a research with a new dataset which contains a comprehensive analysis of the distortions caused by the MPEG volumetric media encoders to the human visual system.

Yang et al. [67] created one of the most complete dataset of PCs and subjective data in terms of different types of distortions, including different impairments and combinations of them: octree-based compression, color noise, downscaling, downscaling plus color noise, downscaling plus geometry gaussian noise, geometry gaussian noise and color noise plus geometry gaussian noise. Yang also evaluated in the same article some objective PCQA metrics.

Clearly the work on PC subjective quality assessment available is enough for the development and improvement of PC objective quality analysis, while some questions on the rendering of the PC before presentation still remain as an uncertainty variable in the subjective PC quality analysis and the scores obtained though them.

Important datasets with associated subjective scores are summarized bellow:

- D1: Torlig 2018 [56]: This database has 6 reference PCs, which include 3 human bodies: RedAndBlack, Loot and LongDress and 3 inanimate objects: Amphoriskos, Biplane and RomanOilLamp. It includes 54 test PCs, impaired at 9 distortion levels. Distortions were produced using an octree-based codec, with color attributes encoded using the JPEG encoder at different quantizer levels. Subjective scores were obtained from experiments carried in two different universities (UnB & EPFL).
- D2: Alexiou 2019 [64]: This database has 8 references and 232 test PCs and its contents contain the objects Amphoriskos, Biplane and RomanOilLamp, the full-

bodies LongDress, Loot, Soldier and The20sMaria, and also a PC with just a human head. The distortions included in this database were generated by the MPEG codecs, namely the video-based point cloud codec (V-PCC) and four variants of the geometry-based point cloud codec (G-PCC). The variants of G-PCC include the Region-Adaptive Hierarchical Transform with Trisoup (RAHT-Trisoup), RAHT with Octree (RAHT-Octree), Wavelet/Lifting-based with Trisoup (Lifting-Trisoup), and Wavelet/Lifting-based with octree (Lifting-octree). Subjective experiments were carried in two universities (UnB & EPFL).

- D3: Stuart 2020 [66]: This dataset contains 6 PC references and 107 test PCs, namely human full-bodies LongDress, Loot, RedAndBlack and Soldier, and partial single-view captured PC upper bodies, Ricardo and Sarah. Tests use the MPEG encoders, in the variants V-PCC and G-PCC in both Octree and Trisoup variants. Subjective quality analysis was performed by 4 Universities, namely UBI, UC, UNIN and UTS.
- D4: Yang 2020 [67]: This dataset contains 9 PC references and 378 test PCs: human full-bodies - Hhi, LongDress, Loot, RedAndBlack and Soldier, also notable objects - RomanOilLamp, Shiva and Statue, and finally a small scene with many objects in a table, including a toy unicorn, named UBL_unicorn. The distortions are of 7 different classes, and are applied in 6 levels each. The distortion classes are: Octree-based compression, Color Noise, Downscaling, Downscaling plus Color noise, Downscaling plus Geometry Gaussian noise, Geometry Gaussian noise and finally Color noise plus Geometry Gaussian noise.

To illustrate the content in each dataset, Figure 2.20 shows PCs 2D projections of all the 4 datasets.

2.2.2 Objective Quality Assessment

Objective PC Quality Assessment (PCQA) methods provide a way to predict the human perception of a PC in an automatic way, without human intervention. Three types of objective quality assessment metrics exist with relation to the required reference information: full-reference (FR), reduced reference (RR) and no-reference (NR). The full-reference methods use the whole content reference in order to estimate the quality. Reduced-reference methods use only partial information from the reference, while no-reference methods asses the quality of the visual stimuli without any reference information. Figure 2.21 shows the different types of objective metrics with regards to reference data.

Since PCs are a new immersive media format, most of the proposed PCQA methods up to this time are full-reference (FR), as having all the data from reference and test content

Figure 2.20: 2D captures of the datasets D1, D2, D3 and D4, two of each, from left to right, top to bottom, respectively.

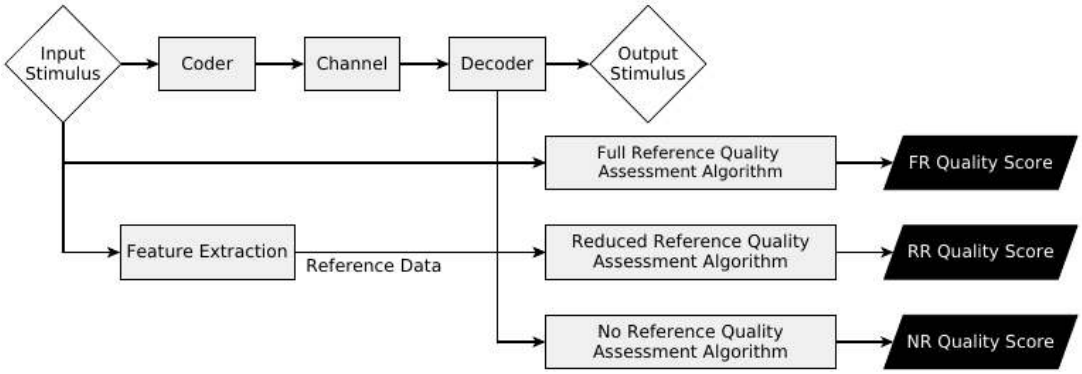


Figure 2.21: Objective quality assessment methods: full-reference, reduced-reference and no reference as shown by Freitas [4].

allows algorithms to work without prior content knowledge. The foundations of this new class of quality assessment methods were established by Tian et al. [68] and Mekuria et al. [69], with the introduction of the so-called point-based metrics. Their proposal included two types of point-based metrics, which could be point-to-point (P2Po) [69] and point-to-plane (P2Pl) [68]. Alexiou et al. [70] proposed a metric which evaluates just the geometry of PCs, without color attributes being considered. Alexiou's proposal introduced the plane-to-plane (Pl2pl) [70] variation of the point-based metrics, which uses the angular distance between tangent planes of each pair of points of a reference and test PCs to estimate the perceptual distortion.

Point-based methods establish a relation between each point in a reference PC and its nearest neighbor in the degraded PC and, then, some distance measure is used to estimate the error in the degraded content. The distances used are typically Euclidean or

Hausdorff. The accumulated error of all PC points is used to obtain a global error measure given a pair of reference and test PC. The distances are calculated in a symmetric way, meaning the nearest neighbor algorithm is used once to find correspondent points of the reference PC in the test PC, and then the other way around, with the distances calculated in both cases, with the maximum error used as final result.

Three types of error calculation were proposed between the points: point-to-point (P2po), point-to-plane (P2pl) and plane-to-plane (Pl2pl). These metrics can be used to estimate geometry or color impairments of PCs. In all cases, first a point-to-point correspondence is established between each point in a reference point cloud R and a degraded point cloud D. The point correspondence of one point k in R to D is found using a nearest neighbor search in D considering the coordinates of k. The geometry error in the point-to-point methods is based on the distance of each point k in D and its correspondent i in R, as expressed as:

$$\text{d(k,i)} = \sqrt{(k-i)^2}. \tag{2.1}$$

After the point-to-point, point-to-plane or plane-to-plane distances calculation, the global error is calculated though mean squared error (MSE) or peak signal-to-noise error (PSNR), as shown in:

$$\text{P2po-MSE} \cdot \sum_{k=1}^{N} (d(k,i))^2 \tag{2.2}$$

and:

$$\text{P2po-PSNR} = 10 \cdot log_{10} \frac{3p^2}{\text{P2po-MSE.}} \tag{2.3}$$

In Equation 2.2 N is the number of points in point cloud R, k is a point in R and i the correspondent point in D. In Equation 2.3, p is the peak distance, and is typically obtained by $2^{pr} - 1$, where pr is the dynamic range of the PC coordinates, in bits [53]. The final score of the metric is the maximum error between the calculation of R to D, and vice versa.

P2po-based metrics are also used for color error estimation, and uses the same steps of the geometry variant for point-to-point correspondence, but it uses color distances between correspondent points to estimate the error. The color information of the PC points is first converted YCbCr color-space, typically using the ITU-R Rec. BT.709 colorimetry equations, and then, for each of the three color channel components, the MSE and PSNR measures are computed, as presented in:

$$\text{P2po-color-MSE} = \frac{1}{N} \cdot \sum_{k=1}^{N} (C(k) - C(i)))^2 \tag{2.4}$$

and:

$$\text{P2po-color-PSNR} = 10 \cdot log_{10} \frac{p^2}{\text{P2po-color-MSE,}} \tag{2.5}$$

in which $C(k)$ represents a color channel of point k, for example, Y, Cb or Cr (or R,G,B), while p in Equation is the peak signal level, which is 255 in 8 bit per component systems or more in higher definition range image systems.

A common method for obtaining a single value for color error measurement was proposed by Ohm et al. [71], and shown below:

$$E_{YCbCr} = \frac{6 \cdot E_y + E_{Cb} + E_{Cr}}{8}, \tag{2.6}$$

in which E can be the MSE or PSNR errors as presented in Equation 2.4 and 2.2.2.

The other type of point-based metrics are the ones which use plane information. P2pl or Pl2pl metrics are based on the fact that a PC represents surfaces given a set of points. P2pl method relies on the same P2po distances, but the distances are multiplied by the projection of a correspondent point i in a degraded PC to the plane perpendicular to the normal of the point k in the reference PC, while the Pl2pl uses the differences of the angular similarity between the tangent planes of point k in R and the tangent plane of its correspondent i in D. In order to obtain the local planes information, a pre-processing of the PCs needs to be done in order to calculate the normal vectors of each point, considering a local neighborhood [68]. The normal vector of each point represents a local surface plane - points on the degraded PC closer to the reference local surface will lead to smaller errors even if far from the reference point. Equation 2.7

$$\text{P2pl-MSE} = \frac{1}{N} \cdot \sum_{k=1}^{N} (d(k,i) \cdot n_g)^2 \tag{2.7}$$

shows the MSE error formula, similar to Equation 2.4, but with the geometric divergence n_g multiplying the point-to-point distance.

Javaheri et al. [58] and Perry et al. [66] conducted subjective quality experiences that tested their datasets and subjective data with the available point-based metrics, concluding that the objective PCQA point-to-plane metrics using the MSE distance is the one that better represents human vision quality perception. Distances other than PSNR and MSE were also proposed, for example, Javaheri [72] proposes the use of the Hausdorff distances.

Nowadays, metrics based on P2Po, P2Pl and Pl2Pl methods, are widely used by MPEG [73], and still considered the state-of-the-art by Javaheri [53]. However, these metrics have several drawbacks and inefficiencies mainly because it is difficult to establish accurate correspondences between the two PCs due to their unstructured nature and the different

types of coding errors, which are not properly sensed by the available metrics. These metrics also lack a unified metric which considers both color and geometry distortions.

Nevertheless, most present-day objective PC quality assessment research compare new methods with the point-based metrics, not only because of the good performance of the metrics for some types of distortions, but also because the C++ source code of these metrics are freely available and are easy and run and adapt [3]. Another approach was adopted by Torlig et al. [56], which proposes the use full-reference 2D images metrics to assess PC 2D projections to the 6 faces of cube containing a 3D object. 2D image metrics tested include PSNR (and variations), SSIM, MSSIM and VIFP metrics. The metrics with best performance were MSSIM and VIFP.

Recently, metrics which consider both color and geometry were introduced, and they often surpasses the performance of point-to-point based metrics. Viola et al. [74] proposes to use distances of histograms with both color and geometry statistics for PC quality estimation. Results compared the MPEG PCQA metrics [75] with the proposed method just with one dataset [64], which contains test PCs generated exclusively with MPEG PC encoder, showing equal or better performance than point-based metrics.

Javaheri et al [76, 72] propose two new geometry PCQA metrics. In [76] the use of the Mahalanobis distance between histograms of test and reference PC is proposed. The histograms are created from the frequency of angular differences among tangent planes of each point of a PC and some neighbours. In [72] it is proposed the use of the generalized Hausdorf distance to improve over prior work. Only one dataset and subjective results were used to compared both propoals with other metrics, and small improvements are shown compared to other geometry PCQA methods.

Meynet et al. [77] proposes a geometry PCQA metric based on a geometry quality assessment metric for mesh. Local distortions are calculated based on normal surface differences of an spherical neighborhood of each point. Authors claim superior performance of the proposed metric when compared with MPEG metrics, but just one dataset [54] and associated subjective data is used, limiting the confidence of the results. Yang *et al.* [78] proposes one of the first graph-based PCQA metrics which use graph-based relations among points in the PC to estimate quality, providing promising results, while requiring extensive and complex graph operations.

More recently, Alexiou et al. [79] proposed a PCQA based on local features extraction, called PointSSIM. The metric tries to emulate the behavior of the 2D metric called SSIM, and presents promising results when compared to other metrics in the 2 datasets selected for evaluation. Contemporary to Alexiou, Meynet *et al.* proposed a metric that also takes into consideration geometry- and color-based features, using logistic regression to combine

[3]dmetric v0.13.5: http://mpegx.int-evry.fr/software/MPEG/PCC/mpeg-pcc-dmetric.git

these features and produce a quality estimate called PCQM [80]. PointSSIM and PCQM are considered to be the state-of-the-art, by analyzing their published results.

While most of the work on PCQA are full-reference proposals, Bello et al.[81] provided a review on the recent use of deep learning in 3D vision tasks, including classification, segmentation, and detection, pointing out that local point relationships are more effective for modeling a PC data-driven approach. Liu et al. [82] proposed the first no-reference method (NR) for PCQA method which uses a data-driven approach and applies a convolutional neural network (CNN).

Apart of the state-of-the-art taxonomy which classifies the metrics in the point-to-plane or point-to-point framework, another important classification is based on the information each algorithm uses. Considering this concept, three types of PCQA methods exist: color only, geometry only, or joint color and geometry methods. Most of the available metrics evaluate only the geometry or color information of PCs separately, with just a few publications jointly considering the color and geometry information. Examples include the works by Viola et al. [74], Meynet [80] and this present work, as discussed in later chapters of this thesis.

Chapter 3

Color And Geometry Textures For Point Cloud Quality Assessment

The contributions to the state-of-the-art of point-clouds objective quality assessment are presented in this chapter. The first section contains all the innovative PC texture descriptors proposed, including the pre-processing voxelization technique to increase the quality assessment performance of the proposed color-based texture descriptors. Also, a novel geometry-based texture descriptor is proposed, which does not use the voxelization technique to obtain good performace. The second section contains the distance measures used to compare the statistics of the proposed texture descriptors applied to reference and degraded point-clouds. The third section of this work describes the quality assessment model used for the proposed point-cloud quality assessment (PCQA) metrics, based on the presented texture descriptors.

3.1 PC Texture Descriptors

In this section, we present innovative texture descriptors for PCs, which use local and global statistics that can be used to describe local and global characteristics of PCs. For the purpose of this work, these new texture descriptors are used for quality estimation. This section is split in subsections: one describing the voxelization pre-processing technique, followed by three color-based texture descriptors, and one geometry-based texture descriptor.

3.1.1 Voxelization

Point-clouds are data structures composed by a list of points containing color components (e.g. RGB, YCbCr, L*a*b*) and three geometric coordinates, typically named X, Y and

Z when using the 3D Euclidean space [83]. Sometimes other components are present, such as reflectance coefficients and normal vectors of the reconstructed tangent surface at each point. To visualize a PC, its points need to have a visual volumetric representation to be properly rendered. One option is to convert a PC to a mesh representation, in which the PC points become nodes of 2D polyhedral surfaces in a 3D space. While meshes provide a good option for PC rendering, they require complex and computationally expensive tasks for connecting the PC points to provide optimal surfaces, since point connectivity information is not captured by capturing sensors. Another option is to convert each PC point to a discrete volumetric unit. Just like 2D images typically use square-shaped pixels, in PCs the volumetric elements (voxels) typically adopt a cube shape. The voxels can be considered as discrete elements in a discrete 3D grid, but while in 2D images the discrete 2D space is dense, in PCs the 3D space is sparsely filled with voxels, which that (typically) represent just the surface of objects.

So, prior to visualization, the voxelization method is typically applied to give shape and volume to the PC points. A voxel with a cube shape is a regular hexahedron, containing 6 faces, 8 vertices, and 12 equal-sized edges. An important parameter for the voxelization is the size of each voxel. If the size is small, the neighboring voxels may not touch each other, leaving visual "holes" between PC elements. On the other hand, if the size is too big, voxels create a swollen visual effect. Figure 3.1 shows examples of a PC with 3 different voxel sizes for each of the three PC rendering examples. Important to note that, when the voxelization is applied for given a voxel size, more than one PC point might be present inside a single voxel. In this case, typically, the color corresponding to these points are averaged to provide the final color value for the voxel.

As a relatively new research area, there is no standardized process to obtain the voxel size given a PC. Voxel's size can be defined either by using a point-by-point analysis or using one voxel size for all points in the PC. Optimally, each voxel needs a volume big enough to touch the neighboring voxels. So, a different size for each voxel could provide an optimal (better) voxelization, but it would also increase the computation complexity cost. Therefore, a voxel size is generally chosen for all points of a PC. In this work, it is adopted this approach. Also, the voxelization is used as a step before applying the PC color-based texture descriptor, which is part of the proposed full-reference PC quality assessment method. In a full-reference method, reference and test PCs are compared, so instead of using a single voxel size for each PC, a single voxel size could be used for both reference and test content. Both options were considered in this work, but in order to improve generalization and to better emulate how PC visualization systems works (the rendering is optimized for the PC to be displayed), one voxel size per PC was adopted.

In order to obtain the desired voxel's edge size (ES), in this work it is proposed an

Figure 3.1: Voxelization effects, from left to right, with a too small voxel size, with a proper voxel size, and with an oversized voxel.

heuristic based on the average distance among nearest neighbors PC points, which is computed as follows:

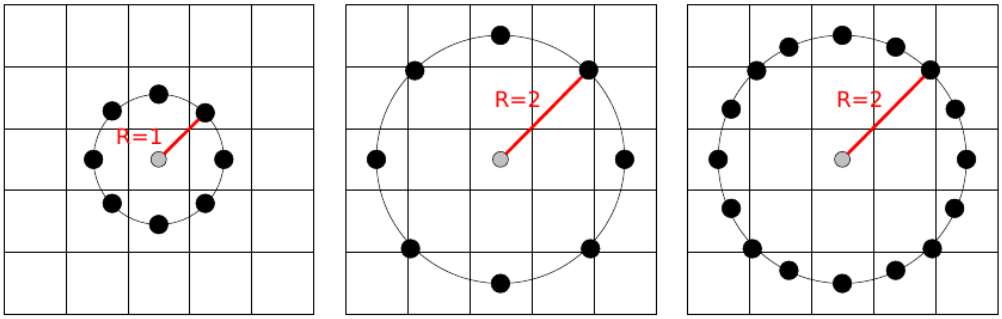
$$\mathrm{ES} = \frac{k}{S} \cdot \sum_{n=1}^{S} \left(\frac{1}{k_{nn}} \cdot \sum_{i=1}^{k_{nn}} \mathbf{d}\left(N_i(P_n), P_n\right) \right), \tag{3.1}$$

where, S is the number of points of the PC, k is a constant obtained experimentally, P_n is the n-th point of the PC, $N_i(P_n)$ are the coordinates of the i-th nearest point to P_n, and k_{nn} is the number of nearest neighbors. The function $\mathbf{d}\left(P_a, P_b\right)$ computes the Euclidean distance between points P_a and P_b. The voxel's volume is obtained computing the edge size cube (ES^3).

The proposed heuristic to obtain the voxel size is based on the average distance of nearest neighbors. The heuristic assumes that correctly sized voxels provide a visual experience without major artifacts (eg. holes), in which the voxels need to approximately touch each other. The number of nearest neighbors is typically 8, considering that an approximation of a 3D surface is a 2D image, in which every element has 8 connected nearest neighbors. The k in Eq. 3.1 is a multiplier adjusted to provide the appropriate voxel size for an optimal visual quality perception of the PC rendering or improve the performance of a texture descriptor. The properly definition of k is needed as PCs are captured using different methods, which produce PCs with distinct intrinsic characteristics, including different types of point dispersion.

3.1.2 Local Binary Patterns for PC

This section contains an adaptation of the Local Binary Pattern (LBP) to work on point-clouds contents. The LBP descriptor is a texture descriptor originally proposed by Ojala *et al.* [84] to improve the accuracy of texture recognition tasks in 2D images. This descriptor is an effective feature extractor for texture-based problems, including for quality assessment purposes [85, 86, 87]. The original LBP descriptor associates a binary code to each pixel of a given image, considering the image luminance pattern of the surrounding pixels in a defined neighborhood. The types of the considered neighbor vary and are typically defined in terms of a radius R between a target point and its neighbors, as shown in Figure 3.2. The label attribution by the LBP descriptor uses the relation of each pixel and its neighborhood, as shown in Figure 3.3, containing a 3x3 2D image.



(a) $R = 1, P = 8$ (b) $R = 2, P = 8$ (c) $R = 2, P = 16$

Figure 3.2: Some neighborhood types of the LBP descriptor extracted at a distance R, in different variations.

The value of each bit in the LBP label is computed by thresholding the differences between a target (central) pixel and its neighboring pixels. The default LBP descriptor for 2D images takes the following form:

$$\mathrm{LBP}_R^N(P_c) = \sum_{n=0}^{N-1} \theta(P_n - P_c) \cdot 2^n, \tag{3.2}$$

where

$$\theta(u) = \begin{cases} 1 & \text{if } u \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

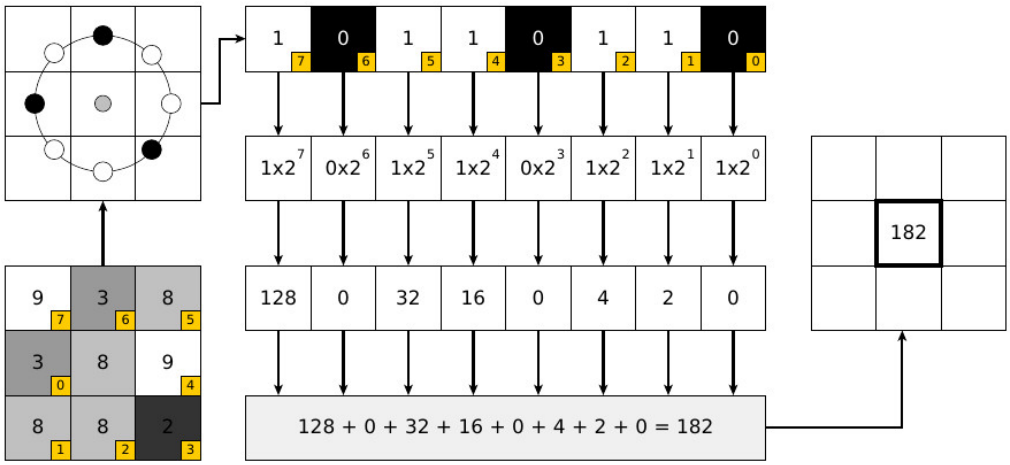


Figure 3.3: LBP label calculation for a Some neighborhood types of the LBP descriptor extrated at a distance R.

In these equations, $P_c = P(x, y)$ is an arbitrary central pixel at the position (x, y) and $P_n = P(x_n, y_n)$ is a neighboring pixel surrounding P_c, where

$$x_n = x + R\cos\left(\frac{2\pi n}{N}\right) \qquad \text{and} \qquad y_n = y + R\sin\left(\frac{2\pi n}{N}\right),$$

and N is the total number of neighboring pixels P_n, sampled with a distance R from P_c. Finally, after the calculation of the LBP labels, a histogram is calculated, as shown in Figure 3.4, which is then used as input to a image quality assessment model.

(a) Lena image. (b) LBP map. (c) LBP histogram

Figure 3.4: LBP application to image in (a), the corresponding LBP labels map, and the histogram of the labels in (c).

The LBP descriptor defined in Eq. 3.2 is designed for 2D images and operates considering the target pixel and a set of neighboring pixels, which are determined by a distance

R. In 2D images, these neighbors can be sampled according to a geometric distribution in a 2D plane (e.g., circular, elliptical, etc). This sampling approach works because pixels in 2D images are equally distributed in a dense 2D grid. However, in PCs, the points are sparsely distributed in the 3D space, which makes the problem of determining the neighborhood for a LBP descriptor more complex. The challenge of dealing with point sparsity of PCs is dealt with the voxelization procedure, which optimally creates a neighborhood of discrete elements which are close enough to provide visually solid objects. Another challenge is the selection of the traversal order of neighbors, as the 2D equations for determining the traversal order do not apply in 3D domain. To solve this last challenge, a different approach is adopted, in which the distances between each neighbor and the central element is used to determine the traversal order, for example, closest to farthest.

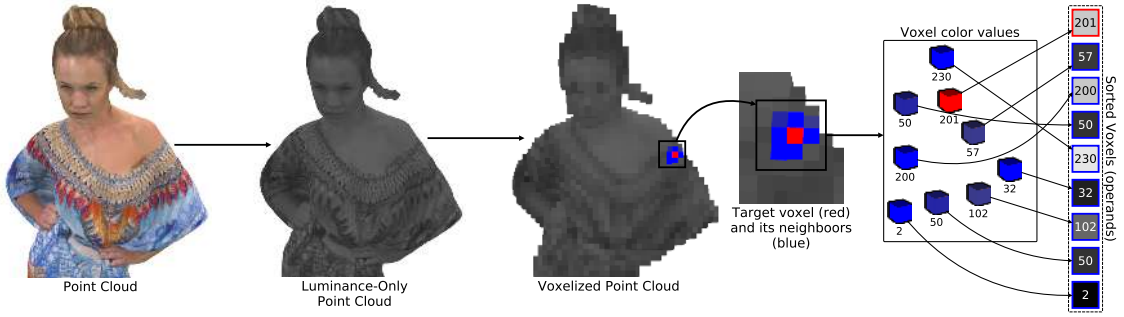


Figure 3.5: Diagram the LBP adaptation for PCs, containing, from the left to right, conversion from RGB to gray-scale, voxelization and the selection of one voxel and its 8-neighborhood.

Fig. 3.5 depicts the application of the LBP for a target voxel $P(n)$ in the 3D space, showing the sampling of the nearest voxels to create the LBP neighborhood. The neighborhood is visited from closer to farther points, which results in a performance that is slightly better than the reverse order (as shown later in the text). This figure illustrates the case where the neighborhood of the target voxel has 8 voxels. Figure 3.6 illustrates how the LBP label is obtained for a given PC element, considering example luminance values for the target element and neighbors. The resulting LBP labels of the PC compose the PC "Feature Map" (FM). In other words, each label $L(n)$ in this FM corresponds to the local texture associated with the voxel $P(n)$. The labels (FM) of the voxels have the size in bits equal to the selected number of neighbors, for example, 8 bits, which means 8 neighbors is used. After the extraction of the FM of a PC, a histogram is calculated to allow the evaluation of a PC visual quality in terms of their texture statistics. The histogram of the labels of the texture descriptor (the FM), is obtained by the following

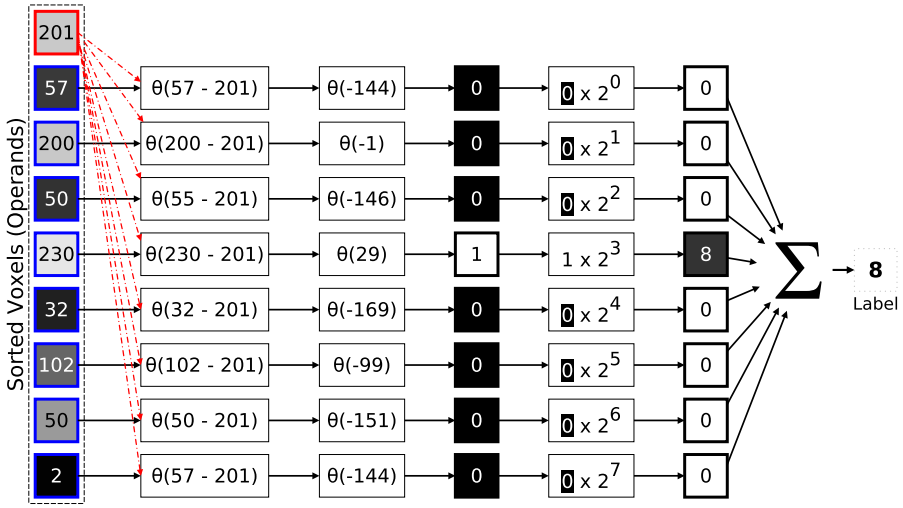


Figure 3.6: Diagram of the LBP label computation with the sorted voxels from closer to farther.

expression:

$$h = \{h[l_0], h[l_1], h[l_2], h[l_3], \cdots\}, \tag{3.4}$$

where h represents the histogram and $h[l_j]$ is the frequency of the label l_j. This label frequency is computed using the following equation:

$$h[l_j] = \sum_{n=1}^{S} \delta(L(n), l_j), \tag{3.5}$$

where

$$\delta(v, u) = \begin{cases} 1, & v = u \\ 0, & \text{otherwise}, \end{cases} \tag{3.6}$$

and S is the number of voxels of a PC.

The histogram of the FM obtained for a PC with the application of the proposed LBP can then be used in a PCQA methods, as described later in the text. Also the performance of the LBP for PCs is present in the chapter with the results.

3.1.3 Local Luminance Patterns

The Local Luminance Patterns (LLP) descriptor is a contribution to the state-of-the-art in PC texture descriptor and is used to extract statistics of a PC that are sensitive to color-based texture degradation. The idea behind this descriptor is to obtain luminance patterns that are representative of intrinsic PC texture characteristics and that can be

useful for quality estimation. A previously, the statistics of the patterns are compared between reference and test PCs. Just like as other color-based PC descriptors we propose, the voxelization is applied prior the descriptor in order to better emulate the rendering process and provide better performance when used in PCQA metric, as demonstrated in the chapter with the results.

The LLP application first obtains the luminance (Y) of each PC point color by converting RGB (typically) to gray using the ITU-R BT.709 colorimetry formula. Then, for each voxel, a neighborhood is defined. Figure 3.7 shows an example, for the given target voxel (shown in red in the third image), where a set of associated neighboring voxels are selected. The only difference between LLP and LBP in the neighborhood data usage, is that instead of the neighborhood being ordered by the spatial location of the neighbors, the LLP does not need consider the spatial topology to traverse the neighborhood voxels. Each of the voxels has a Y value ranging from 0 to $2^b - 1$, where b is the number of bits used to represent the luminance PC voxel, typically 8 bits. In the example in Figure 3.7 the target voxel (in red) and its neighbors have luminance values equal to 35, 27, 118, 59, 114, 113, 137 and 71.

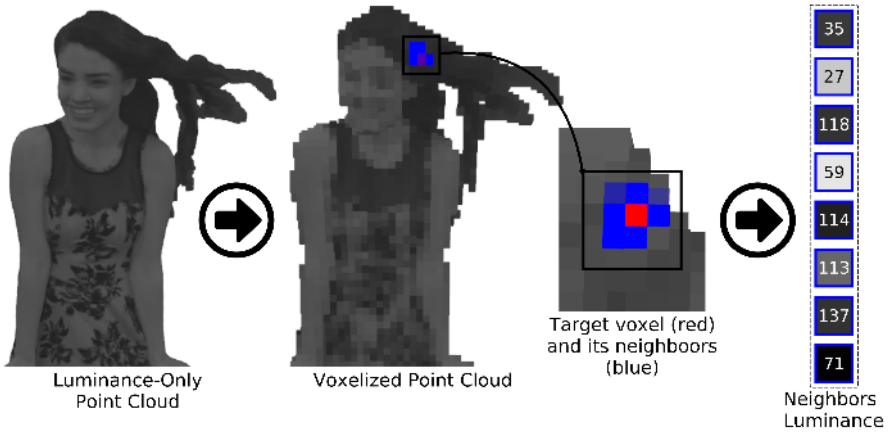


Figure 3.7: Diagram of the LLP label computation with a set of neighbor voxels.

The LLP descriptor maps the luminance values of N neighboring voxels, for each voxel, into a label of B bits. Each luminance interval is represented by a single bit in the label and the resulting bits are combined to form a label of B-bits. In other words, the LLP descriptor associates a label L of B bits, initially set as zero, to each target voxel $P(n)$. Then, for each $Y[i]$ luminance value of the i-th neighbor, it is applied iteratively

the following equation:

$$L = \begin{cases} L \vee (1 \ll \lfloor \frac{Y[i]-15}{15} \rfloor), & \text{if } 15 \leq Y[i] < 240; \\ L \vee (1 \ll 15), & \text{if } 240 \leq Y[i] \geq 255. \end{cases} \tag{3.7}$$

or

$$L = \begin{cases} L \vee (1 \ll \lfloor \frac{Y[i]-20}{20} \rfloor), & \text{if } 20 \leq Y[i] < 240; \\ L \vee (1 \ll 11), & \text{if } 240 \leq Y[i] \geq 255. \end{cases} \tag{3.8}$$

.

The symbol $\vee$ is a bitwise OR and $\ll$ is a bitwise left shift. The darker the neighbor, a smaller label value (a less significant bit) is added to the label, and vice-versa. Equation 3.7 contains the case for when the label size B is 16 bits. The 12 bit version is described in Equation 3.8.

Considering there is no dependency on the scale and position among each set of neighboring voxels for the LLP calculation, this descriptor is scale and rotation invariant. The LLP texture descriptor was tested with varying parameters, including different neighborhood sizes, voxelization parameter k (as in Eq. 3.1) and label sizes. As an example of the LLP label extration, Table 3.1 provides the label calculation for the voxel luminance values in Figure 3.7, containing a set of 8 neighbors related a target voxel. Equation 3.7 is used iteratively to calculate the label, with each iteration represented by each line of Table 3.1. The final label L is shown in the rightmost bottom cell of the table, in bold.

Table 3.1: Example of LLP label calculation with the luminance values from Figure 3.7.

Neighbor (i)	Y[i]	Bit Set	Label (accumulated)
0	35	1	00000000 00000010
1	27	0	00000000 00000011
2	118	6	00000000 01000011
3	59	2	00000000 01000111
4	114	6	00000000 01000111
5	113	6	00000000 01000111
6	137	8	00000001 01000111
7	71	3	**00000001 01001111**

Notice that, for each neighbor luminance value, a corresponding bit is set. Luminances smaller than 15 has no interval associated, and thus do not alter the label, but indeed also represent a different symbol, the '0'. If no voxel values correspond to a specific interval, the interval bit is set to '0'. If one or more voxel values correspond to a given interval, the interval bit is set to '1'. The label value L in the considered example and shown in Table 3.1 is 0000000101001111, or $14F$ in hexadecimal. The resulting label L describes the an intrinsic relation among a set of neighboring voxels.

3.1.4 Local CIEDE2000 Patterns

The Local CIEDE2000 Pattern (LCP) is an innovative color texture descriptor based on perceptual color difference patterns among voxels and its neighbors. To calculate the color differences, the CIEDE2000 (CIELab ΔE 2000) [88] color distance metric is used. The CIEDE 2000 is more advanced than its predecessor color-difference metrics CIELAB ΔE^*ab and CIE944, providing perceptually uniform color distances. The CIEDE2000 distance uses the CIELab color space, which has 3 channels: $L*$ for lightness, $a*$ for green-red opponent colors, and $b*$ for blue-yellow opponent colors, while for the color distance calculation some adjustments are made to compensate perceptual nonlinearities of the CIELab color space (which was the reason in first place for an updated color difference formula, the CIEDE2000).

Color distances in RGB or YCbCr color spaces do not have a linear correlation to the perception of color differences by the human vision. The LCP texture descriptor addresses this problem by creating patterns based on color differences provided by CIEDE2000, which are linearly related to the color differences as perceived by the human eyes.

In the LCP, for each voxel P_n, the CIEDE2000 distances are computed between the voxel and each of its N-nearest neighbors voxels P_i. Then, based on these distances, we compute a label of B bits for each PC voxel. The label L for each voxel is calculated by computing, for all N neighbors, the CIEDE2000 distances $C[i]$ corresponding to each i-th neighbor. First, we set L equal to zero. Then, for each of the N neighbors, the following equations are applied iteratively, for the case where B is 8 bits or 12 bits respectively:

$$L = \begin{cases} L \vee (1 \ll \lfloor \frac{C[i]-2.5}{2.5} \rfloor), & \text{if } 2.5 \leq C[i] < 20.0; \\ L \vee (1 \ll 7), & \text{if } C[i] \geq 20, \end{cases} \tag{3.9}$$

or

$$L = \begin{cases} L \vee (1 \ll \lfloor \frac{C[i]-1.5}{1.5} \rfloor), & \text{if } 1.5 \leq C[i] < 18.0; \\ L \vee (1 \ll 11), & \text{if } C[i] \geq 18, \end{cases} \tag{3.10}$$

where the symbol $\vee$ is a bitwise OR and $\ll$ is a bitwise left shift.

After all neighbors are analyzed, a final B-bits label L is obtained with the bits corresponding to the distances $C[i]$ to these neighbors being set. Important to note that CIEDE2000 C distances smaller than 2.5 (for the 8 bits version) or 1.5 (for the 12 bits version), which are values close to Just Noticeable Difference (JND) threshold [88], do not set any bit in the label, meaning that a neighbor has approximately the same perception of color of the target voxel. CIEDE2000 distances greater than 20.0 or 18.0 (8 bits and 12 bits version respectively) represent very different color perception between two colors, and set the most significant bit in the LCP label. Another way of visualizing the LCP

execution is through the C-language code shown in Listing 3.1 (8 bits version).

Listing 3.1: C code for the LCP label extraction for each voxel.

```
// pre-calculation of distances C[] for all N neighbors
L = 0;
for (i = 0; i < N; i++)
{
    if (C[i] >= 2.5 && C[i] < 20.0)
      L |= 1 << floor((C[i] - 2.5) / 2.5);
    if (C[i] >= 20.0)
      L |= 1 << 7;
}
```

This process generates binary frequency values for the color distance intervals, which indicate if there is at least one neighboring voxel at this interval distance. If the label L corresponding to a particular voxel is a small number, this means most neighboring voxels are similar (in color) to the central voxel. On the other hand, if L is a large number there are neighboring voxels that are dissimilar (in color) from this central voxel. Figure 3.8 shows an example of the LCP label calculation for a target point that corresponds to the 10000-th point of the "Soldier" PC sample [67]. From the left to the right, the figure shows a selection of a PC point (exemplified in red), the calculation of the CIEDE2000 distances for each of the 12 neighbors, and, finally, the label extraction. In the example, N is equal to 12. Although the voxelization step is not shown, it is used, just like in the others color-based pattern descriptors proposed in this work.

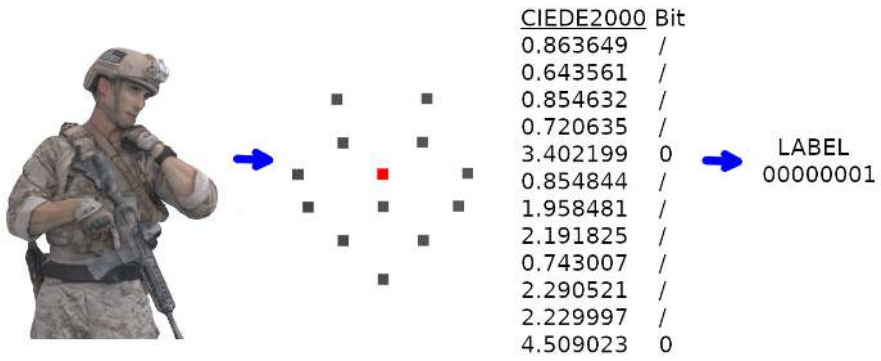


Figure 3.8: Diagram of the LCP label computation with an example of neighboring voxels.

The algorithm of the LCP is similar to the one of the LLP, but instead of using the luminance of neighbors as input, the perceptual color distance between a voxel and each

neighbor is computed. Consequently, also the intervals associated to each bit in the label has a different physical meaning.

3.1.5 Geometry-based Texture Descriptor

The PC texture descriptors proposed in this work are color-based textures descriptors: LBP, LLP and LCP. After testing these descriptors, it became clear that the geometric information of a PC also plays a role in the perceived PC quality, as already evaluated by Alexiou *et al.* [61]. While the proposed color-based descriptors can, to some extent, identify some geometric distortions, the color-based descriptors miss to represent most geometric distortions.

The proposed geometry-based texture descriptor considers the geometric information of the surface tangent to each PC point and its neighbors. In order to establish a relation between each point and its neighborhood, the normal vector information is used. The normal vector of a PC point is the vector orthogonal to the point's local surface. Since typical PC capture devices do not capture normal vectors, only depth-plus-color information being generally available, the normal vectors need to be computed prior the descriptor application.

The normal vectors are computed through the eigenvectors from the covariance matrix of the local neighborhood 3D coordinates. For each PC point, this local neighborhood can have at most 16 points, which are located inside a maximum radius of 6 times the average distance of the 8 nearest neighbors. To overcome the fact that each PC point has 2 normal vectors that correctly represent the tangent plane, we orient the normals to an arbitrary direction, in order to remove ambiguities. In our case, we oriented all PC normals to the direction $(0, 0, 1)$ and normalized the magnitude normal values to 1.

A diagram of the geometry-based descriptor is shown in Figure 3.9. For each point P_t in a PC, we define the distance between P_t's normal and each of the N-nearest neighbors P_i's normals, as the distance between two 3D normal vectors:

$$G = \sqrt{\sum_{d=1}^{3} (n_{t_d} - n_{i_d})^2},$$

where n_{td} is the normalized normal vector of point P_t, n_{id} is the normal vector of a neighbor P_i, and d represents each of the 3 dimensions (x, y, z) of a normal vector. Considering that the normalized normals range from 0 to 1, the maximum possible distance between normals is 2.

After the normal distances are computed, a label of B bits for each point is created. In the example in Figure 3.9, $B = 16$ bits and $N = 6$. For a given point P_t, its label L is

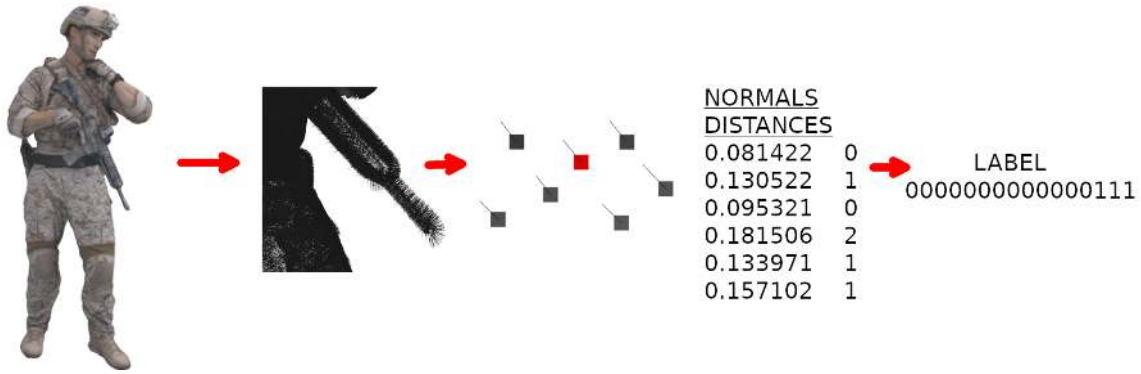


Figure 3.9: Diagram of the geometric texture label computation, with the normal vectors represented as black lines.

computed through the iteration of the distances $G[i]$ of each $i-th$ nearest neighbors of P_t as follows:

$$L = \begin{cases} L \vee 1, & 0.05 \leq G[i] < 0.10; \\ L \vee 1 \ll 1, & 0.1 \leq G[i] < 0.175; \\ L \vee 1 \ll 2, & 0.175 \leq G[i] < 0.275; \\ L \vee 1 \ll \lfloor \frac{G[i]-0.275}{0.125} + 3 \rfloor, & 0.275 \leq G[i] < 1.65; \\ L \vee 1 \ll 14, & 1.65 \leq G[i] < 1.80; \\ L \vee 1 \ll 15, & 1.65 \leq G[i] \leq 2.0, \end{cases} \tag{3.11}$$

where the symbol $\vee$ is a bitwise OR and $\ll$ is a bitwise left shift.

The geometry-based descriptor works in a similar way than the LLP and the LCP for label bits attribution, but with different ranges and texture information. The smaller the error G among a neighbor, the less significant the bit to be set in L, while larger G will set higher a significant bit in L, what means that a small geometry distortion will create smaller values in L, and vice-versa. Very small G (smaller than 0.05) do not change the bits in label L. The ranges were established considering that the maximum distances between two vectors with maximum magnitude of 1, is 2. The intermediate ranges when $B = 16$ are set to 0.125 wide, while for smaller G, the ranges are a bit smaller, in order to address the fact that most of the distances G are expected to be small. For higher G, the ranges are a bit bigger (and less likely to happen, as seem experimentally), as shown in Eq. 3.11 and also in table 3.2 as an alternative way to describe the algorithm, for B equal to 16 bits. Table 3.2 contains all the G ranges and the corresponding bit mask which is used to create the final label by the use of the logical OR bitwise operation applied to all bit masks.

Distance G	**Operation**	**Label Bit**
$G < 0.050$	no bit is set	0000000000000000
$[0.050, 0.100)$	bit 0 is set	0000000000000001
$[0.100, 0.175)$	bit 1 is set	0000000000000010
$[0.175, 0.275)$	bit 2 is set	0000000000000100
$[0.275, 0.400)$	bit 3 is set	0000000000001000
$[0.400, 0.525)$	bit 4 is set	0000000000010000
$[0.525, 0.650)$	bit 5 is set	0000000000100000
$[0.650, 0.775)$	bit 6 is set	0000000001000000
$[0.775, 0.900)$	bit 7 is set	0000000010000000
$[0.900, 1.025)$	bit 8 is set	0000000100000000
$[1.025, 1.150)$	bit 9 is set	0000001000000000
$[1.150, 1.275)$	bit 10 is set	0000010000000000
$[1.275, 1.400)$	bit 11 is set	0000100000000000
$[1.400, 1.525)$	bit 12 is set	0001000000000000
$[1.525, 1.650)$	bit 13 is set	0010000000000000
$[1.650, 1.800)$	bit 14 is set	0100000000000000
$[1.800, 2.000)$	bit 15 is set	1000000000000000

Table 3.2: Label computation for the geometry-based descriptor, with B of 16 bits, considering the different G intervals.

The proposed geometry-based texture descriptor complement the color-based descriptors in describing intrinsic features of PC, which can be used for PC Quality Assessment. The geometry-based texture descriptor has similarities to the color-based LLP and LCP descriptors, especially for the label construction, as all of them use an iterative algorithm which set bits in a label according to each neighbor, but independent of any order among the neighbors, providing rotation invariant descriptors. The main difference is that in the geometry-based descriptor, the voxelization step does not improve its performance, as it compromises and alters the geometry properties of a PC. The different behavior between color-based and geometry-based texture descriptors is expected and shows it is capturing different characteristics of a PC.

3.2 PC Texture Histogram Distances

A full-reference PCQA metric compares reference and degraded PC characteristics. In the case of the PCQA method proposed in this thesis, the metric is based on a distance between histograms of the labels extracted from reference and test PC by the application of the proposed texture descriptors. The distance measures between histograms are used to estimate the degradation of a compressed PC compared to a reference PC. A representation of a histogram is given by the following expression:

$$H = \{h[l_0], h[l_1], h[l_2], h[l_3], \cdots\}, \tag{3.12}$$

where H represents the histogram and $h[l_k]$ corresponds to the frequency of the label l_k, where k ranges from 0 to $2^B - 1$, with B the defined size of the descriptor label, in bits.

Each label frequency is computed as shown:

$$h[l_j] = \frac{1}{S} \cdot \sum_{a=0}^{S-1} \delta(L(P_a), l_j), \tag{3.13}$$

where $L(P_t)$ is the texture descriptor label of the point t of a PC, S is the number of PC points, and δ is an impulse function, as shown by the following equation:

$$\delta(v, u) = \begin{cases} 1, & v = u \\ 0, & \text{otherwise.} \end{cases} \tag{3.14}$$

After computing the histograms of the reference point cloud H_r and of the test point cloud H_t, these histograms are compared using a distance metric $\mathbf{D} = D(H_r, H_i)$. To compare histograms, several distance metrics can be used, such as Bray-Curtis, Canberra [89], Cityblock [90], Chebyshev, Cosine, Euclidean, Jensen-Shannon [91], Wasserstein [92], and Energy. Arguably the most relevant distance to modern math is the Euclidean distance, defined in the fifth postulate of Euclid's *Elements* around 300 B.C. [83]. In modern math syntax, the Euclidean distance formula applied to the histogram elements, computed as follows:

$$\mathbf{d} = \mathbf{d}(h_u, h_v) = \sqrt{(h_u - h_v)^2}, \tag{3.15}$$

which returns the Euclidean distance between histogram frequencies h_u and h_v, related to the labels l_u and l_v. Considering all the labels of the histogram of a PC, the Euclidean histogram distance between two PC histograms can be defined as follows:

$$\mathbf{d} = \mathbf{D}(Hr, Ht) = \sum_{j=0}^{2^B-1} \sqrt{(hr_j - ht_j)^2}, \tag{3.16}$$

where D is the final distance, Hr is the histogram of the reference PC, Ht is the histogram of the test PC, B is the size of the label (in bits) and hr_j and ht_j are the j-th element of the histograms Hr and Ht, respectively.

Another very important distance for quality assessment is the Jensen-Shannon divergence [91]. The Jensen-Shannon divergence is expressed in terms of the Shannon entropy, given by the following equation:

$$\mathbf{D_{js}}(\mathbf{P}||\mathbf{Q}) = S\left(\frac{P+Q}{2}\right) - \frac{1}{2}\left[S(P) + S(Q)\right], \tag{3.17}$$

where P and Q are two ordered sequences, and S is the Shannon entropy, as defined by:

$$\mathbf{S}(\mathbf{P}) = \sum_{i=1}^{M} P_i \log_2 P_i, \tag{3.18}$$

where P_i is the i-th element of a sequence and M is the size of the sequence.

The distance measure between PC histograms created by the proposed texture descriptors provide the value which is used to estimate the quality of a degraded PC. As exposed later in this text, the Jensen-Shannon divergence provides the best distance for the proposed metric framework.

3.3 PC Quality Prediction Modeling

In order to predict the quality of a given visual content, it is typical to use a regression model to estimate the perceived quality. In quality assessment methodologies, the regression model is often used to adjust the subjective quality scores provided by the different quality datasets. In the case of the PCQA methodologies proposed in this thesis, the coefficients of the regression function obtained with data from subjective experiments are used to map the distances of the PC texture descriptor histograms (described in section 3.2). The mapping can also be applied to a combination of two or more histogram distances, providing a way to jointly use different PC texture descriptors for quality estimation. The most relevant combination of color and a geometry texture descriptors are addressed in the results.

The regression algorithm takes as input the distance **D** of the histograms and maps it into an objective (predicted) quality score, using the available subjective Mean Opinion Score (MOS) values as ground-truth values. Different regression models exist to map the distances **D** into objective quality scores. Some examples include the Random Forest Regressor, Extra Trees Regressor, Gradient Boosting Regressor, Bayesian Ridge, ARD Regression, Lars, Elastic Net, Elastic NetCV, Lasso, RANSAC Regressor, KNeighbors Regressor, MLP Regressor and the Logistic function, which is recommended by an International Telecommunications Union (ITU) tutorial about objective quality assessment [93]. Regression models how the human visual system perceives the different levels and types of distortions and, therefore, how the distance metrics are mapped into predicted quality scores. As discussed in the following results' section (section 4), it is shown that the Logistic function provides a good correlation with subjective scores (these results

are also published in paper [9], and discussed in section 4.1). The Logistic function is given by:

$$Y_i^w = w_i \left[\frac{\beta_1 - \beta_2}{1 + e^{-\frac{X_i - \beta_3}{|\beta_4|}}} + \beta_2 \right] + \varepsilon_i^w, \tag{3.19}$$

where Y_i is the i-th MOS value, σ is the standard deviation of scores and ε_i^w is the i-th residual value. The initial estimates for the parameters in Eq. 3.19 are $\beta_1 = \max(Y_i)$, $\beta_2 = \min(Y_i)$, $\beta_3 = \tilde{X}$, $\beta_4 = 1$, $w_i = \sigma^{-1}$, and $Y_i^w = w_i \cdot Y_i^w$.

Considering that degradation to a PC can occur in both color and geometry components, the best performance by the metric is obtained when a color-based texture descriptor is used together with a geometry-based texture descriptor. One possible setup of the proposed full-reference PCQA method can be summarized in Figure 3.10, in which the LCP and the geometry-based texture descriptor are used together to estimate the quality of a PC. The output distances of the LCP and the geometry-based descriptors are simply averaged in the example.

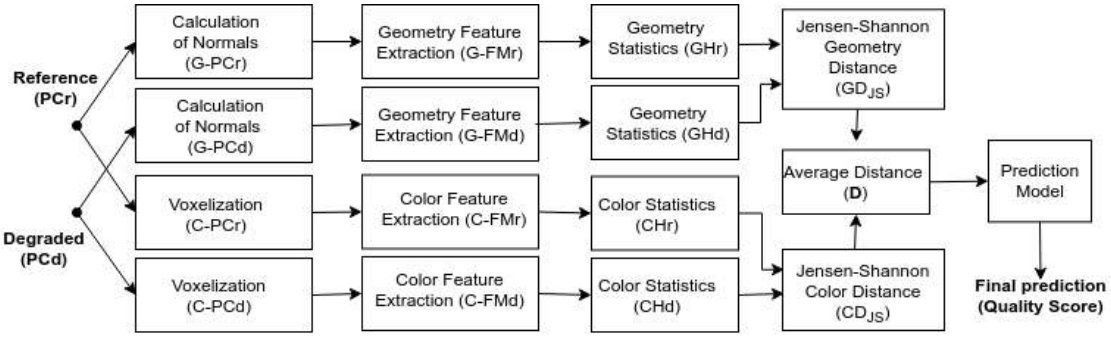


Figure 3.10: Diagram of the proposed PC quality assessment metric framework.

In the Figure 3.10, the color-based descriptor path is outlined in the bottom part of the figure, which shows first the voxelization procedure (C-PC$_{r,d}$), the color-based feature map extraction (C-FM$_{r,d}$) is the next step, and then the color-based feature histogram creation (CH$_{r,d}$). The geometry path is in the top of the figure, in which the first step is the normals calculation (G-PC$_{r,d}$), followed by the geometry-based feature extraction (G-FM$_{r,d}$), then the geometry feature histogram is created (GH$_{r,d}$). For both color and geometry paths, the histograms distance of the reference and degraded PCs (referenced by small r and d, respectively) are calculated. In the example, the Jensen-Shannon distance is used, next, both distances (GD and CD) are averaged (**D**) and used in the prediction model, which applies a regression method to provide the final quality prediction score.

Chapter 4

Results And Comparison To State-Of-The-Art Metrics

This chapter presents the experimental setup of the proposed quality assessment method, its simulation results, the performance comparison between the proposed quality assessment method to other state-of-the-art PCQA metrics, and finally some conclusions.

4.1 Experimental Setup

The implementation of the texture descriptors was done in a multi-threaded C and C++ code. The implemented code uses the Open3D[94] library, which provides the structures for point cloud memory storage and an optimized nearest neighbor search algorithm. The statistical analysis and regressions code was developed using the Scikit-Learn library [95]. The proposals were tested in both a high-end computer and a notebook computer. The high-end computer is a dual eight-core Intel Xeon E5-2620, with 80GB of RAM memory, while the notebook setup is Lenovo ThinkPad T430 with a quad-core Intel Core i7-3632QM with 16GB of RAM. While the runtime complexity of the metric was not evaluated, by comparing the implementation done for this thesis and the other metrics implementations tested, the proposed PCQA metric is faster than the others.

The proposed PC quality assessment methods presented in this thesis are tested with a variety of PC datasets and compared to other state-of-the-art PC quality assessment metrics, which were presented Chapter 2. The selected datasets and associated subjective scores represent the most up-to-date and diverse datasets available in the literature [56, 64, 66, 67], presented with more detail in Section 2.2.1. Also in Section 2.2.1, it was named D1 the dataset by Torlig et al. [56], D2 the dataset by Alexiou et al. [64], D3 the dataset by Stuart el at. [66], and D4 the dataset by Yang et al. [67]. Important to note

that D2 and D3 contain only MPEG codec compression distortions, while D1 and D4 have a more diverse set of distortions.

In order to justify the choices for the regression model and for the distance calculation of the texture descriptor histograms, in this section it is shown a regression model analysis, a distance metrics analysis, the texture descriptors analysis, and, then, the comparison with other state-of-the-art metrics. Figure 4.1 shows the block diagram containing the histogram distance calculation and the regression model used in the proposed quality metric workflow, considering the case of a single descriptor being used for quality estimation.

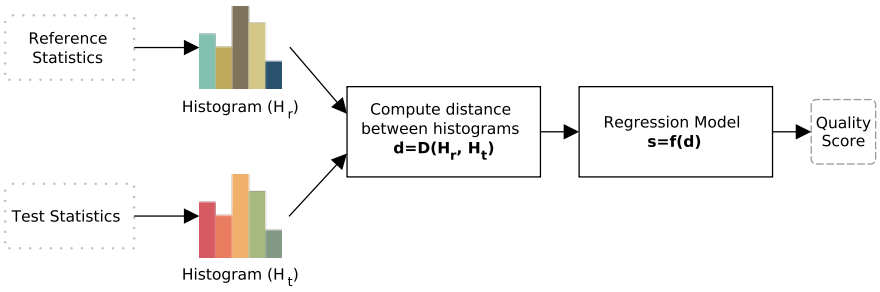


Figure 4.1: Block diagram of the quality assessment workflow illustrating the histogram distance calculation and the regression model in the quality assessment workflow..

In this chapter all the evaluations that compare the predicted quality scores with the subjective scores provided in the benchmark databases use all or some of the following correlation metrics: the Spearman rank-order correlation coefficient (SROCC), the Pearson linear correlation coefficient (PCC), and the the root mean square error (RMSE), shown below:

$$PCC(m_i, p_i) = \frac{\sum_i (m_i - m_a)(p_i - p_a)}{\sqrt{\sum_i (m_i - m_a)^2} \sqrt{\sum_i (p_i - p_a)^2}}, \tag{4.1}$$

where m_i is the subjective MOS score, p_i is the predicted score, and m_a and p_a are their average.

$$SROCC(m_i, p_i) = 1 - \frac{6 \sum_{i=1}^{L} (m_i - r_i)^2}{L(L^2 - 1)}, \tag{4.2}$$

where m_i is the subjective MOS score, p_i is the predicted score, r_i is the rank order of p_i and L is the number of test content PCs.

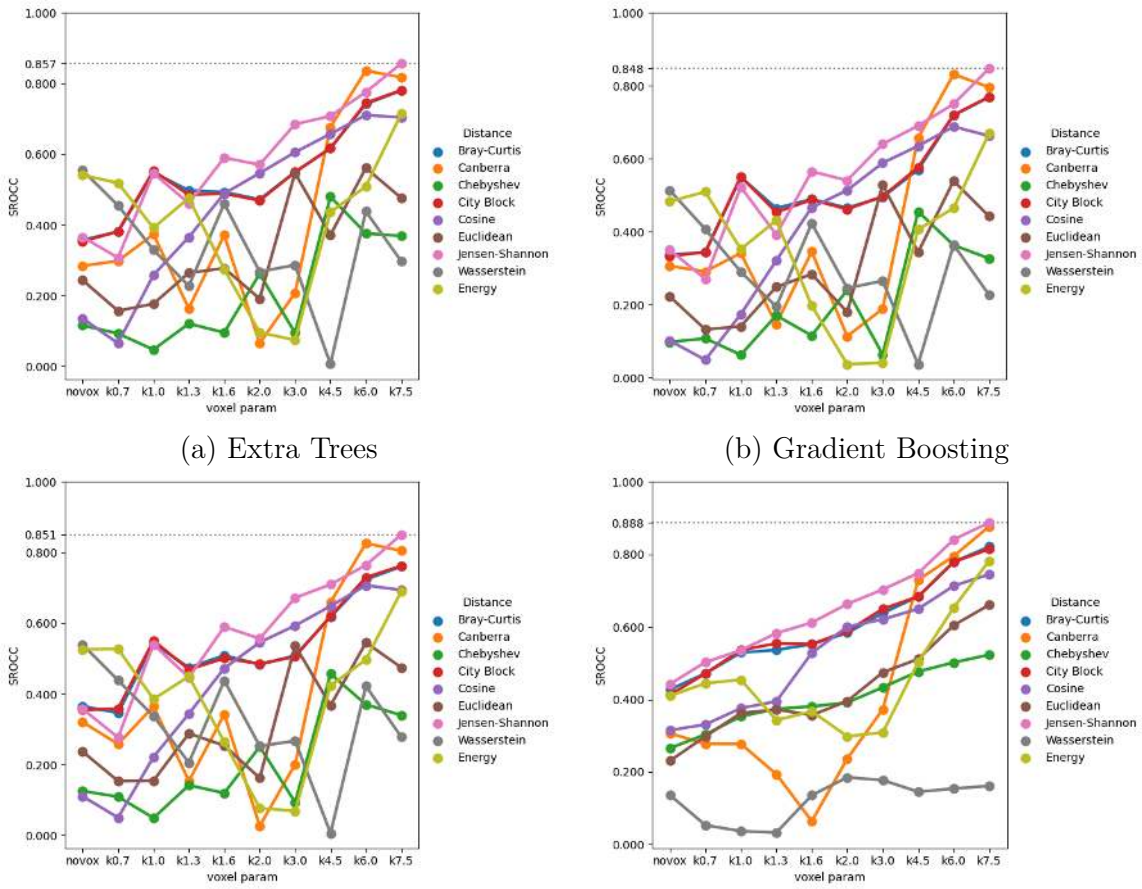
$$RMSE(m_i, p_i) = \sqrt{\frac{\sum_{i=1}^{L}(m_i - p_i)}{L}}, \tag{4.3}$$

where the variables have the same meaning as in equation 4.2 and 4.1.

The use of these measures are in accordance to ITU recommendations for visual quality assessment [93]. Greater PCC and SROCC is better, meaning higher correlation to the subjective ground truth, while the RMSE represents the residual error, and so, the less the better.

With regards to the regression model, Extra Trees, Gradient Boosting, Random Forest and the Logistic function regressors were evaluated. The Extra Trees regressor is a meta estimator that fits a number of randomized decision trees (extra-trees) on various sub-samples of the dataset. It uses averaging to improve the predictive accuracy and control over-fitting. The Random Forrest Regressor is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset. It also uses averaging to improve the predictive accuracy and control over-fitting. Finally the Gradient Boosting regressor builds an additive model in a forward stage-wise manner, optimizing arbitrary differentiable loss functions. In the case of this work, the loss function tested was the Huber loss. In each stage a regression tree is fit on the negative gradient of the given loss function [96]. The Extra Trees, Gradient Boosting, Random Forest regressors were used as implemented in Scikit-learn software [95]. Additionally to these regressors, the logistic function as described by ITU [93] was also used as regressor, and its equation is shown in equation 3.19.

The evaluation of the regressor models are present in Figures 4.2, 4.3 and 4.4, in which the SROCC, PCC and RMSE values are shown for dataset D1. The LCP descriptor (Section 3.1.4) was adopted for the regressors evaluation. The LCP with 8-bit label and 12 neighbors was used, while different histogram distances were applied (see Section 3.2). The k parameter of the voxelization (equation 3.1) was tested with values of 0.7, 1.0, 1.3, 1.6, 2.0, 3.0, 4.5, 6.0, 7.5 and the case without the application of the voxelization ("novox" in the figures).



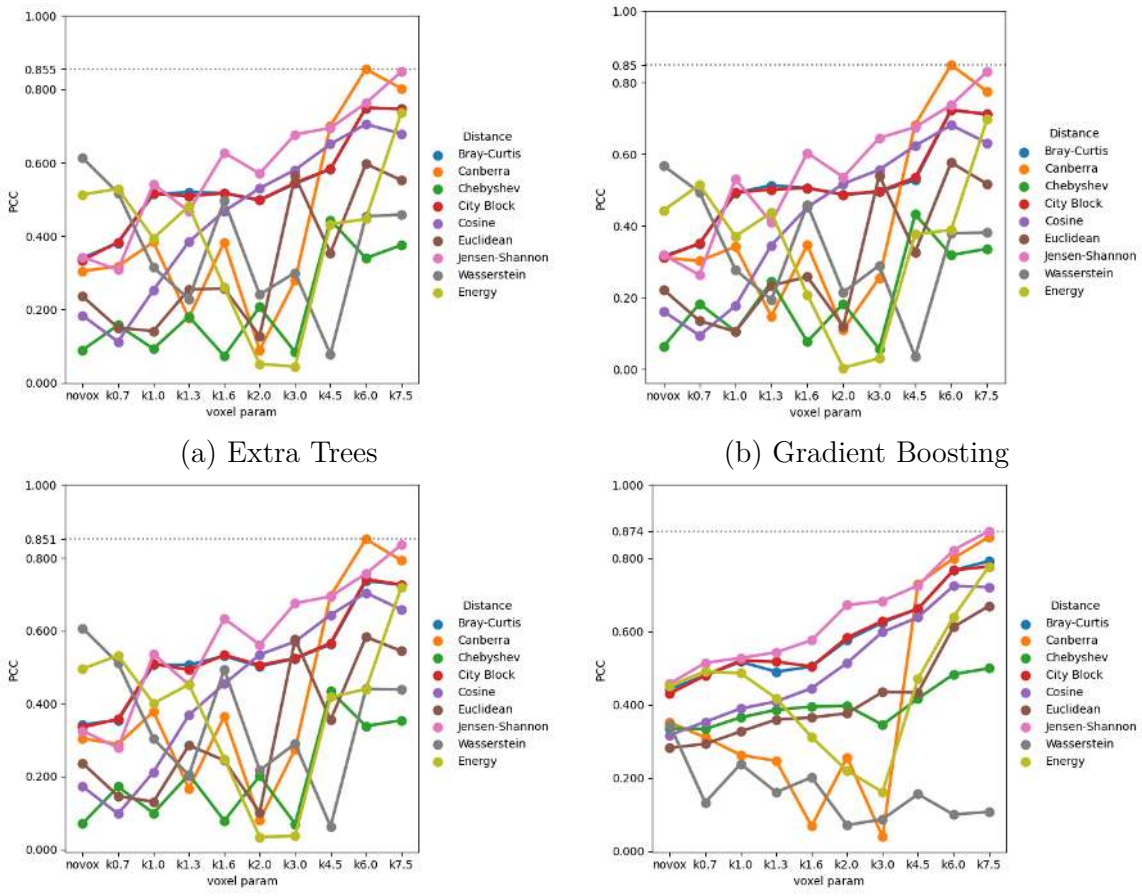
(a) Extra Trees

(b) Gradient Boosting

(c) Random Forest

(d) Logistic

Figure 4.2: Evaluation of the SROCC correlation of the Extra Trees, Gradient Boosting, Random Forest regressors, with different histogram distances and varying voxelization k parameter.



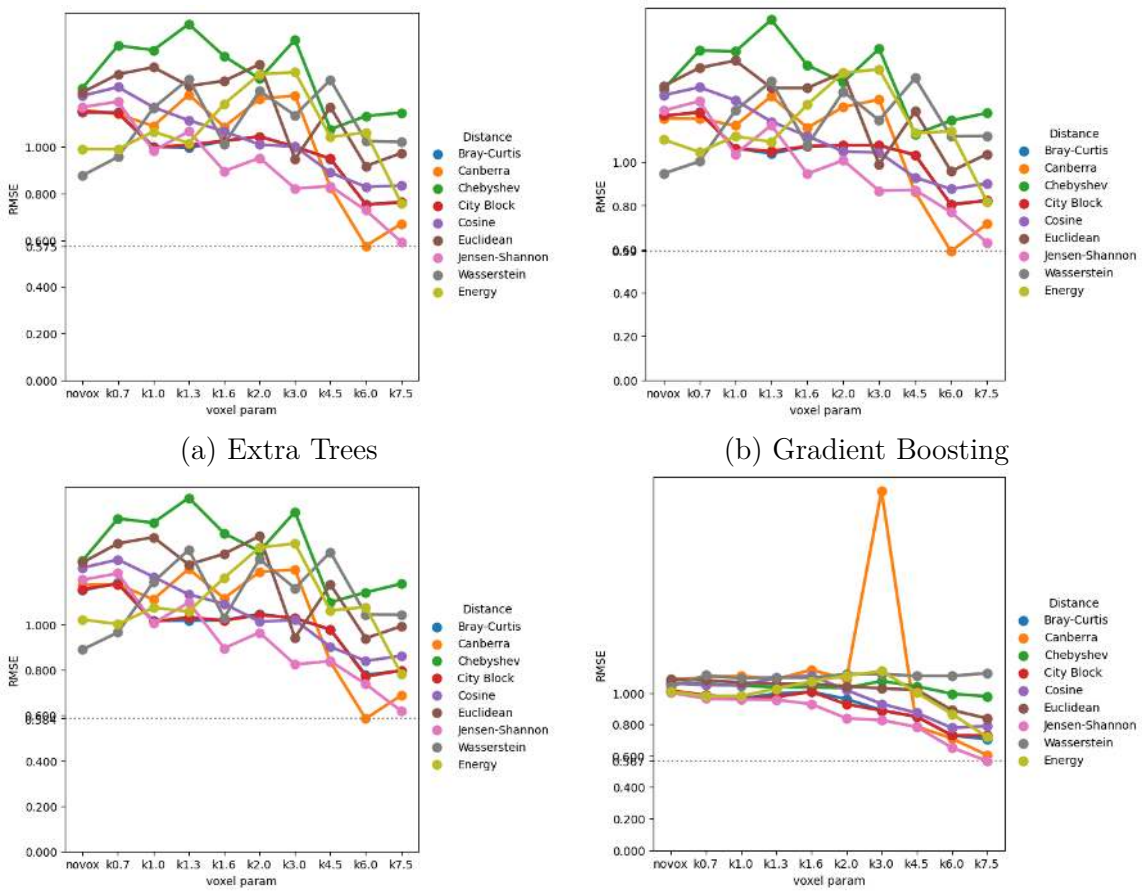
(a) Extra Trees

(b) Gradient Boosting

(c) Random Forest

(d) Logistic

Figure 4.3: Evaluation of the PCC correlation of the Extra Trees, Gradient Boosting, Random Forest regressors, with different histogram distances and varying voxelization k parameter.



(a) Extra Trees

(b) Gradient Boosting

(c) Random Forest

(d) Logistic

Figure 4.4: Evaluation of the RMSE of the Extra Trees, Gradient Boosting, Random Forest regressors, with different histogram distances and varying voxelization k parameter.

While the present analysis on the available state-of-the-art regressors applied to this PC quality assessment proposal is not extensive, the comparison uses high performance regressors, and provide the basis for the adoption of the Logistic regressor throughout in this work. The Logistic regression performs best, and in terms of PCC (0.874), SROCC (0.888), and RMSE (0.567). It is worth mentioning that the Logistic regression is already recommended as a regression method for 2D image quality assessment [93]. This analysis just confirms that the Logistic regression is also good for 3D PCs and this is the reason why it is adopted it in this work.

4.2 Simulation Results

In this section, the results of simulations are presented along with an analysis of the proposed PC texture descriptors and the associated distance metrics used to calculate

the distance between the texture descriptor histograms.

Figures 4.5 to 4.28 show the analysis of the following PC texture descriptors proposed in this work: LBP, LLP with 16 bits and 12 bits labels, LCP with 12 bits and 8 bits labels, and the geometry-based texture descriptor with 16 bits label. The results were obtained through the application of the texture descriptors to all 4 datasets listed before, with neighborhood size of 6, 8, 10 and 12 (one neighborhood size per line, respectively), and voxelization k parameter with values of 0.7, 1.0, 1.3, 1.6, 2.0, 3.0, 4.5, 6.0, 7.5, and "novox" (no voxelization). The distance calculation between reference and test histograms obtained by the application of the texture descriptors evaluated were: Bray-Curtis, Canberra [89], Cityblock [90], Chebyshev, Cosine, Euclidean, Jensen-Shannon [91] Wasserstein [92], and Energy, as implemented in the Scipy library [97]. Important to note that the regression applied to one dataset independently to other datasets, what means the histogram distances are fit to the subjective data of the dataset in evaluation. While fitting the histogram distances of one dataset to the subjective ground truth of the same dataset is an over-fitting procedure, this is justifiable as there are still very few PC datasets with associated subjective data, and as so, this procedure is accepted in the PCQA methods up to date, while most the literature up to now compare the state-of-the-art metrics using such over-fitting method.

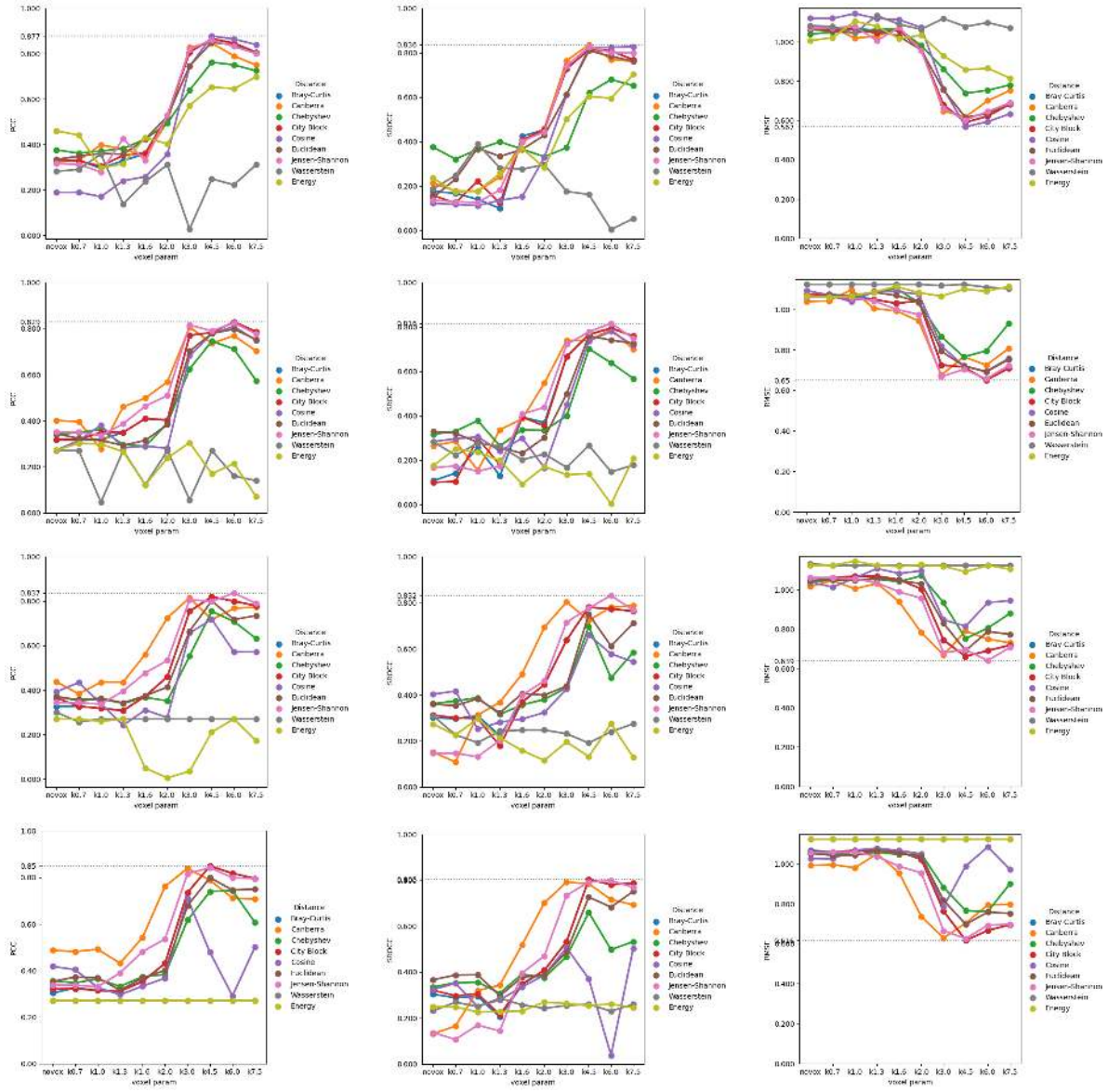


Figure 4.5: D1 LBP descriptor performance with different histogram distances evaluated, with each row representing the LBP with 6, 8, 10 and 12 neighbors, respectively.

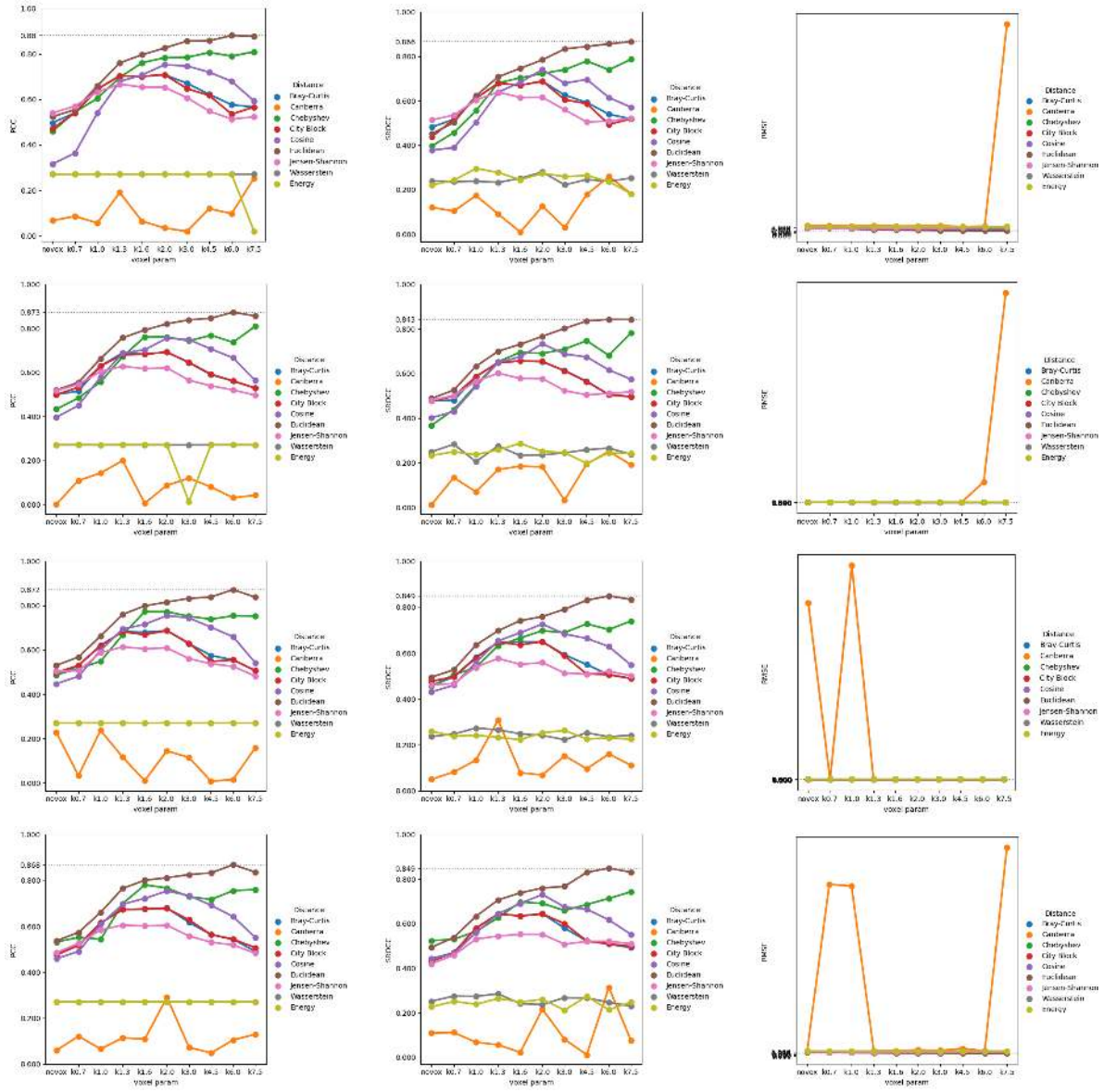


Figure 4.6: D1 LLP 16 bits descriptor performance with different histogram distances evaluated, with each row representing the LLP with 6, 8, 10 and 12 neighbors, respectively.

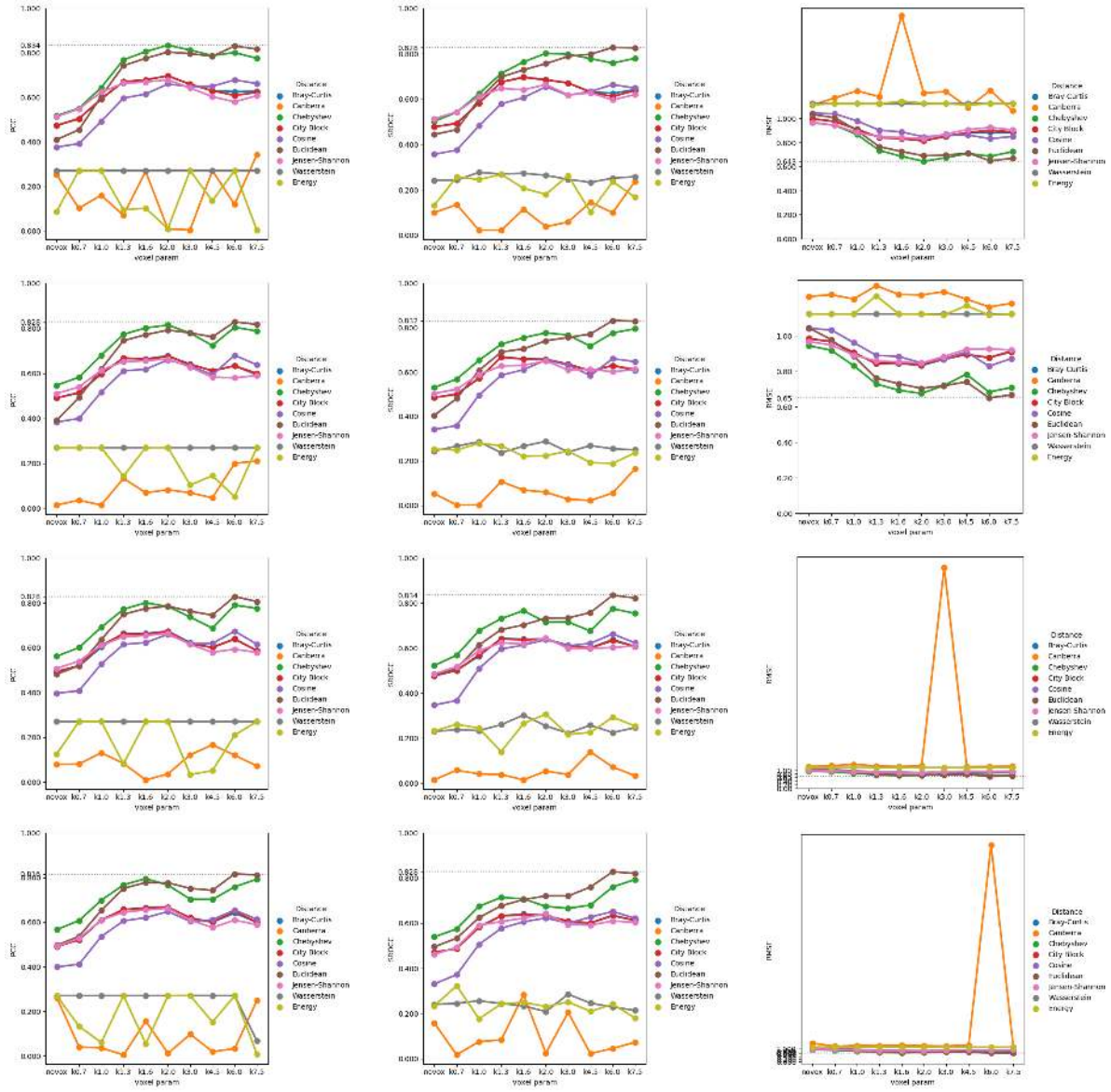


Figure 4.7: D1 LLP 12 bits descriptor performance with different histogram distances evaluated, with each row representing the LLP with 6, 8, 10 and 12 neighbors, respectively.

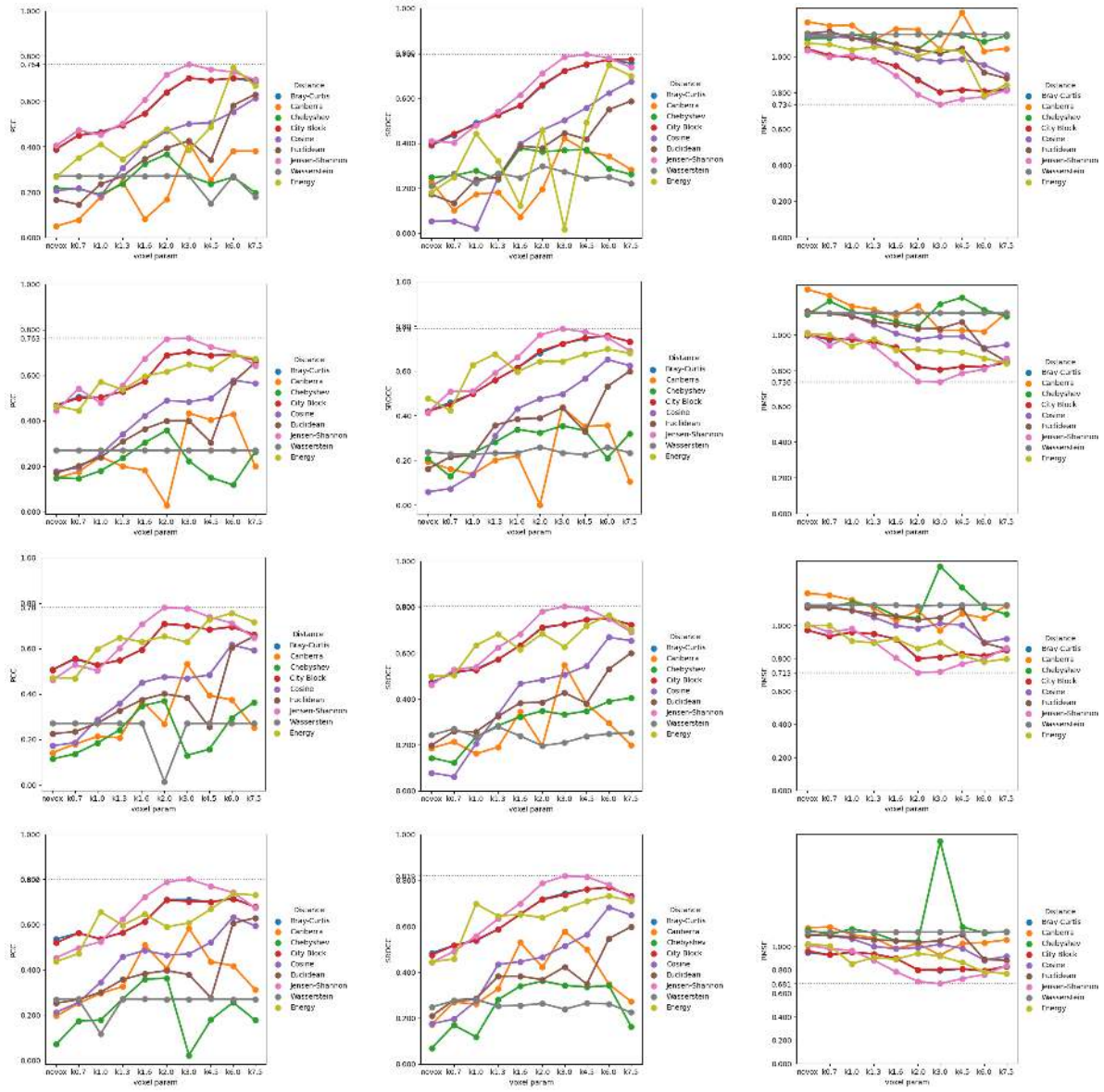


Figure 4.8: D1 LCP 12 bits descriptor performance with different histogram distances evaluated, with each row representing the LCP with 6, 8, 10 and 12 neighbors, respectively.

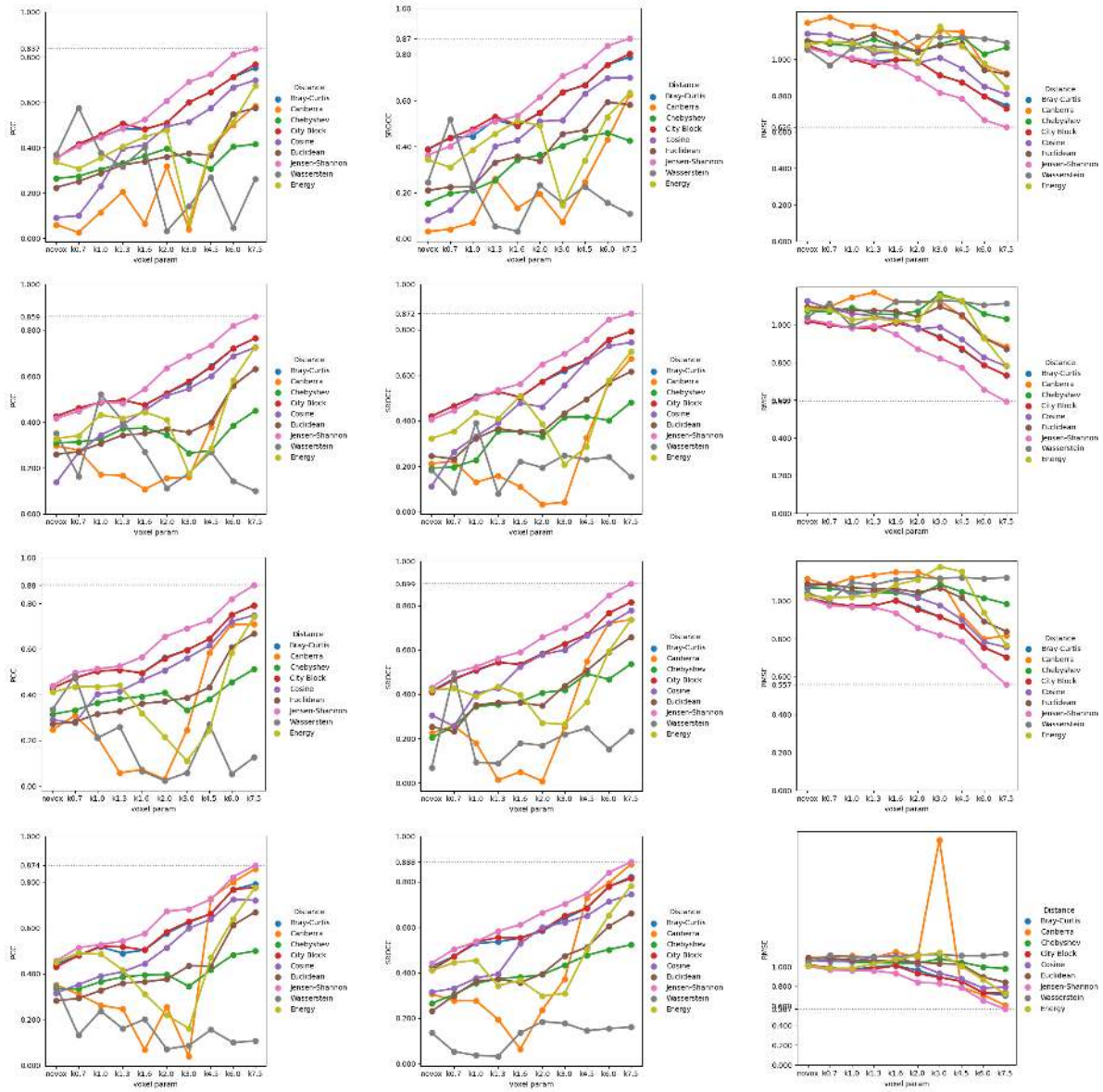


Figure 4.9: D1 LCP 8 bits descriptor performance with different histogram distances evaluated, with each row representing the LCP with 6, 8, 10 and 12 neighbors, respectively.

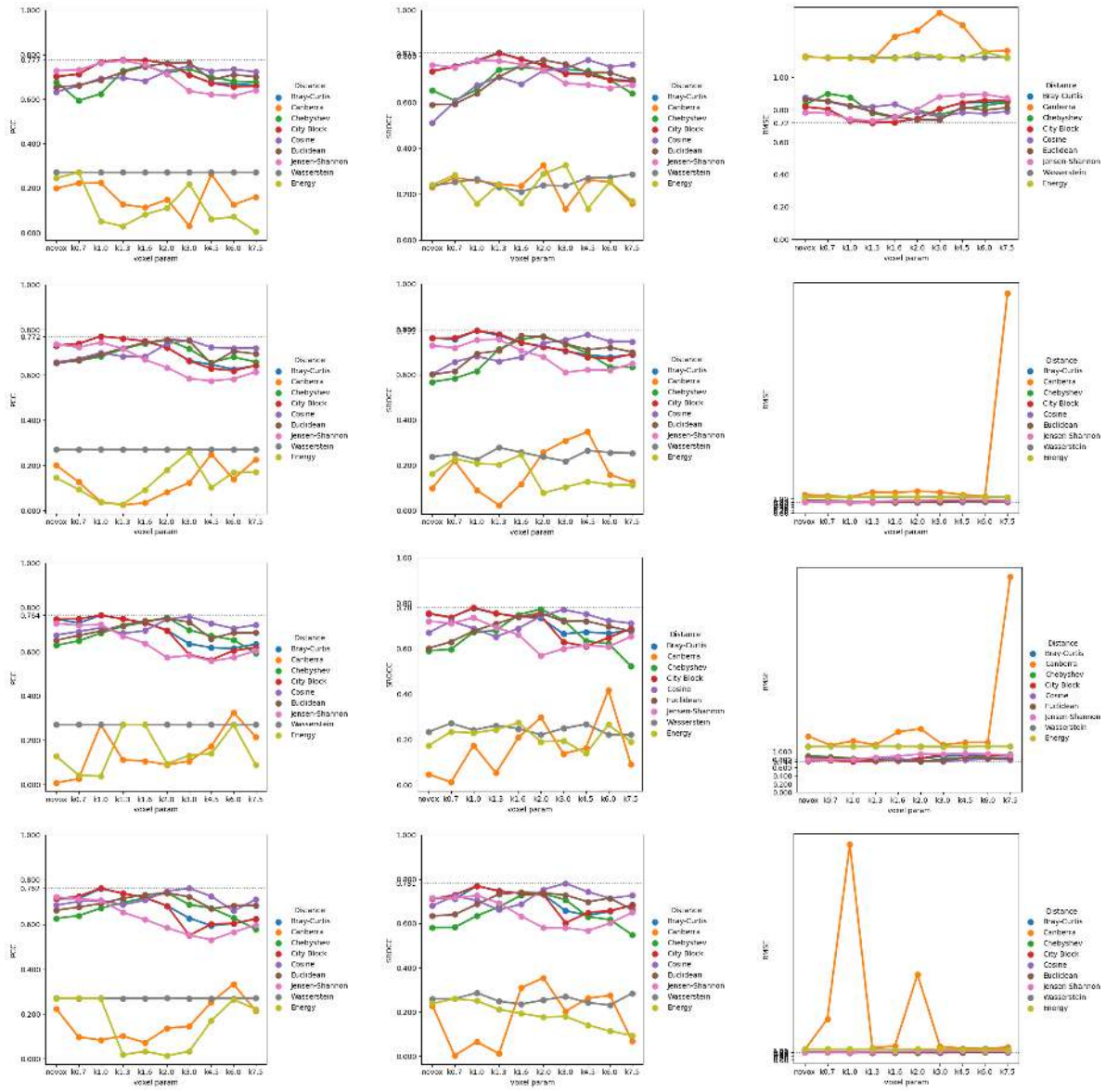


Figure 4.10: D1 geometry-based descriptor performance with different histogram distances evaluated, with each row representing the geometry-based descriptor with 6, 8, 10 and 12 neighbors, respectively.

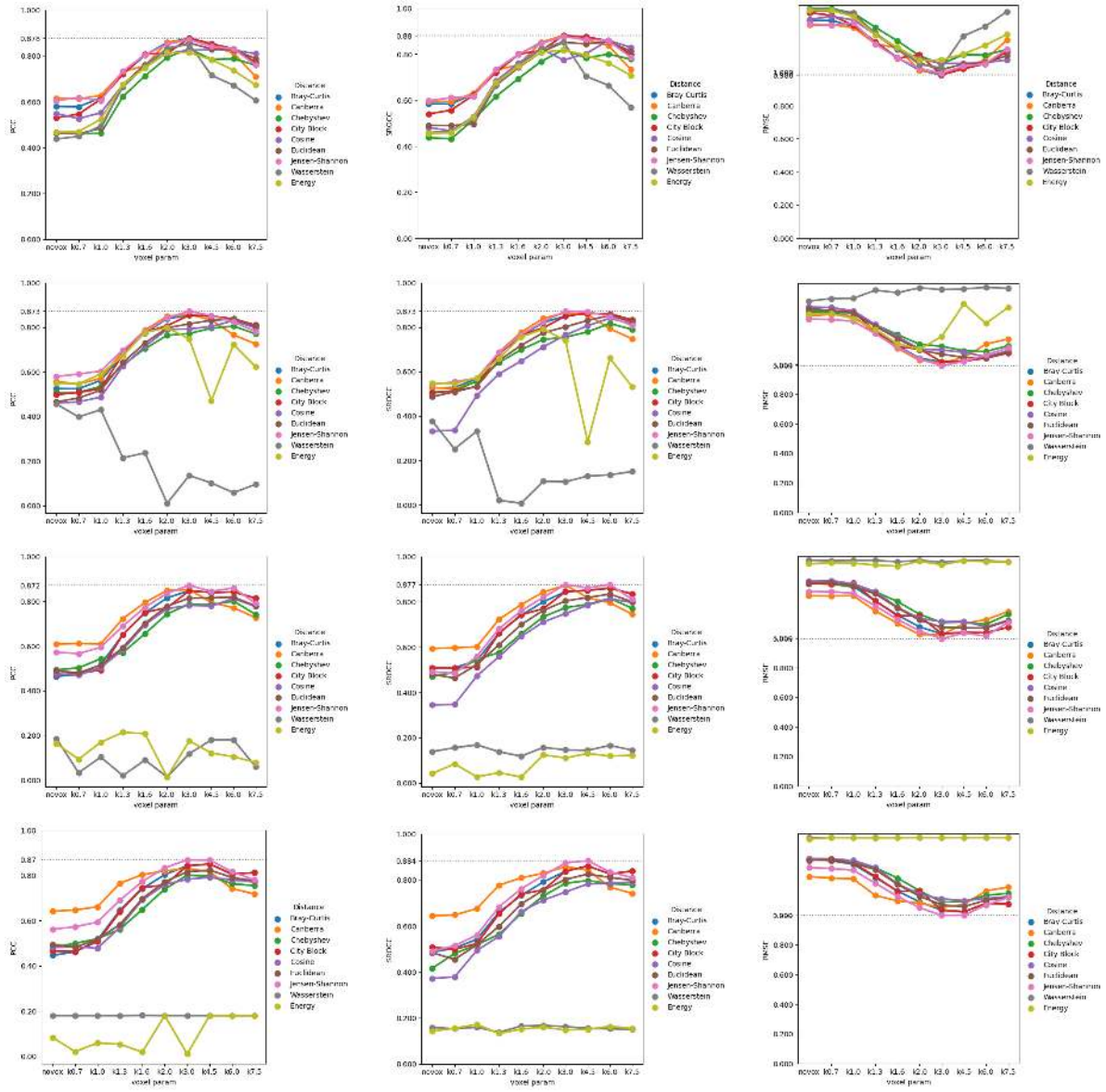


Figure 4.11: D2 LBP descriptor performance with different histogram distances evaluated, with each row representing the LBP with 6, 8, 10 and 12 neighbors, respectively.

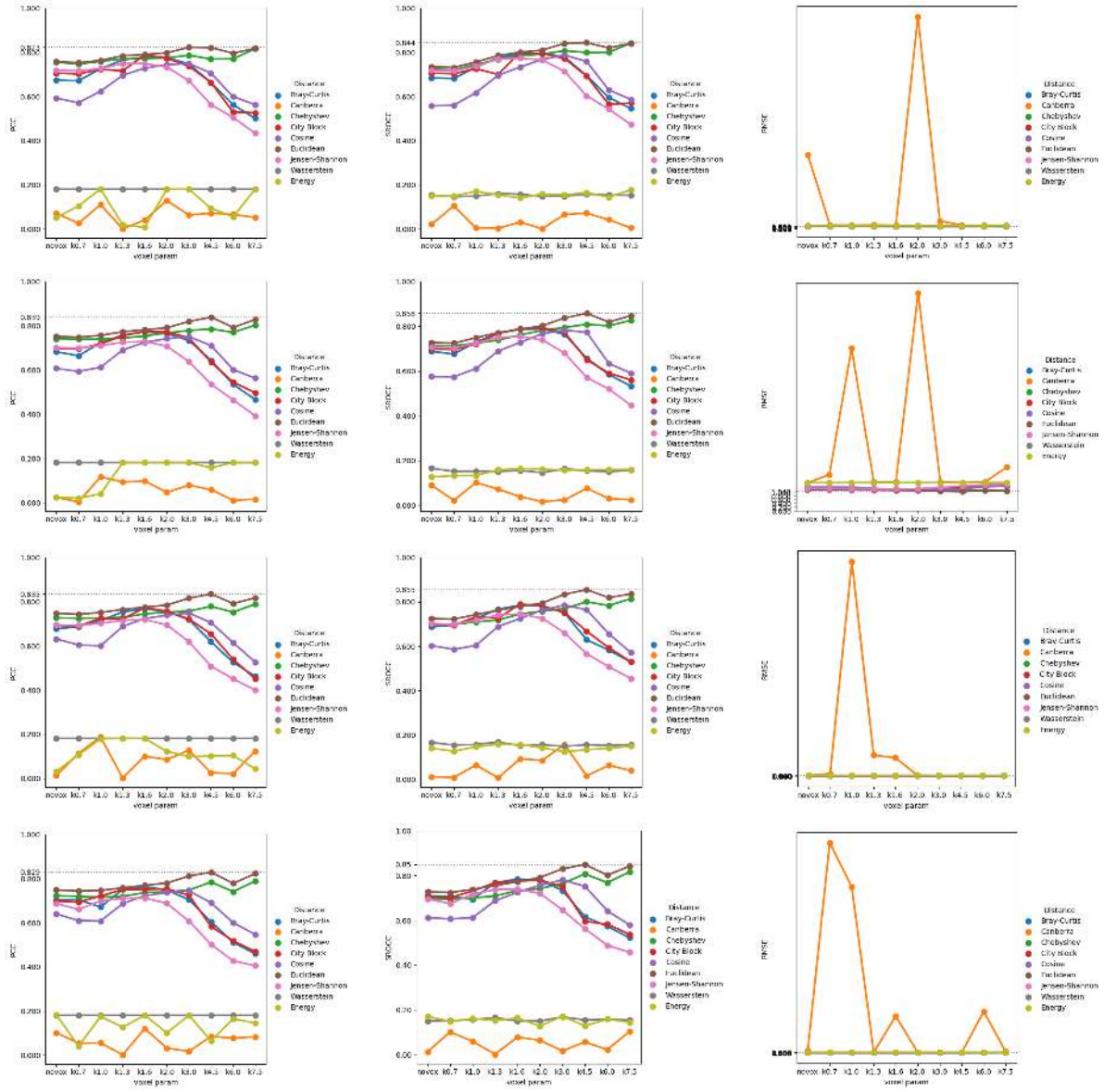


Figure 4.12: D2 LLP 16 bits descriptor performance with different histogram distances evaluated, with each row representing the LLP with 6, 8, 10 and 12 neighbors, respectively.

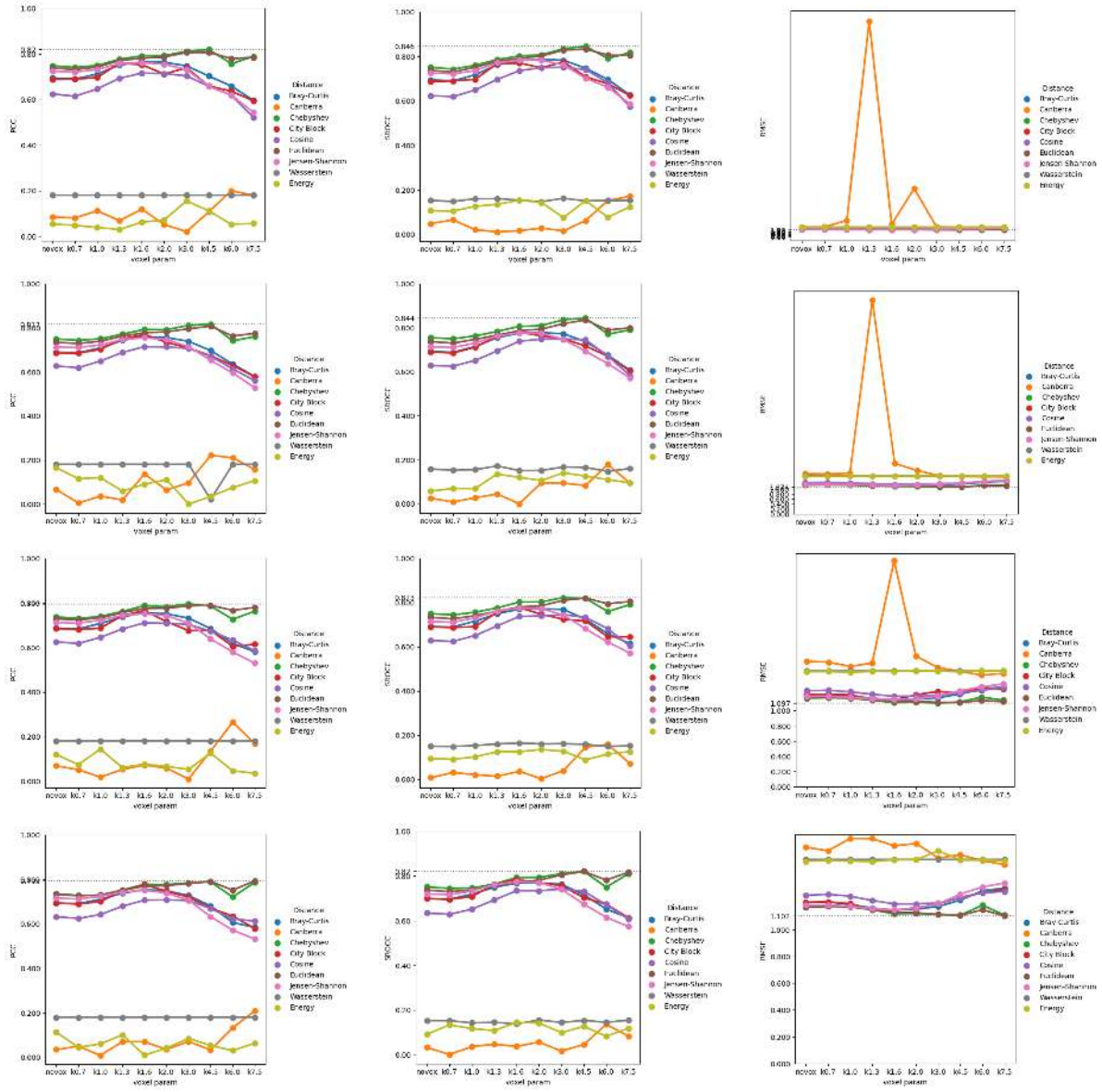


Figure 4.13: D2 LLP 12 bits descriptor performance with different histogram distances evaluated, with each row representing the LLP with 6, 8, 10 and 12 neighbors, respectively.

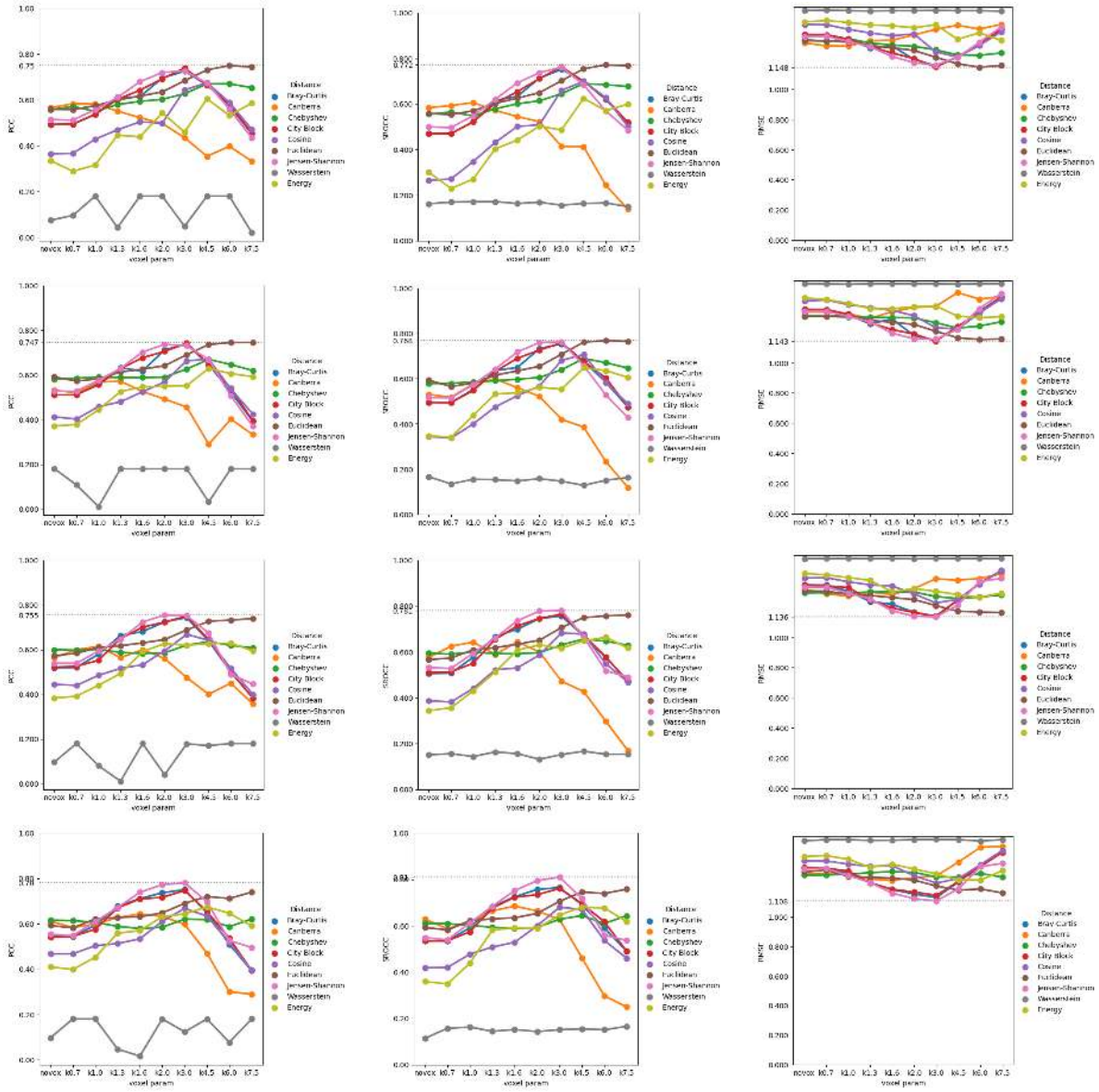


Figure 4.14: D2 LCP 12 bits descriptor performance with different histogram distances evaluated, with each row representing the LCP with 6, 8, 10 and 12 neighbors, respectively.

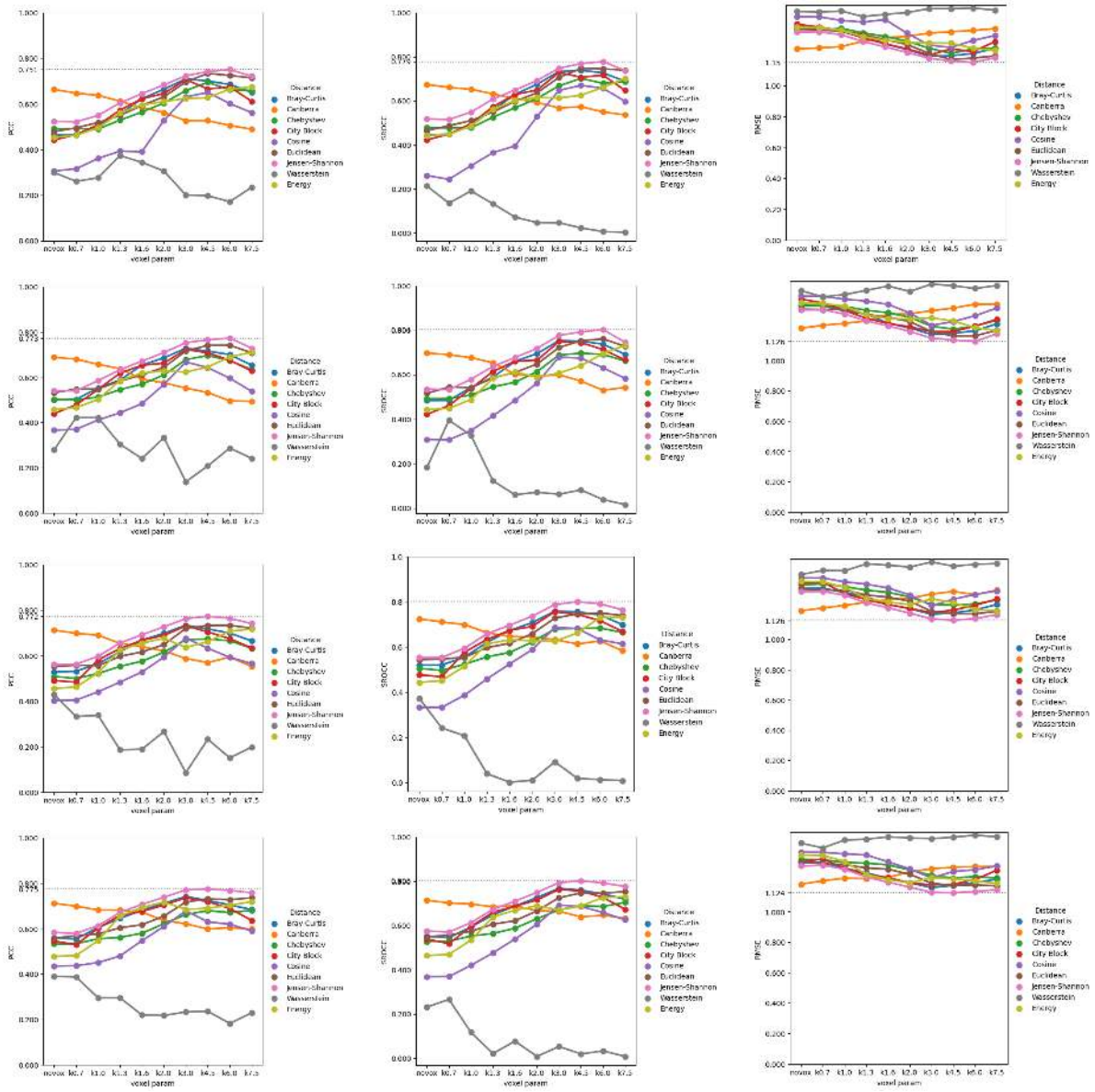


Figure 4.15: D2 LCP 8 bits descriptor performance with different histogram distances evaluated, with each row representing the LCP with 6, 8, 10 and 12 neighbors, respectively.

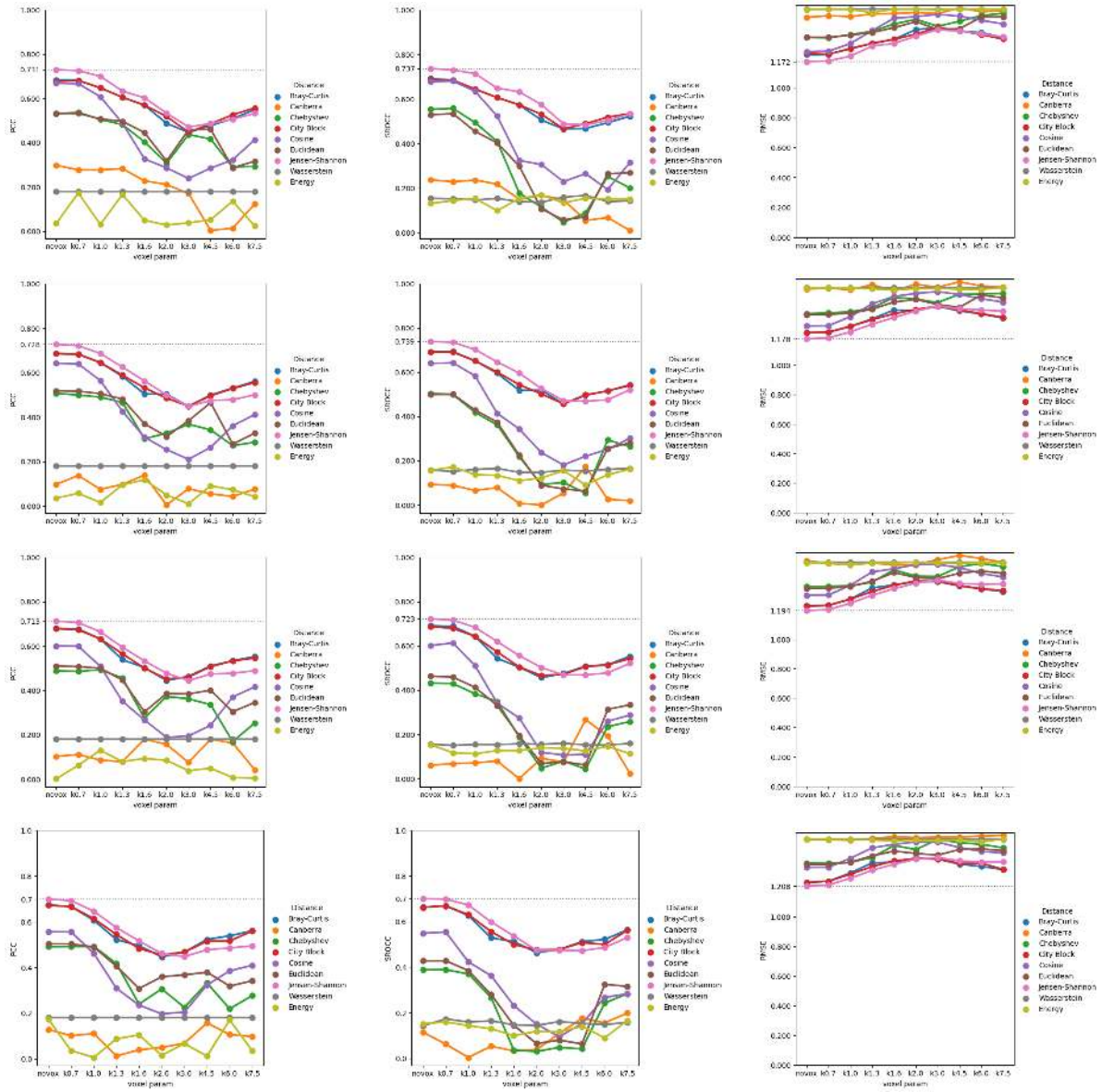


Figure 4.16: D2 geometry-based descriptor performance with different histogram distances evaluated, with each row representing the geometry-based descriptor with 6, 8, 10 and 12 neighbors, respectively.

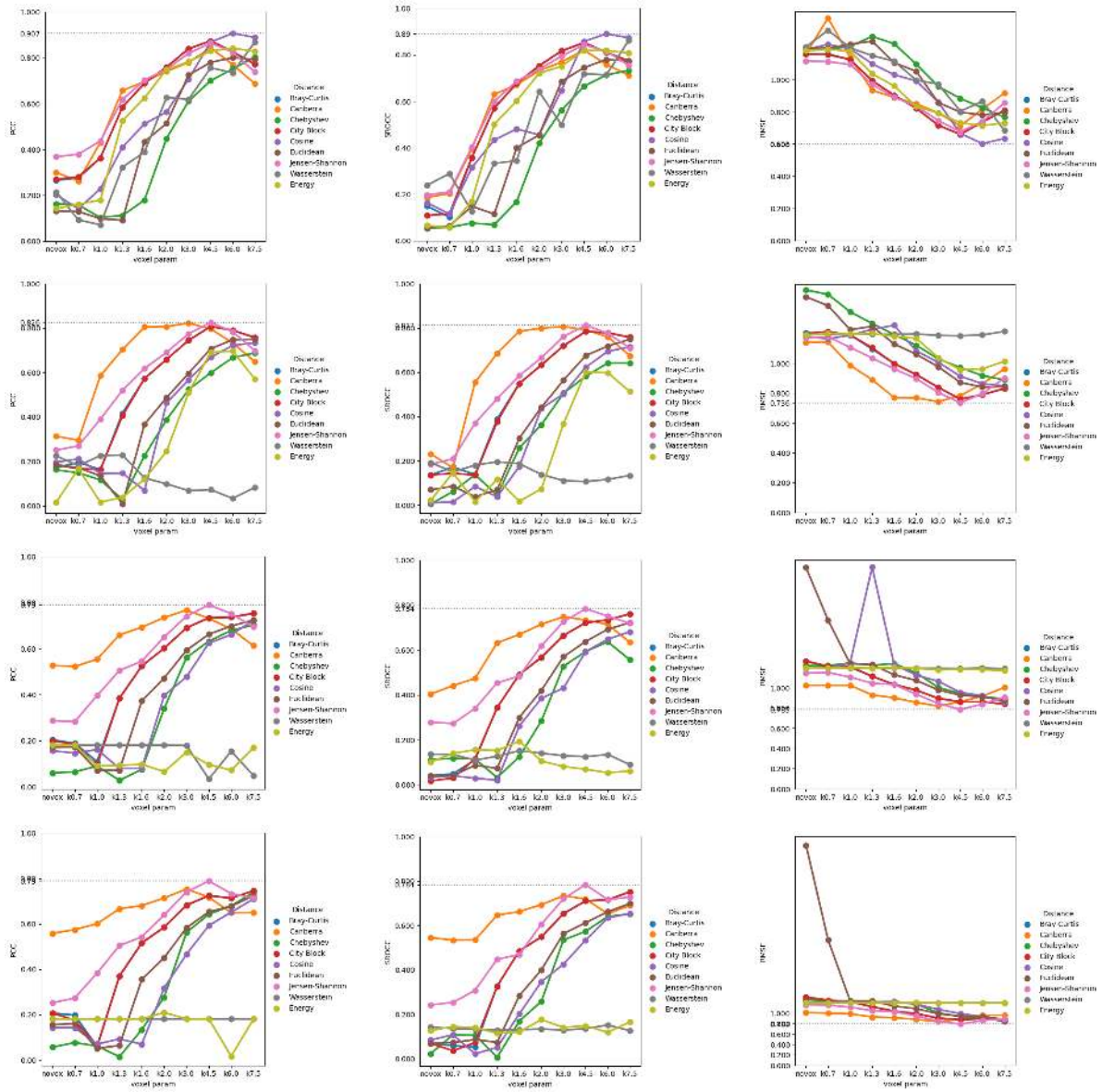


Figure 4.17: D3 LBP descriptor performance with different histogram distances evaluated, with each row representing the LBP with 6, 8, 10 and 12 neighbors, respectively.

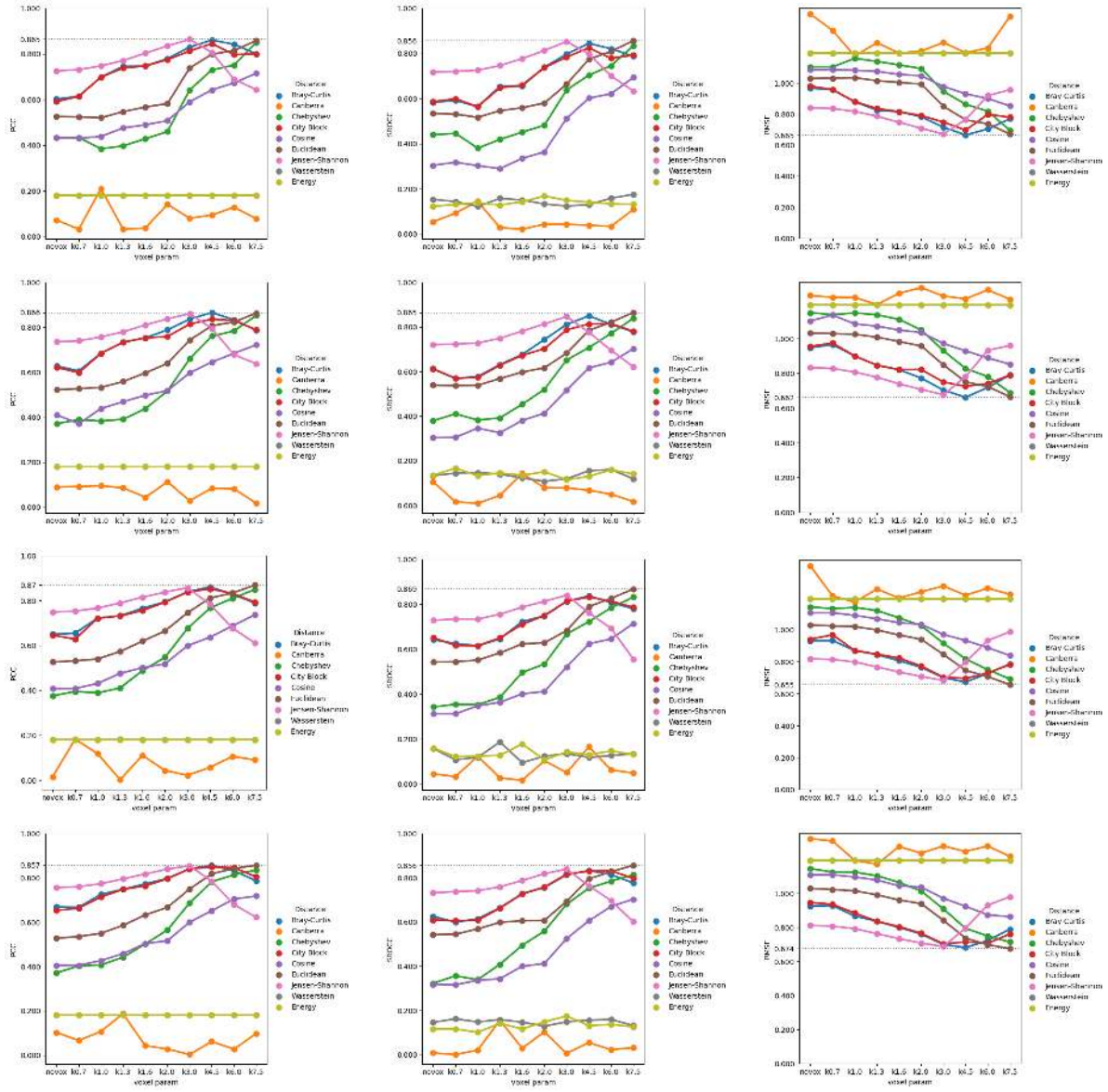


Figure 4.18: D3 LLP 16 bits descriptor performance with different histogram distances evaluated, with each row representing the LLP with 6, 8, 10 and 12 neighbors, respectively.

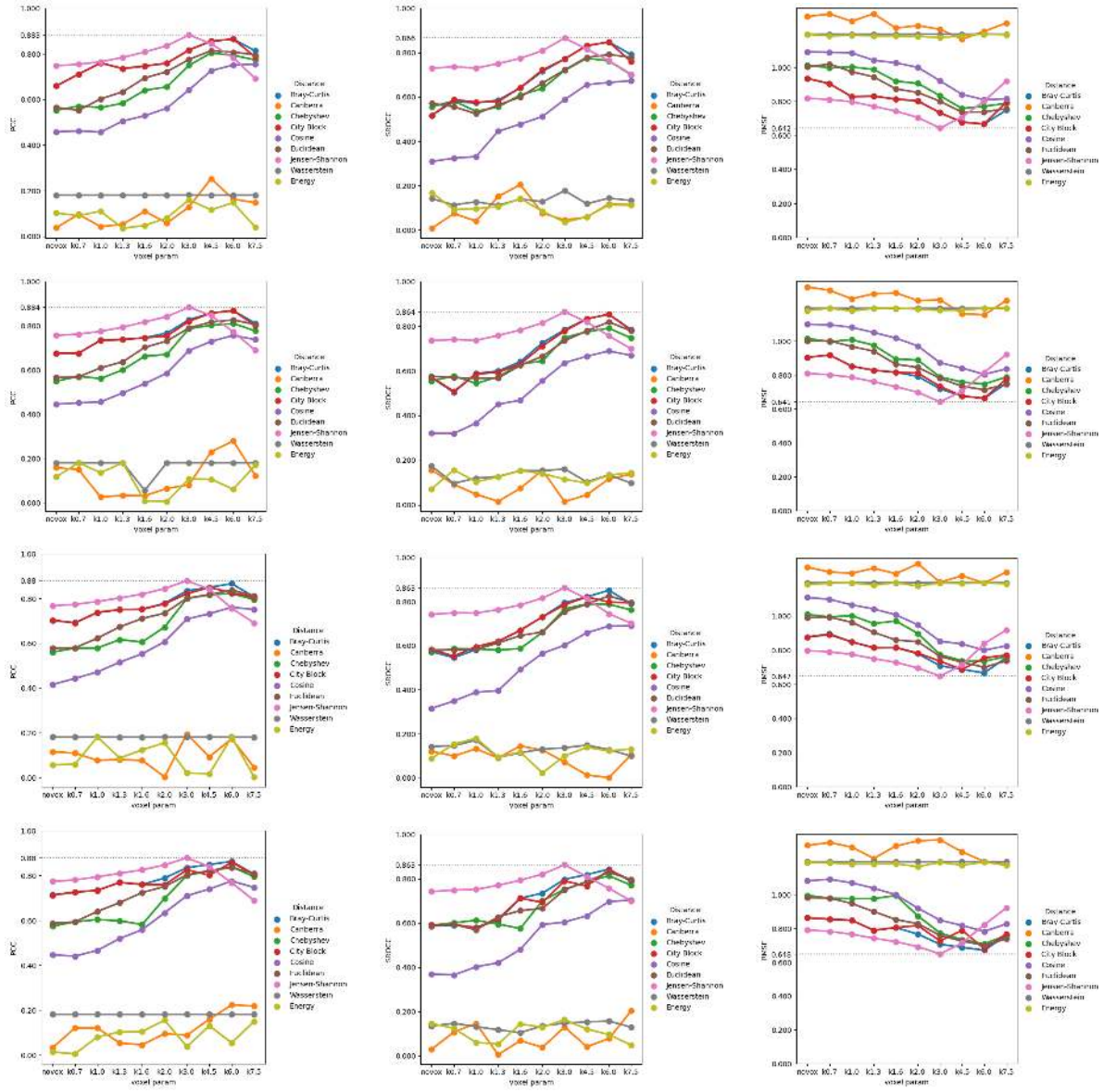


Figure 4.19: D3 LLP 12 bits descriptor performance with different histogram distances evaluated, with each row representing the LLP with 6, 8, 10 and 12 neighbors, respectively.

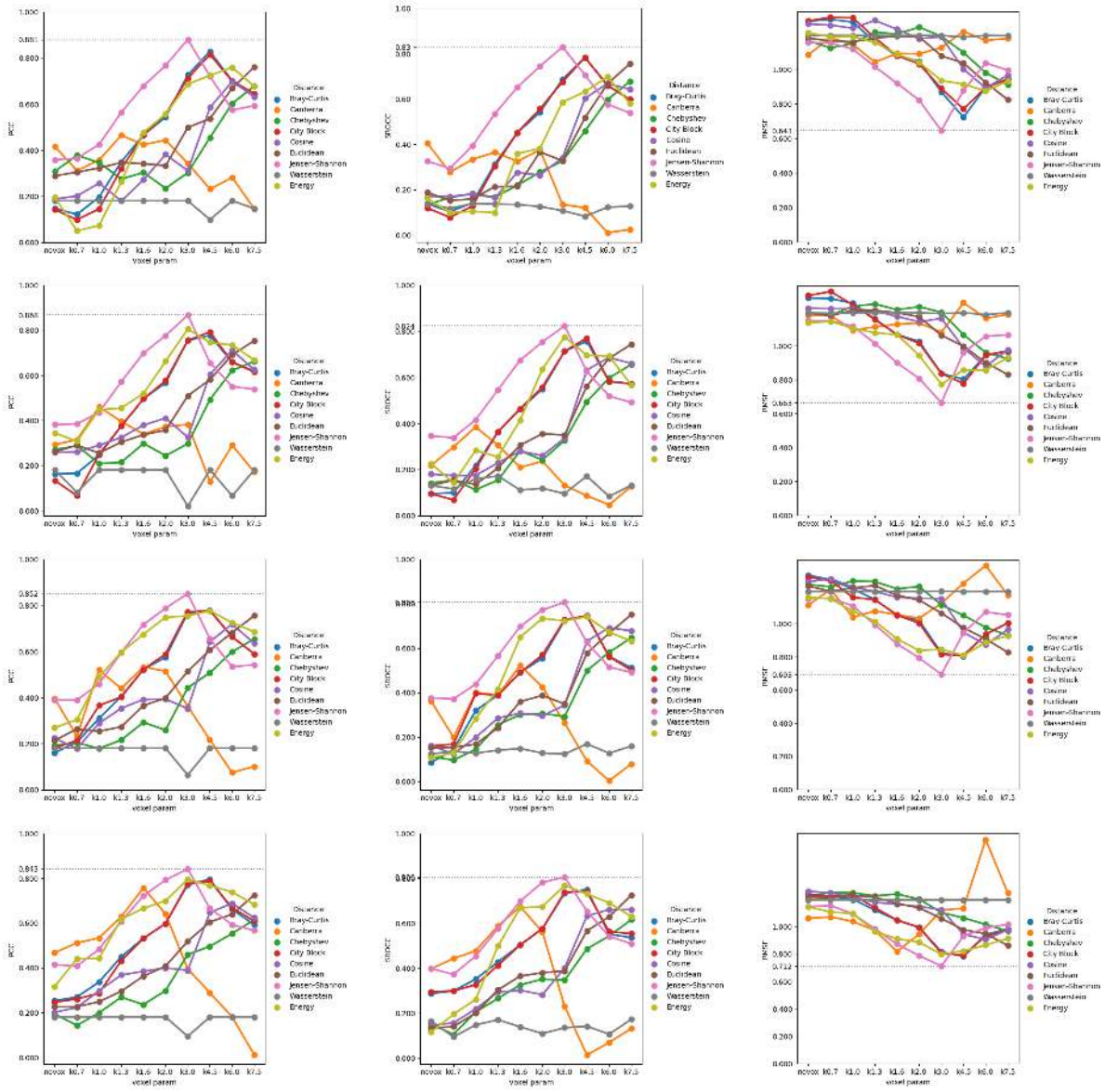


Figure 4.20: D3 LCP 12 bits descriptor performance with different histogram distances evaluated, with each row representing the LCP with 6, 8, 10 and 12 neighbors, respectively.

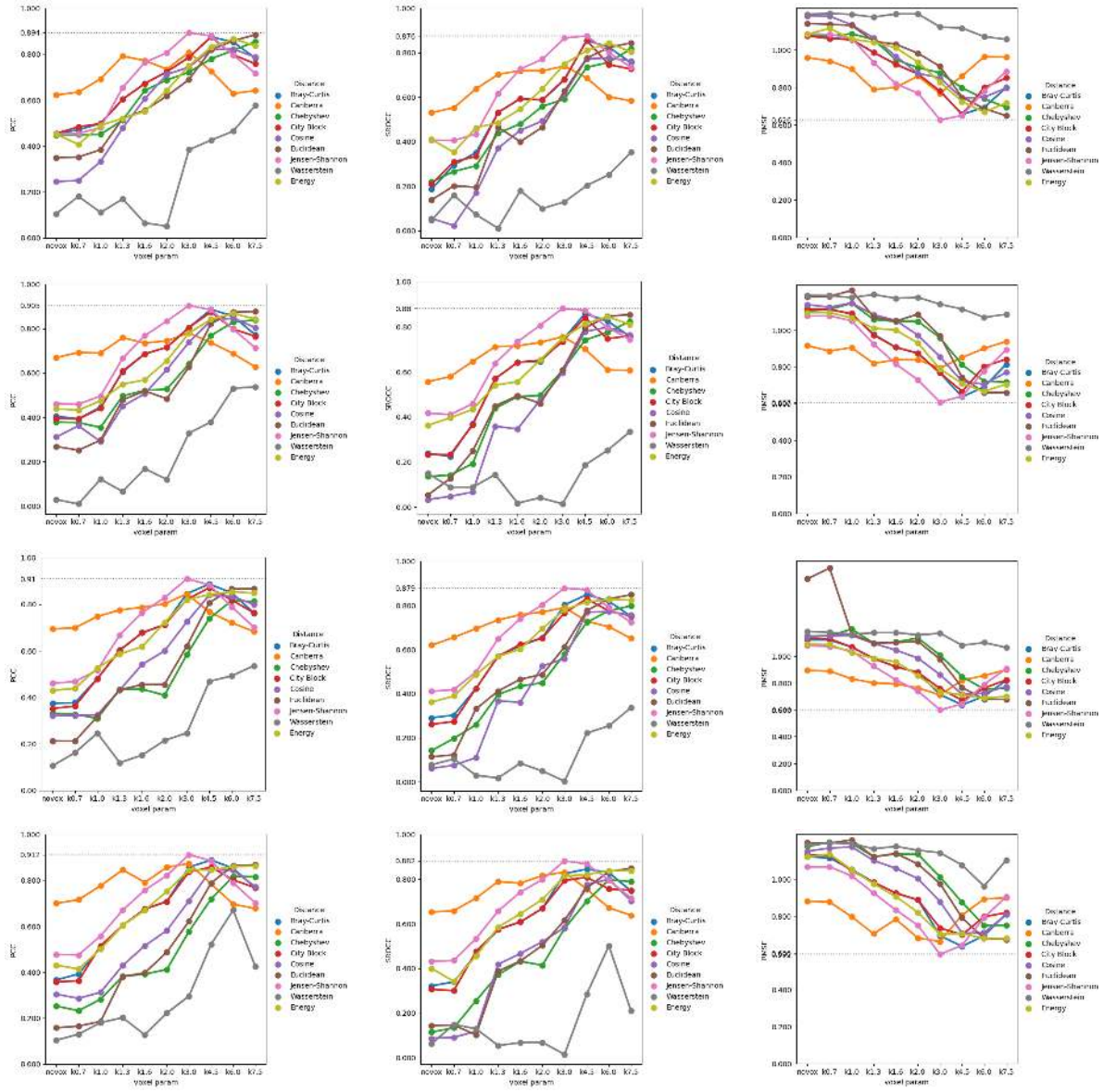


Figure 4.21: D3 LCP 8 bits descriptor performance with different histogram distances evaluated, with each row representing the LCP with 6, 8, 10 and 12 neighbors, respectively.

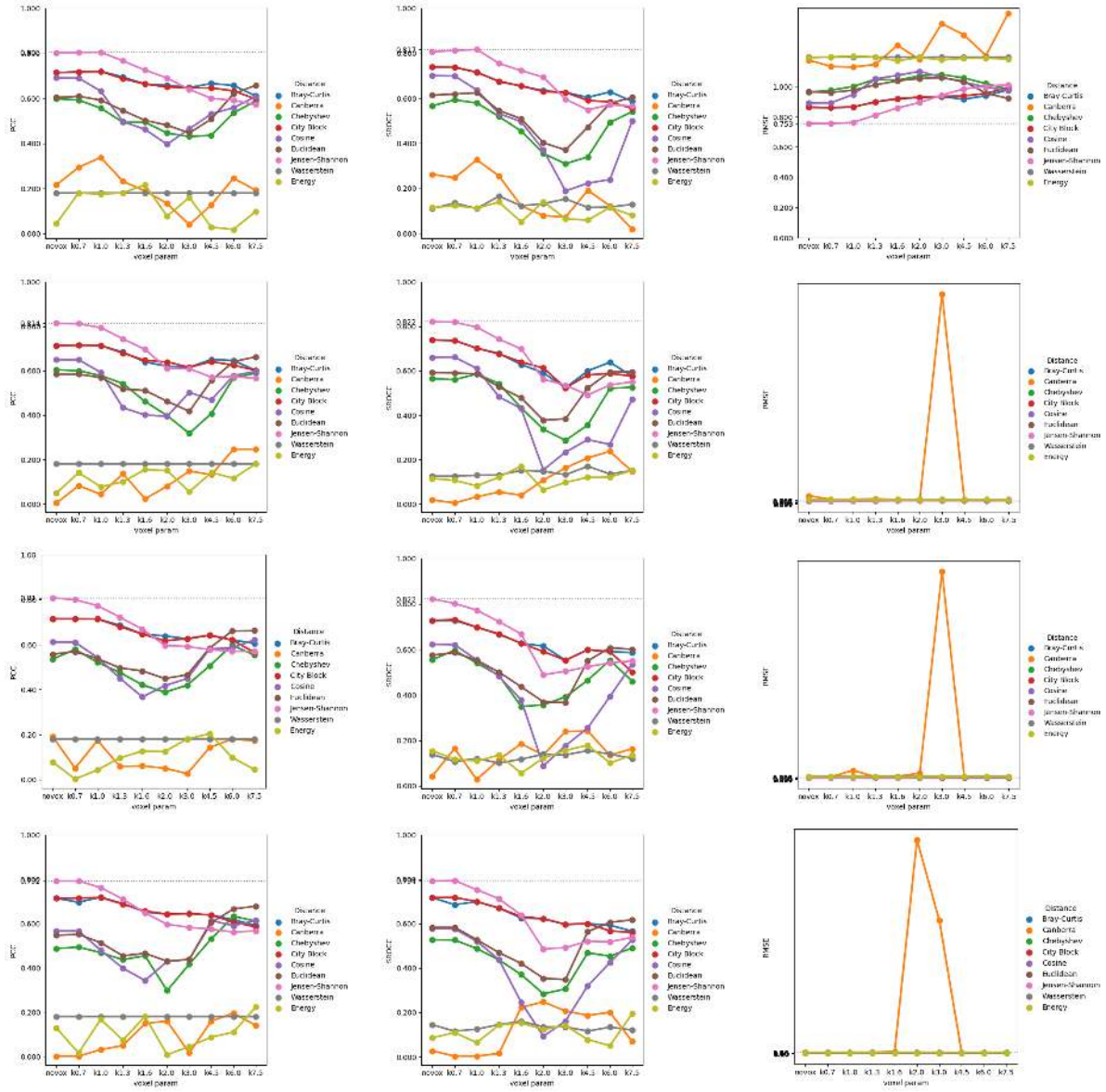


Figure 4.22: D3 geometry-based descriptor performance with different histogram distances evaluated, with each row representing the geometry-based descriptor with 6, 8, 10 and 12 neighbors, respectively.

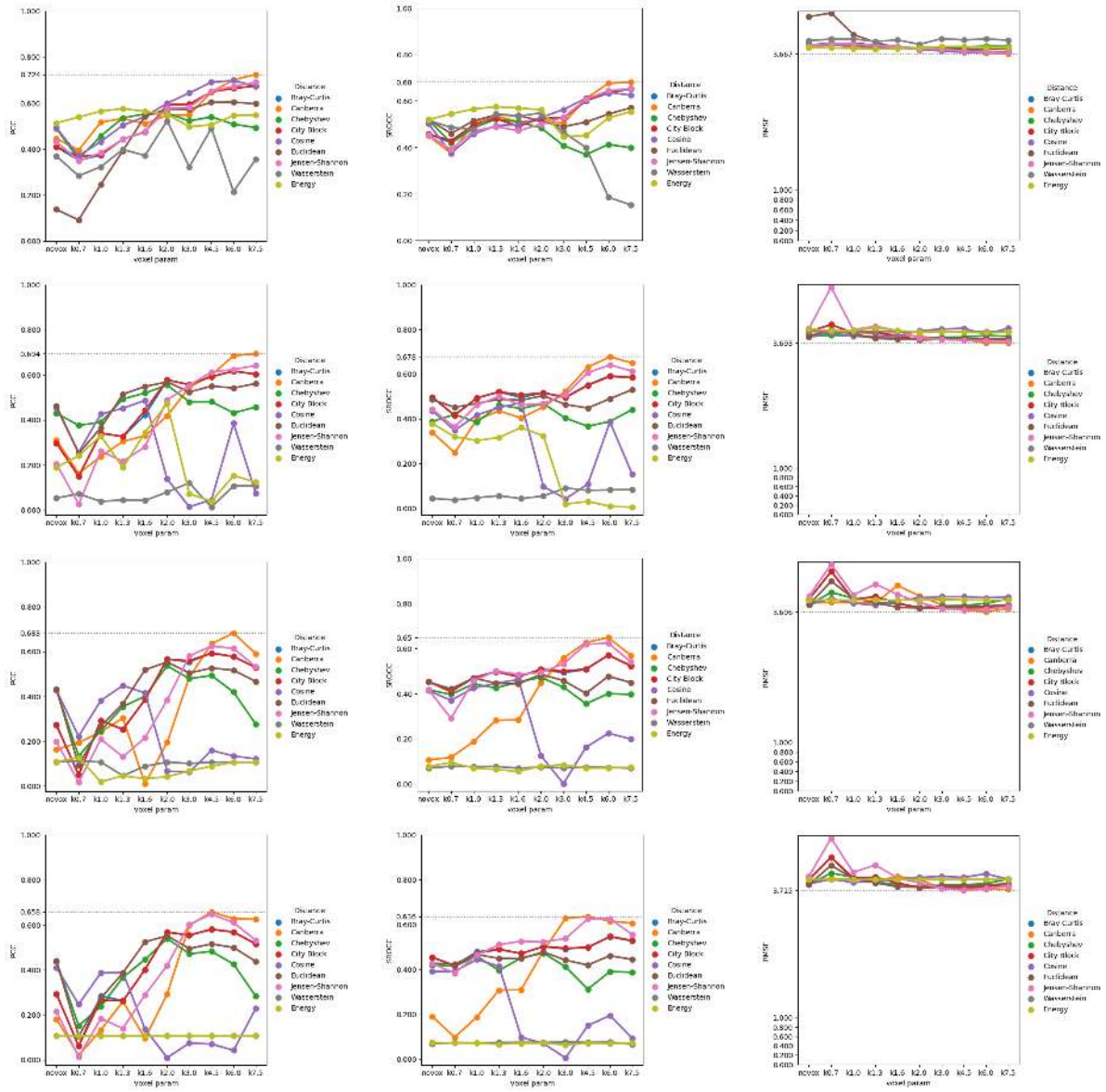


Figure 4.23: D4 LBP descriptor performance with different histogram distances evaluated, with each row representing the LBP with 6, 8, 10 and 12 neighbors, respectively.

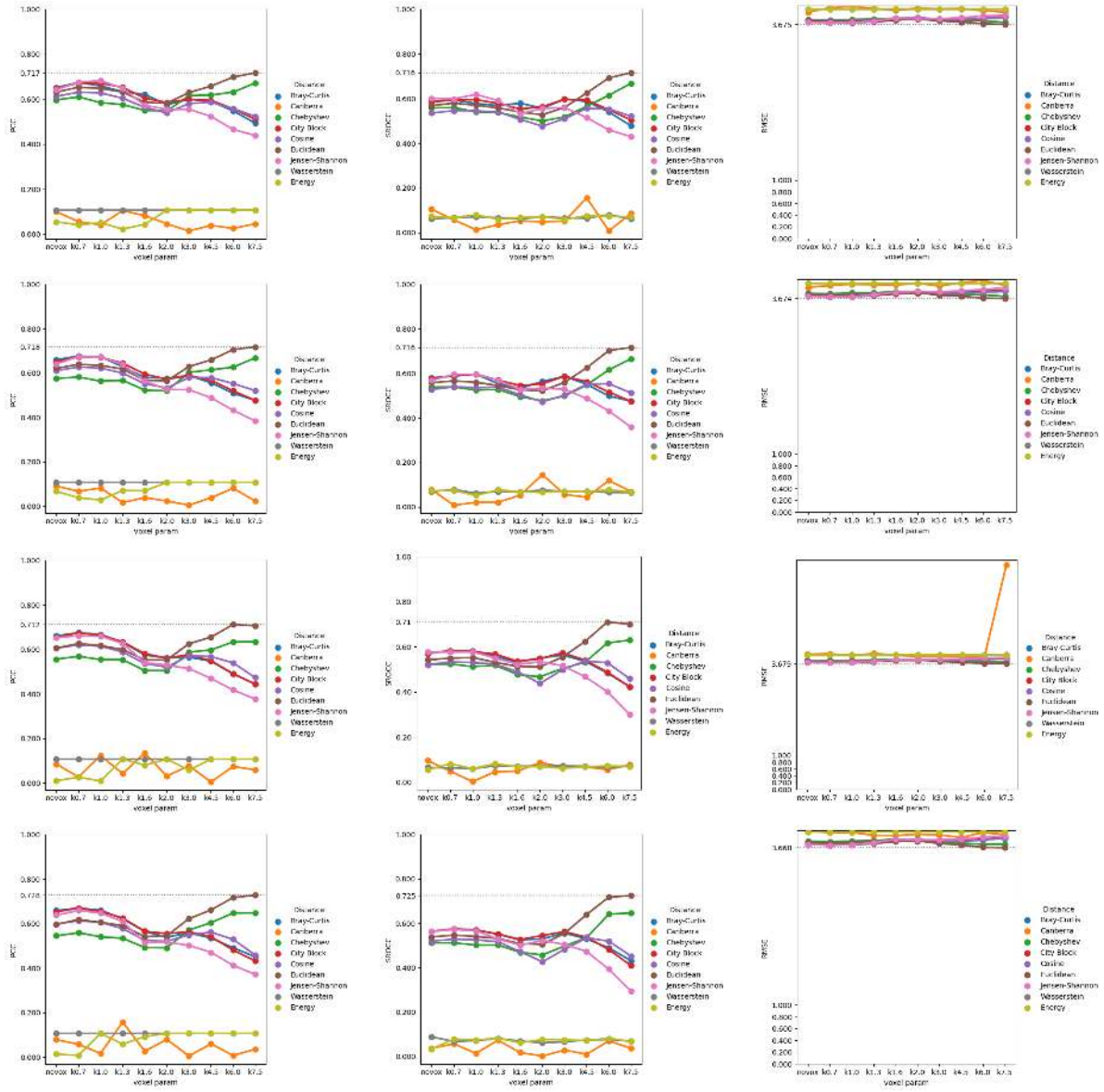


Figure 4.24: D4 LLP 16 bits descriptor performance with different histogram distances evaluated, with each row representing the LLP with 6, 8, 10 and 12 neighbors, respectively.

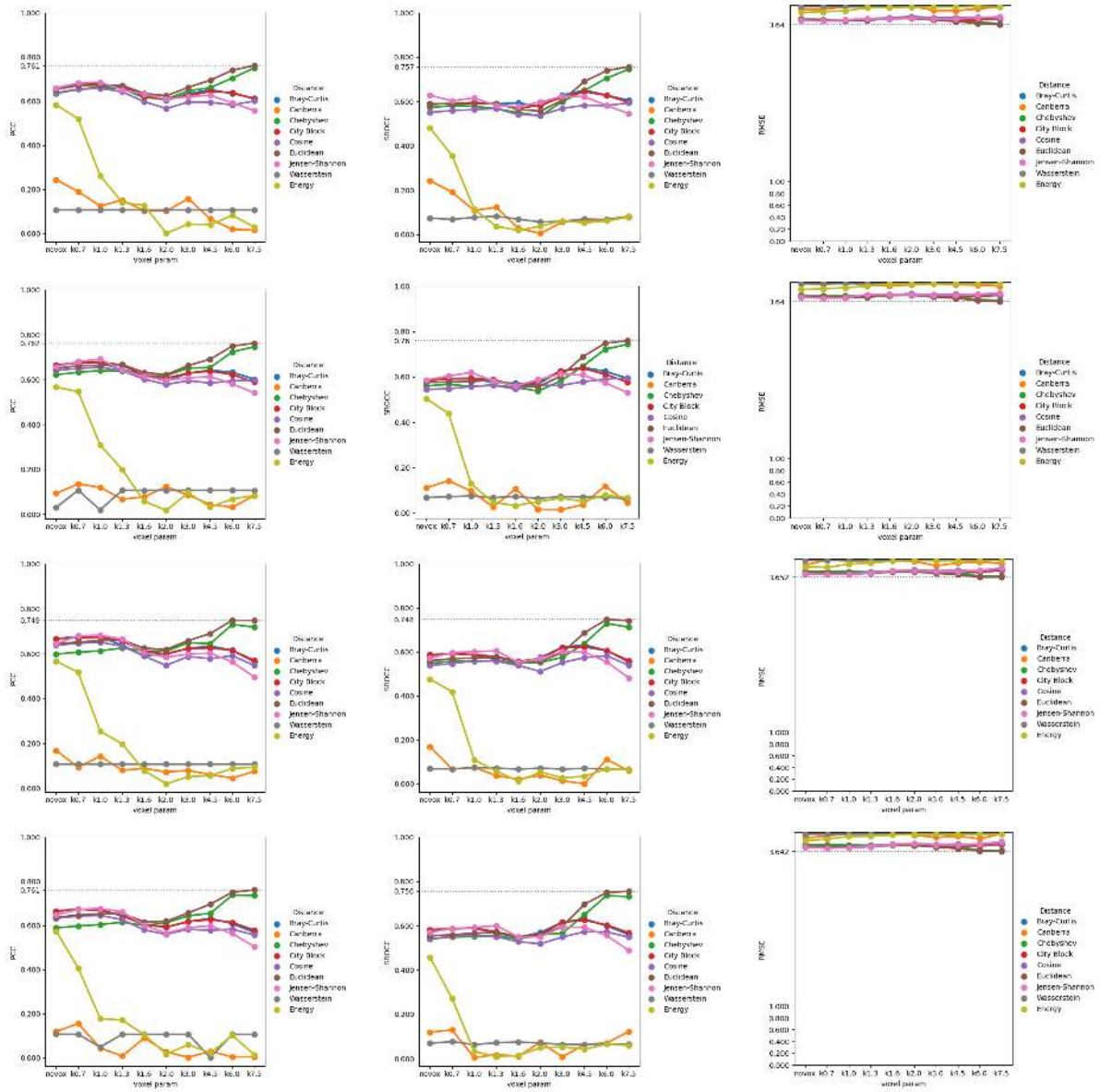


Figure 4.25: D4 LLP 12 bits descriptor performance with different histogram distances evaluated, with each row representing the LLP with 6, 8, 10 and 12 neighbors, respectively.

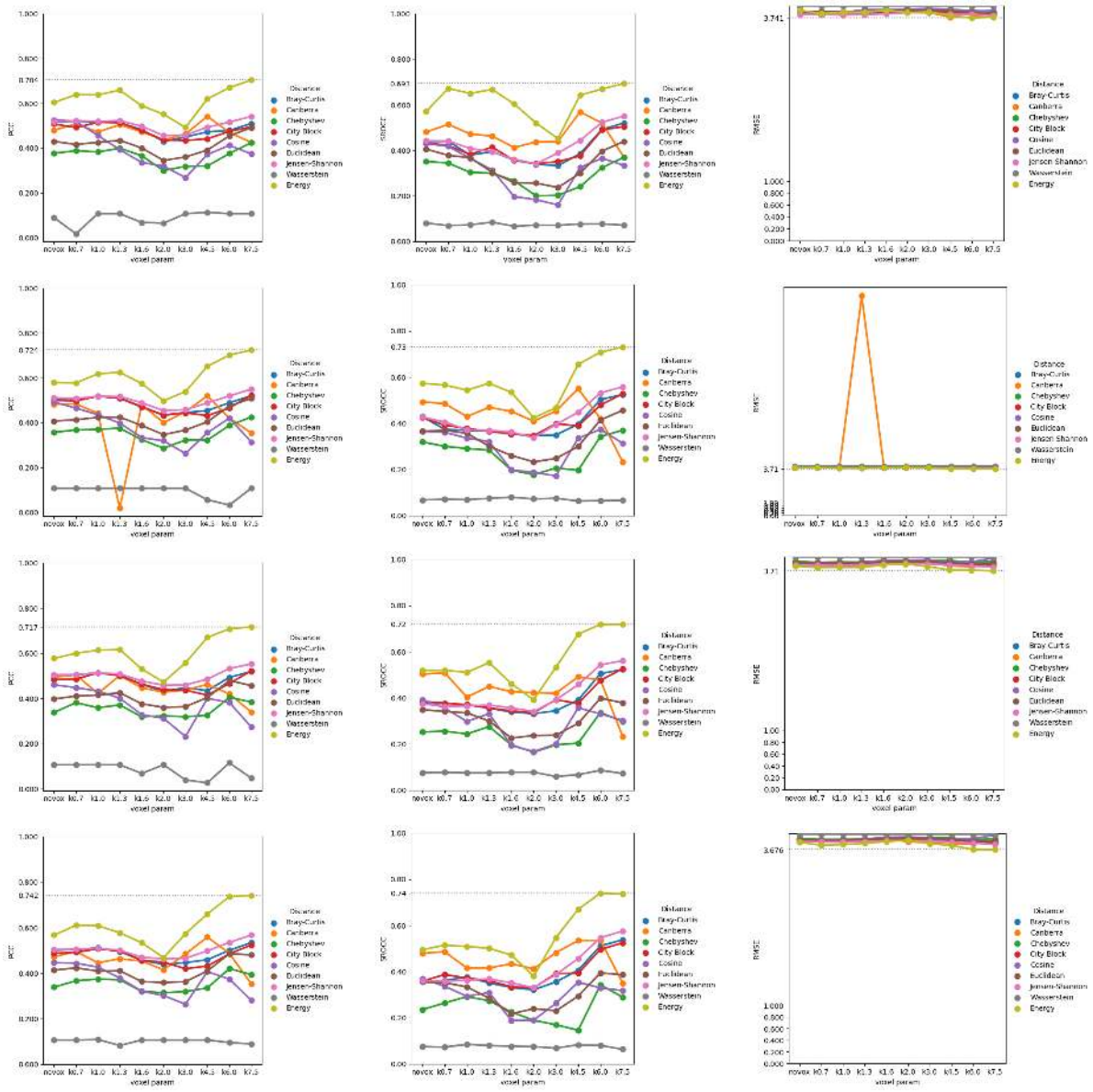


Figure 4.26: D4 LCP 12 bits descriptor performance with different histogram distances evaluated, with each row representing the LCP with 6, 8, 10 and 12 neighbors, respectively.

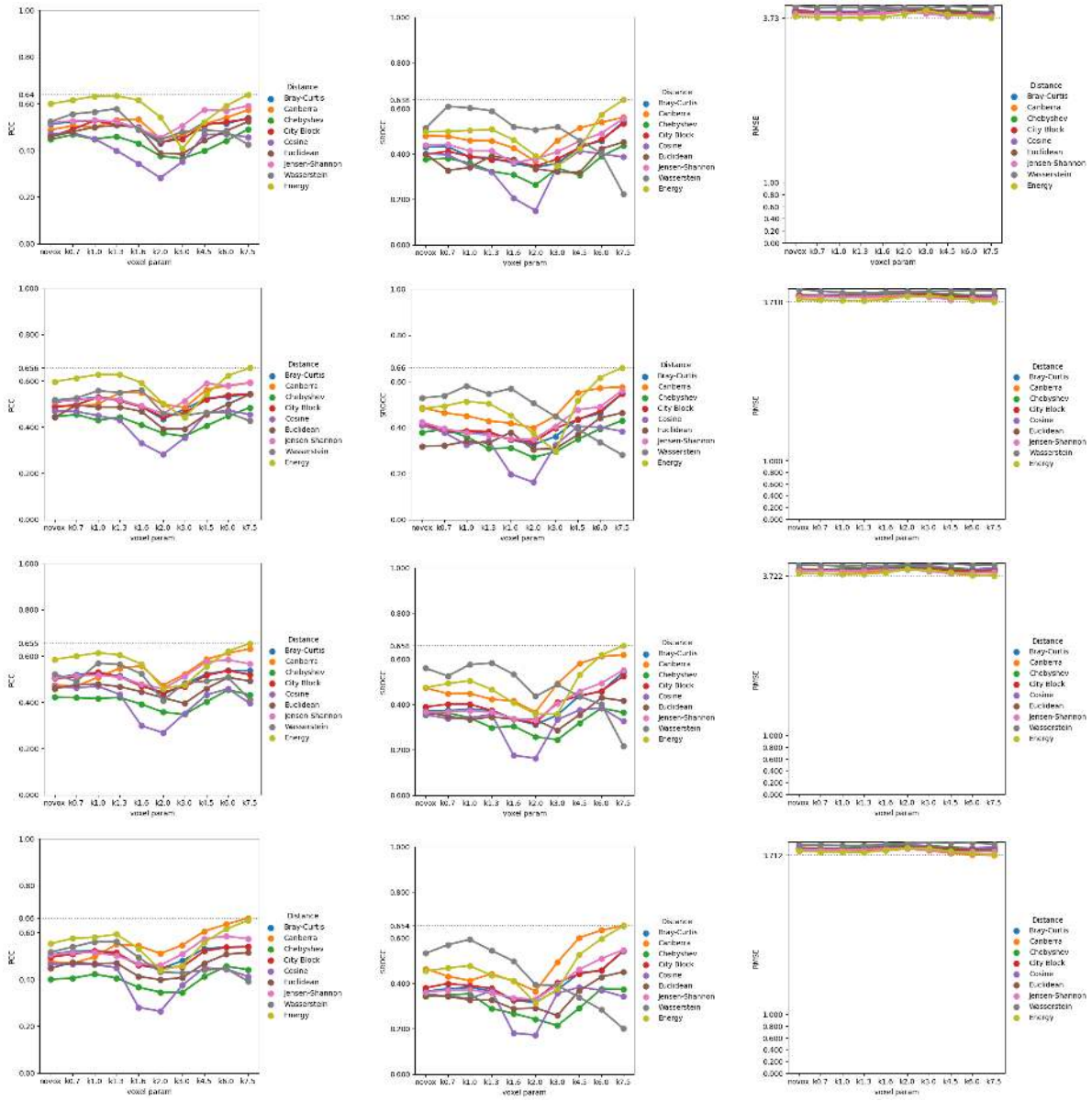


Figure 4.27: D4 LCP 8 bits descriptor performance with different histogram distances evaluated, with each row representing the LCP with 6, 8, 10 and 12 neighbors, respectively.

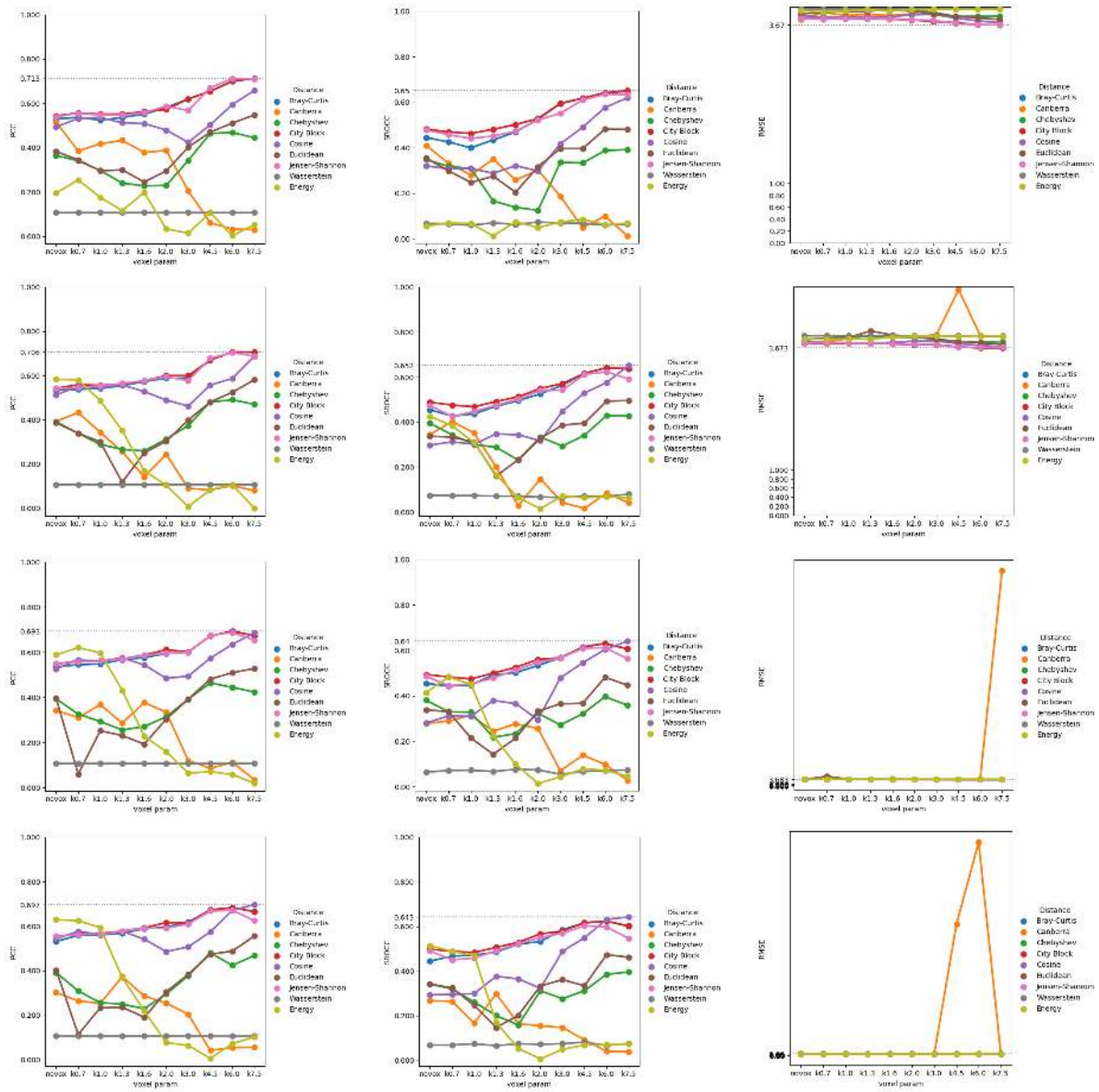


Figure 4.28: D4 geometry-based descriptor performance with different histogram distances evaluated, with each row representing the geometry-based descriptor with 6, 8, 10 and 12 neighbors, respectively.

The simulation results presented by the graphics of this section contain variations of all the descriptors, with varying size of neighborhood, voxel size and distance metrics. The simulation results show how the descriptors behave with the variation in the neighborhood size, the voxel size k parameter, and the different histogram distances.

In the case of the LBP color-based PC texture descriptor descriptor, which has a label size equal to the neighborhood size, in bits, the label is composed by the closer neighbor setting (or not) the most significant bit, while the farthest the neighbor, the less significant bit is enabled or not, as detailed in Section 3.1.2. As seen if Figures 4.5, 4.11, 4.17, 4.23, with the results obtained with datasets D1, D2, D3 and D4 respectively, the most prominent analysis is that the LBP performance is highly influenced by the voxelization, in all the datasets. While without the use of the voxelization the performance is poor, with k voxelization parameter between 2 and 6 the higher correlations are attained, specially with the histogram distances Canberra and Jensen-Shannon, which are the best performing distances for the LBP, according to the results. While for D1, D2 and D3 there is almost no different of the performance among different neighborhood sizes, for D4 the variation with 6 neighbors perform a bit better. LBP PCC performance peaks at 0.877, 0.878 and 0.907 in D1, D2 and D3 respectively, while in D4, a more complex dataset with wider variety of distortion, LBP PCC peaks at 0.724. The SROCC tendency is the same of the PCC, while the RMSE values mostly follows the inverse tendency of the correlation values.

Two variations of the LLP color-based PC texture descriptor were presented, a 16 bits version, and a 12 bits one. The 16 bits version results are presented in Figures 4.6, 4.12, 4.18 and 4.24, while the 12 bits version results are presented in Figure 4.7, 4.13, 4.19 and 4.25, for D1 to D4 datasets respectively. The LLP performance is less homogeneous among the datasets, with the histogram distance measures diverging considerably between each other. The Euclidean distance is the best distance for the 16 bits variant, while for 12 bit variant, there is no clear best, with the the Euclidean, Jensen-Shannon and Chebyshev alternating best performance depending on the dataset. The best voxelization parameter is between 4.5 and 7.5 for the 16 bits version, while for the 12 bits there is no clear best. The performance difference among the evaluated neighborhood sizes is not noticeable in both LLP variations. The LLP in its 16 bits variation has a PCC with peaks of 0.880, 0.839, 0.870 and 0.728, while the 12 bits version peaked at 0.834, 0.820, 0.884 and 0.762. The SROCC followed the same pattern of the PCC values, while the RMSE values havethe inverse pattern, as expected, with the exception of the Canberra RMSE for the LLP 16 bits, that is huge. The LLP 16 bits presented a more homogeneous behavior with respect to the best histogram distance measure and the voxelization operating range.

Also, two variations of the LCP color-based PC texture descriptor were presented in

the results, a 12 bits and a 8 bits variation. The 12 bits version results are presented in Figures 4.8, 4.14, 4.20, 4.26, while the 8 bits version results are shown in Figures 4.9, 4.15, 4.21 and 4.27, for D1 to D4 datasets respectively. The LCP 12 bits variant results presented no clear best version with respect to the histogram distance metric. While the Jensen-Shannon distance is best for D1, D2 and D3, the Energy is best for D4. In the case of the LCP 8 bits, the Jensen-Shannon is best for D1, D2 and D3, while the performance difference between Jensen-Shannon is much smaller to Energy in D4 than in the LCP 12 bits case. The voxelization improves the performance with most of the histogram distances tested, while not in the same intensity across different datasets. The LCP 12 bits had PCC peak values of 0.802, 0.780, 0.881 and 0.660, while the LCP 8 bits PCC peaked at 0.880, 0.775, 0.912 and 0.660, for D1 to D4 datasets respectively. The results show the performance of the LCP 8 bits slightly better than the 12 bits version, while the results with 12 neighbors present also a slightly better performance to both variants when compared to other neighborhood sizes. LCP SROCC correlation followed the same PCC values pattern, while the RMSE followed a inverse trend, as expected, with some exceptions to the Canberra distance, which presented at times very high RMSE peaks.

Finally, the geometry-based PC texture descriptor results are presented in Figures 4.10, 4.16, 4.22, 4.28, obtained with the application of the descriptor to datasets D1, D2, D3 and D4. The first divergence with concern to the other color-based descriptors is that the voxelization strongly degrades the performance of the geometry-based descriptor in datasets D2 and D3, for all histogram distances evaluated. In case of dataset D1, the voxelization improved very little the performance, and for dataset D4 the performance is a bit improved with the voxelization, but just for some histogram distance measures, notably Jensen-Shannon, Bray-Curtis and Cityblock. The results regarding the voxelization show that the voxelization is not suitable for the proposed geometry-based PC texture descriptor. Considering all the datasets, the best histogram distance is the Jensen-Shannon. The geometry-based descriptor PCC peaked at 0.777, 0.731, 0.814 and 0.713 for datasets D1 to D4 respectively, while the SROCC followed the same PCC curve trend, and RMSE the opposite trend, while again Canberra distance provided some very high RMSE peaks.

While the results show some good correlations, independently evaluating the color texture or the geometry texture is not an optimal solution for PC quality assessment, as this approach (only color or only geometry analysis) has no sufficient data to identify all possible color and geometry distortions. Towards a more complete quality assessment, which is the final proposal of this thesis, it is presented the joint performance results of one color-based texture and the geometry-based texture descriptors to provide the quality prediction. The procedure to combine two (or more) descriptors is done by the averaging of the distances of the histogram distance calculation. After the combination

of the histogram distances of the descriptor distances, the logistic regression calculation is done as already explained.

The methods proposed in this work were evaluated with varying parameters, with the goal of understanding the behavior of the descriptors for different datasets. After the evaluation of the texture descriptors, it was possible to fix the parameters so that they could be compared with state-of-the-art metrics. I adopted a fixed combination of one color-based texture descriptor and one geometry-based texture descriptor, in order to compare the proposed methods to the best available PCQA metrics.

One combination was done using the LCP with 8 bits label (correlation results shown in Figures 4.9, 4.15, 4.21 and 4.27), voxelization k parameter set to 6.0, neighborhood size of 12, together with the geometry-based descriptor (Figures 4.10, 4.16, 4.22, 4.28), no voxelization applied, and neighborhood size of 6. In both descriptor the Jensen-Shannon distance was used for histogram distances calculation. The second combination was done using the LBP, with voxelization k parameter set to 1.6, neighborhood size of 8, also with the same geometry-based descriptor, and using the Jensen-Shannon distance. The third experiment includes the LLP with 12 bits label, 8 neighbors, and k equal to 2.0, while the distance used to measure the LLP histograms was the Euclidean. The LLP with the aforementioned settings was combined with the geometry-based texture descriptor with the same parameters of the other combinations.

Table 4.1 shows the performance comparison of the LCP combined with the geometry-based descriptor (LCP + GEO in the table), LBP combined with the geometry-based descriptor (LCP + GEO), and LLP combined with the geometry-based descriptor (LLP + GEO). The conditions the other metrics were evaluated were exactly the same which were applied to the proposed methods, including exactly the same PC source content, same normal vectors (for the metrics which need normals), and the logistic regressor. Some MPEG metrics evaluate color channels independently, so in order to obtain a unique result (represented as YCbCr in the table) it is used the Eq. 2.6 to combine the color components. The point-to-point based metrics are the first 12 metrics, which are based on the MPEG released metrics. Then, there are also two recently released metrics that are considered in the comparison as state-of-the-art: the PCQM metric [80] and two modes of the PointSSIM metric [79]. Best values in the table are shown in bold, while second best are shown in italic. The last column of Table 4.1 shows the average values, which shows clearly that "LCP + GEO" metric has arguable the best performance, with averaged PCC of 0.840, an average SROCC of 0.845, and average RMSE of 1.462. The other best performing metrics, PCQM and PointSSIM-Color, have average PCC of 0.603 and 0.824, respectively, average SROCC of 0.873 and 0.822, respectively, and average RMSE of 3.616 and 3.160. While the PCQM has the best average SROCC by a small

Metrics	Data Sets														
	D1			D2			D3			D4			AVG		
	PCC	SROCC	RMSE	PCC	SROCC	RMSE	PCC	SROCC	RMSE	PCC	SROCC	RMSE	PCC	SROCC	RMSE
po2point_MSE	0.270	0.250	1.122	0.808	0.835	1.095	*0.941*	0.920	*0.534*	0.418	0.350	3.857	0.609	0.589	1.652
PSNR-po2point_MSE	0.518	0.484	0.953	0.494	0.430	1.352	0.538	0.549	1.025	0.470	0.376	3.832	0.505	0.460	1.791
po2point_Haus	0.270	0.215	1.122	0.627	0.421	1.282	0.496	0.446	1.024	0.261	0.224	3.900	0.414	0.327	1.832
PSNR-po2point_Haus	0.512	0.469	0.968	0.454	0.396	1.379	0.549	0.527	1.008	0.481	0.455	3.833	0.500	0.462	1.797
Color-YCbCr_MSE	0.383	0.367	1.039	0.553	0.571	1.333	0.755	0.682	0.921	0.500	0.512	3.822	0.548	0.533	1.779
PSNR-Color-YCbCr_MSE	0.368	0.337	1.097	0.536	0.565	1.351	0.793	0.801	0.797	0.504	0.503	*3.805*	0.550	0.552	1.763
Color-YCbCr_Haus	0.147	0.172	1.131	0.413	0.375	1.380	0.377	0.306	1.122	0.191	0.095	3.955	0.282	0.237	1.897
PSNR-Color-YCbCr_Haus	0.386	0.320	1.059	0.435	0.391	1.417	0.445	0.449	1.100	0.344	0.270	3.875	0.403	0.358	1.863
po2plane_MSE	0.270	0.275	1.122	*0.845*	0.858	**1.031**	**0.958**	*0.945*	**0.492**	0.432	0.370	3.859	0.626	0.612	*1.626*
PSNR-po2plane_MSE	0.484	0.421	0.984	0.499	0.495	1.361	0.542	0.579	1.021	0.380	0.390	3.893	0.476	0.471	1.815
po2plane_Hausdorff	0.270	0.247	1.122	0.604	0.427	1.267	0.586	0.418	0.981	0.223	0.188	3.990	0.421	0.320	1.840
PSNR-po2plane_Haus	0.440	0.408	1.016	0.428	0.367	1.394	0.497	0.463	1.034	0.464	0.451	3.836	0.457	0.422	1.820
PCQM	0.797	**0.898**	2.656	0.607	*0.915*	2.899	0.738	**0.970**	3.123	0.271	0.708	5.786	0.603	**0.873**	3.616
PointSSIM-Color	0.842	0.823	2.234	**0.910**	**0.918**	2.436	0.869	0.865	2.697	*0.676*	*0.682*	5.354	*0.824*	0.822	3.180
PointSSIM-Geometry	0.804	0.820	2.102	0.784	0.834	2.321	0.849	0.905	2.534	0.527	0.560	5.323	0.741	0.780	3.070
LCP + GEO	**0.876**	*0.896*	**0.572**	0.819	0.839	1.068	0.936	0.932	0.544	**0.730**	**0.714**	**3.663**	**0.840**	*0.845*	**1.462**
LBP + GEO	*0.845*	0.837	*0.620*	*0.845*	0.850	*1.037*	0.863	0.869	0.672	0.579	0.543	3.764	0.783	0.775	1.523
LLP + GEO	0.790	0.795	0.702	0.812	0.822	1.077	0.873	0.877	0.651	0.672	0.660	3.705	0.787	0.789	1.534

Table 4.1: Performance of joint analysis of two proposed descriptors compared with other state-of-the-art metrics applied.

margin, its PCC is 0.603, which is much lower than the "LCP+GEO" proposed method. Also the RMSE of the proposed method is by far the smallest.

Chapter 5

Conclusions

In this chapter we present the conclusions taken from the work, and further work to be done. While none of the descriptors alone can capture all types of distortions, when two descriptors are used together, the performance of the propose metric is up to par with the state-of-art, if not better. The contributions of this work include the voxelization process, the four proposed texture descriptors, and the analysis of the texture histogram distances and of the quality prediction model based.

The results show the relevance of the voxelization process, which is the process that provides a volume to each PC point, so that these points can be rendered and be visualized. But, for the purpose of quality assessment, voxelization is used to emulate the rendering process and improve the texture descriptor performance. The voxelization process contains a certain amount of uncertainty, as the subjective scores used as the basis for the comparisons are obtained through different types of rendering techniques, which can affect negatively the design of PC objective metrics. On the other hand, this lead me to develop a heuristic study to define the voxel size, as discussed in Section 3.1.1. While the voxelization applied to color-based texture descriptors improves the correlation of the proposed metric with the perceived quality, the voxelization does not improve the performace of the geometry-based texture descriptor,

Concerning the proposed texture descriptors, all of them have a fair performance when analyzed independently, when compared to the MPEG proposed metrics. While independently the color-based texture descriptors perform better than the geometry-based texture descriptor, none of them outperforms the combined LCP and the geometry-based texture descriptors together.

The LBP has a label size defined by the size of the neighborhood, while the other proposed descriptors have a label size independent than the neighborhood size. The LCP needs less bits to represent the color texture than the LLP descriptor, as the ranges associated to the bits in the label are based on the perceptual comparison using the

CIEDE2000 distance, while allows a higher compression of the texture information. The totally new texture descriptors are a relevant contribution to the state-of-the-art, while the LCP and the geometry-based proposals provide the best performance when used together. Also, as a side-effect of the way the proposed texture descriptors extract the labels, is that the descriptors are scale and rotational invariant, as they are based on a neighborhood defined by relative distances.

Future works include the optimization of the performance of the proposed method. The proposed optimization method will be based on a profound statistical analysis of the available PC datasets. The adoption of a fixed k voxelization parameter for the purpose of establishing PCQA metrics is not optimal, as the rendering process is not standardized across different datasets. The simulation results show that the optimal k which leads to higher correlation results vary between datasets, so a method for automatic setting the voxelization parameter k will be developed. Also a more adapted ranges associated to each descriptor label bits will be proposed, based on the more profound statistics analysis. Also, a no-reference PCQA method will be proposed. The NR metric will be based on machine learning techniques, which use the texture descriptor histograms as input, as opposed to using the histogram distance between reference and test PCs.

Clearly this work provides results that are good with different types of datasets and distortions, while providing a powerful framework for PC quality assessment, in the spite of the state-of-the-art PC quality assessment metrics. This proposal is among the best state-of-the-art PCQA metrics, which I plan to present to standardization bodies as a metric to be used for objective quality measures of PCs. The ultimate goal of this work was to contribute for a wider adoption of visual volumetric media, which will allow for a much more realistic representation of the world.

References

[1] Adelson, Edward H, James R Bergen, *et al.*: *The plenoptic function and the elements of early vision*. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1991. xii, 6, 7

[2] Milgram, Paul and Fumio Kishino: *A taxonomy of mixed reality visual displays*. IEICE TRANSACTIONS on Information and Systems, 77(12):1321–1329, 1994. xii, 10, 11

[3] Diniz+, Rafael and Myléne CQ Farias: *Real-time 3d volumetric human body reconstruction from a single view rgb-d capture device*. Electronic Imaging, 2019(16):6–1, 2019. xii, 4, 13, 14

[4] Freitas, Pedro Garcia: *Using Texture Measures for Visual Quality Assessment*. PhD thesis, University of Brasília, Brasília, September 2017. xiii, 3, 23

[5] ISO/IEC 23090-5: *Information technology – Coded representation of immersive media – Part 5: Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC)*. International Organization for Standardization, Geneva, Switzerland, June 2021. `https://www.iso.org/standard/73025.html`. 1, 11, 21

[6] CISCO: *Global networking trends report*. 2021. 2

[7] Diniz, Rafael, Pedro Garcia Freitas, and Mylene CQ Farias: *Towards a point cloud quality assessment model using local binary patterns*. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6. IEEE, 2020. 4

[8] Diniz, Rafael, Pedro Garcia Freitas, and Mylene CQ Farias: *Multi-distance point cloud quality assessment*. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1–5. IEEE, 2020. 4

[9] Diniz, Rafael, Pedro Garcia Freitas, and Mylène CQ Farias: *Local luminance patterns for point cloud quality assessment*. In *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2020. 4, 44

[10] Diniz, Rafael, Mylene Q Farias, and Pedro Garcia-Freitas: *Color and geometry texture descriptors for point-cloud quality assessment*. IEEE Signal Processing Letters, 2021. 4, 5

[11] Diniz, Rafael, Pedro Garcia Freitas, and Mylène Farias: *A novel point cloud quality assessment metric based on perceptual color distance patterns*. Electronic Imaging, 2021(9):256–1, 2021. 4

[12] Gabor, Dennis: *A new microscopic principle*. nature, 161:777–778, 1948. 7

[13] Goodman, Joseph W and RW Lawrence: *Digital image formation from electronically detected holograms*. Applied physics letters, 11(3):77–79, 1967. 7

[14] Tahara, Tatsuki, Akifumi Maeda, Yasuhiro Awatsuji, Takashi Kakue, Peng Xia, Kenzo Nishio, Shogo Ura, Toshihiro Kubota, and Osamu Matoba: *Single-shot dual-illumination phase unwrapping using a single wavelength*. Optics letters, 37(19):4002–4004, 2012. 7

[15] Xia, Peng, Qinghua Wang, Shien Ri, and Hiroshi Tsuda: *Calibrated phase-shifting digital holography based on a dual-camera system*. Optics letters, 42(23):4954–4957, 2017. 7

[16] Xia, Peng, Qinghua Wang, and Shien Ri: *Random phase-shifting digital holography based on a self-calibrated system*. Optics Express, 28(14):19988–19996, 2020. 7

[17] Wu, Gaochang, Belen Masia, Adrian Jarabo, Yuchen Zhang, Liangyong Wang, Qionghai Dai, Tianyou Chai, and Yebin Liu: *Light field image processing: An overview*. IEEE Journal of Selected Topics in Signal Processing, 11(7):926–954, 2017. 7

[18] Domański, Marek, Olgierd Stankiewicz, Krzysztof Wegner, and Tomasz Grajek: *Immersive visual media—mpeg-i: 360 video, virtual navigation and beyond*. In *2017 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 1–9. IEEE, 2017. 8

[19] Graziosi, D, O Nakagami, S Kuma, A Zaghetto, T Suzuki, and A Tabatabai: *An overview of ongoing point cloud compression standardization activities: Video-based (v-pcc) and geometry-based (g-pcc)*. APSIPA Transactions on Signal and Information Processing, 9, 2020. 10, 11

[20] Hinks, Tommy, Hamish Carr, Linh Truong-Hong, and Debra F Laefer: *Point cloud data conversion into solid models via point-based voxelization*. Journal of Surveying Engineering, 139(2):72–83, 2012. 10

[21] Sugimoto, Kazuo, Robert A Cohen, Dong Tian, and Anthony Vetro: *Trends in efficient representation of 3d point clouds*. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017*, pages 364–369. IEEE, 2017. 11

[22] ISO/IEC DIS 23090-9 (E): *Information technology – Coded representation of immersive media – Part 9: Geometry-based point cloud compression*. International Organization for Standardization, Geneva, Switzerland, June 2020. `https://www.iso.org/standard/78990.html`. 11, 21

[23] Schnabel, Ruwen and Reinhard Klein: *Octree-based point-cloud compression.* Spbg, 6:111–120, 2006. 12

[24] Dricot, Antoine, Fernando Pereira, and João Ascenso: *Rate-distortion driven adaptive partitioning for octree-based point cloud geometry coding.* In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2969–2973. IEEE, 2018. 12

[25] Merry, Bruce, Patrick Marais, and James Gain: *Compression of dense and regular point clouds.* In *Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pages 15–20. ACM, 2006. 12

[26] Shao, Yiting, Zhaobin Zhang, Zhu Li, Kui Fan, and Ge Li: *Attribute compression of 3d point clouds using laplacian sparsity optimized graph transform.* arXiv preprint arXiv:1710.03532, 2017. 12

[27] Cohen, Robert A, Dong Tian, and Anthony Vetro: *Attribute compression for sparse point clouds using graph transforms.* In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 1374–1378. IEEE, 2016. 12

[28] Dricot, Antoine and João Ascenso: *Adaptive multi-level triangle soup for geometry-based point cloud coding.* In *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2019. 12

[29] Kowalski, Marek, Jacek Naruniec, and Michal Daniluk: *Live scan3d: A fast and inexpensive 3d data acquisition system for multiple kinect v2 sensors.* In *3D Vision (3DV), 2015 International Conference on*, pages 318–325. IEEE, 2015. 12

[30] Seitz, Steven M, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski: *A comparison and evaluation of multi-view stereo reconstruction algorithms.* In *null*, pages 519–528. IEEE, 2006. 12

[31] Berger, Matthew, Andrea Tagliasacchi, Lee M Seversky, Pierre Alliez, Gael Guennebaud, Joshua A Levine, Andrei Sharf, and Claudio T Silva: *A survey of surface reconstruction from point clouds.* In *Computer Graphics Forum*, volume 36, pages 301–329. Wiley Online Library, 2017. 12

[32] Firman, Michael, Oisin Mac Aodha, Simon Julier, and Gabriel J. Brostow: *Structured prediction of unobserved voxels from a single depth image.* In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 13

[33] Alexiadis, Dimitrios S, Nikolaos Zioulis, Dimitrios Zarpalas, and Petros Daras: *Fast deformable model-based human performance capture and fvv using consumer-grade rgb-d sensors.* Pattern Recognition, 79:260–278, 2018. 13

[34] Bondi, Enrico, Pietro Pala, Stefano Berretti, and Alberto Del Bimbo: *Reconstructing high-resolution face models from kinect depth sequences.* IEEE Transactions on Information Forensics and Security, 11(12):2843–2853, 2016. 13

[35] Choi, Jongmoo, Gerard Medioni, Yuping Lin, Luciano Silva, Olga Regina, Mauricio Pamplona, and Timothy C Faltemier: *3d face reconstruction using a single or multiple views*. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3959–3962. IEEE, 2010. 13

[36] Jiang, Luo, Juyong Zhang, Bailin Deng, Hao Li, and Ligang Liu: *3d face reconstruction with geometry details from a single image*. IEEE Transactions on Image Processing, 27(10):4756–4770, 2018. 13

[37] Chang, Angel X, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, *et al.*: *Shapenet: An information-rich 3d model repository*. arXiv preprint arXiv:1512.03012, 2015. 13

[38] Thanh Nguyen, Duc, Binh Son Hua, Khoi Tran, Quang Hieu Pham, and Sai Kit Yeung: *A field model for repairing 3d shapes*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5676–5684, 2016. 13

[39] Rock, Jason, Tanmay Gupta, Justin Thorsen, JunYoung Gwak, Daeyun Shin, and Derek Hoiem: *Completing 3d object shape from one depth image*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2484–2493, 2015. 13

[40] Song, Shuran, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser: *Semantic scene completion from a single depth image*. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 190–198. IEEE, 2017. 13

[41] Hernandez, Matthias, Jongmoo Choi, and Gérard Medioni: *Near laser-scan quality 3-d face reconstruction from a low-quality depth stream*. Image and Vision Computing, 36:61–69, 2015. 13

[42] Newcombe, Richard A, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon: *Kinectfusion: Real-time dense surface mapping and tracking*. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011. 13

[43] Salvi, Joaquim, Sergio Fernandez, Tomislav Pribanic, and Xavier Llado: *A state of the art in structured light patterns for surface profilometry*. Pattern recognition, 43(8):2666–2680, 2010. 14

[44] Lange, Robert and Peter Seitz: *Solid-state time-of-flight range camera*. IEEE Journal of quantum electronics, 37(3):390–397, 2001. 14

[45] Zhang, Zhengyou: *Microsoft kinect sensor and its effect*. IEEE multimedia, 19(2):4–10, 2012. 14

[46] Pagliari, Diana and Livio Pinto: *Calibration of kinect for xbox one and comparison between the two generations of microsoft sensors*. Sensors, 15(11):27569–27589, 2015. 16

[47] Bamji, Cyrus S, Swati Mehta, Barry Thompson, Tamer Elkhatib, Stefan Wurster, Onur Akkaya, Andrew Payne, John Godbaz, Mike Fenton, Vijay Rajasekaran, *et al.*: *Impixel 65nm bsi 320mhz demodulated tof image sensor with 3µm global shutter pixels and analog binning*. In *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*, pages 94–96. IEEE, 2018. 16

[48] Tian, Yuan, Yuxin Ma, Shuxue Quan, and Yi Xu: *Occlusion and collision aware smartphone ar using time-of-flight camera*. In *International Symposium on Visual Computing*, pages 141–153. Springer, 2019. 16

[49] Lee, Byoungho, Dongheon Yoo, Jinsoo Jeong, Kiseung Bang, and Seokil Moon: *Ultra-high-definition holography for near-eye display*. In *Ultra-High-Definition Imaging Systems III*, volume 11305, page 113050L. International Society for Optics and Photonics, 2020. 16

[50] Orts-Escolano, Sergio, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, *et al.*: *Holoportation: Virtual 3d teleportation in real-time*. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 741–754. ACM, 2016. 19

[51] ITU-R: *Recommendation bt.500-14 (10/2019): Methodologies for the subjective assessment of the quality of television images*. International Telecommunication Union, Geneva, 2019. 19

[52] Dumic, Emil, Carlos Rafael Duarte, and Luis A da Silva Cruz: *Subjective evaluation and objective measures for point clouds—state of the art*. In *2018 First International Colloquium on Smart Grid Metrology (SmaGriMet)*, pages 1–5. IEEE, 2018. 19, 21

[53] Javaheri, Alireza, Catarina Brites, Fernando Manuel Bernardo Pereira, and Joao M Ascenso: *Point cloud rendering after coding: Impacts on subjective and objective quality*. IEEE Transactions on Multimedia, 2020. 20, 24, 25

[54] Alexiou, Evangelos and Touradj Ebrahimi: *On the performance of metrics to predict quality in point cloud representations*. In *Applications of Digital Image Processing XL*, volume 10396, page 103961H. International Society for Optics and Photonics, 2017. 20, 26

[55] Zhang, Juan, Wenbin Huang, Xiaoqiang Zhu, and Jenq Neng Hwang: *A subjective quality evaluation for 3d point cloud models*. In *Audio, Language and Image Processing (ICALIP), 2014 International Conference on*, pages 827–831. IEEE, 2014. 20

[56] Torlig, Eric M, Evangelos Alexiou, Tiago A Fonseca, Ricardo L de Queiroz, and Touradj Ebrahimi: *A novel methodology for quality assessment of voxelized point clouds*. In *Applications of Digital Image Processing XLI*, volume 10752, page 107520I. International Society for Optics and Photonics, 2018. 20, 21, 26, 45

[57] Mekuria, Rufael, Kees Blom, and Pablo Cesar: *Design, implementation, and evaluation of a point cloud codec for tele-immersive video.* IEEE Transactions on Circuits and Systems for Video Technology, 27(4):828–842, 2017. 20

[58] Javaheri, Alireza, Catarina Brites, Fernando Pereira, and João Ascenso: *Subjective and objective quality evaluation of 3d point cloud denoising algorithms.* In *Multimedia & Expo Workshops (ICMEW), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017. 20, 25

[59] Javaheri, A., C. Brites, F. Pereira, and J. Ascenso: *Subjective and objective quality evaluation of compressed point clouds.* In *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6, 2017. 20

[60] Alexiou, Evangelos, 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)*, pages 1–6. IEEE, 2017. 20

[61] Alexiou, Evangelos and Touradj Ebrahimi: *On subjective and objective quality evaluation of point cloud geometry.* In *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–3. IEEE, 2017. 20, 39

[62] Alexiou, Evangelos, Marco V Bernardo, Luis A da Silva Cruz, Lovorka Gotal Dmitrovic, Carlos Duarte, Emil Dumic, Touradj Ebrahimi, Dragan Matkovic, Manuela Pereira, Antonio Pinheiro, *et al.*: *Point cloud subjective evaluation methodology based on 2d rendering.* In *10th International Conference on Quality of Multimedia Experience (QoMEX)*, number CONF, 2018. 20, 21

[63] Alexiou, Evangelos and Touradj Ebrahimi: *Impact of visualisation strategy for subjective quality assessment of point clouds.* In *2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2018. 21

[64] Alexiou, Evangelos, Irene Viola, Tomás M Borges, Tiago A Fonseca, Ricardo L De Queiroz, and Touradj Ebrahimi: *A comprehensive study of the rate-distortion performance in mpeg point cloud compression.* APSIPA Transactions on Signal and Information Processing, 8, 2019. 21, 26, 45

[65] Christaki, Kyriaki, Emmanouil Christakis, Petros Drakoulis, Alexandros Doumanoglou, Nikolaos Zioulis, Dimitrios Zarpalas, and Petros Daras: *Subjective visual quality assessment of immersive 3d media compressed by open-source static 3d mesh codecs.* In *International Conference on Multimedia Modeling*, pages 80–91. Springer, 2019. 21

[66] Perry, Stuart, Huy Phi Cong, Luís A da Silva Cruz, João Prazeres, Manuela Pereira, Antonio Pinheiro, Emil Dumic, Evangelos Alexiou, and Touradj Ebrahimi: *Quality evaluation of static point clouds encoded using mpeg codecs.* In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3428–3432. IEEE, 2020. 21, 22, 25, 45

[67] Yang, Qi, Hao Chen, Zhan Ma, Yiling Xu, Rongjun Tang, and Jun Sun: *Predicting the perceptual quality of point cloud: A 3d-to-2d projection-based exploration.* IEEE Transactions on Multimedia, 2020. 21, 22, 38, 45

[68] Tian, Dong, Hideaki Ochimizu, Chen Feng, Robert Cohen, and Anthony Vetro: *Geometric distortion metrics for point cloud compression.* In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3460–3464. IEEE, 2017. 23, 25

[69] Mekuria, RN, Zhu Li, C Tulvan, and P Chou: *Evaluation criteria for pcc (point cloud compression).* ISO/IEC MPEG Doc. N16332, 2016. 23

[70] Alexiou, Evangelos and Touradj Ebrahimi: *Point cloud quality assessment metric based on angular similarity.* In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2018. 23

[71] Ohm, Jens Rainer, Gary J Sullivan, Heiko Schwarz, Thiow Keng Tan, and Thomas Wiegand: *Comparison of the coding efficiency of video coding standards—including high efficiency video coding (hevc).* IEEE Transactions on circuits and systems for video technology, 22(12):1669–1684, 2012. 25

[72] Javaheri, Alireza, Catarina Brites, Fernando Pereira, and João Ascenso: *A generalized hausdorff distance based quality metric for point cloud geometry.* In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6. IEEE, 2020. 25, 26

[73] Group, MPEG 3DG: *Common test conditions for point cloud compression.* ISO/IEC JTC1/SC29/WG11 Doc. N18474, 2019. 25

[74] Viola, Irene, Shishir Subramanyam, and Pablo Cesar: *A color-based objective quality metric for point cloud contents.* In *2020 12th International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6. IEEE, 2020. 26, 27

[75] Tian, D, H Ochimizu, C Feng, R Cohen, and A Vetro: *Updates and integration of evaluation metric software for pcc.* ISO/IEC JTC1/SC29/WG11 input document MPEG2017 M, 40522, 2017. 26

[76] Javaheri, Alireza, Catarina Brites, Fernando Pereira, and João Ascenso: *Mahalanobis based point to distribution metric for point cloud geometry quality evaluation.* IEEE Signal Processing Letters, 27:1350–1354, 2020. 26

[77] Meynet, Gabriel, Julie Digne, and Guillaume Lavoué: *Pc-msdm: A quality metric for 3d point clouds.* In *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–3. IEEE, 2019. 26

[78] Yang, Qi, Zhan Ma, Yiling Xu, Zhu Li, and Jun Sun: *Inferring point cloud quality via graph similarity.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020. 26

[79] Alexiou, Evangelos and Touradj Ebrahimi: *Towards a point cloud structural similarity metric.* In *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2020. 26, 78

[80] Meynet, Gabriel, Yana Nehme, Julie Digne, and Guillaume Lavoué: *PCQM: A full-reference quality metric for colored 3d point clouds*. In *2020 12th International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6. IEEE, 2020. 27, 78

[81] Bello, Saifullahi Aminu, Shangshu Yu, Cheng Wang, Jibril Muhmmad Adam, and Jonathan Li: *deep learning on 3d point clouds*. Remote Sensing, 12(11):1729, 2020. 27

[82] Liu, Yipeng, Qi Yang, Yiling Xu, and Le Yang: *Point cloud quality assessment: Large-scale dataset construction and learning-based no-reference approach*. arXiv preprint arXiv:2012.11895, 2020. 27

[83] Euclid: *Elements*. Alexandria, c. 300 B.C. 29, 42

[84] Ojala, Timo, Matti Pietikainen, and Topi Maenpaa: *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*. IEEE Transactions on pattern analysis and machine intelligence, 24(7):971–987, 2002. 31

[85] Zhang, Min, Chisako Muramatsu, Xiangrong Zhou, Takeshi Hara, and Hiroshi Fujita: *Blind image quality assessment using the joint statistics of generalized local binary pattern*. IEEE Signal Processing Letters, 22(2):207–210, 2014. 31

[86] Freitas, Pedro Garcia, Welington YL Akamine, and Mylène CQ Farias: *Blind image quality assessment using multiscale local binary patterns*. Electronic Imaging, 2017(12):7–14, 2017. 31

[87] Freitas, Pedro Garcia, Welington YL Akamine, and Mylène CQ Farias: *Using multiple spatio-temporal features to estimate video quality*. Signal Processing: Image Communication, 64:1–10, 2018. 31

[88] Luo, M Ronnier, Guihua Cui, and Bryan Rigg: *The development of the cie 2000 colour-difference formula: Ciede2000*. Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur, 26(5):340–350, 2001. 37

[89] Jurman, Giuseppe, Samantha Riccadonna, Roberto Visintainer, and Cesare Furlanello: *Canberra distance on ranked lists*. In *Proceedings of advances in ranking NIPS 09 workshop*, pages 22–27. Citeseer, 2009. 42, 51

[90] Mandal, Bappaditya, Zhikai Wang, Liyuan Li, and Ashraf A Kassim: *Performance evaluation of local descriptors and distance measures on benchmarks and first-person-view videos for face identification*. Neurocomputing, 184:107–116, 2016. 42, 51

[91] Shi, Dan Dan, Dan Chen, and Gui Jun Pan: *Characterization of network complexity by communicability sequence entropy and associated jensen-shannon divergence*. Physical Review E, 101(4):042305, 2020. 42, 51

[92] Ramdas, Aaditya, Nicolás García Trillos, and Marco Cuturi: *On wasserstein two-sample testing and related families of nonparametric tests*. Entropy, 19(2):47, 2017. 42, 51

[93] Union, IT: *Objective perceptual assessment of video quality: Full reference television. itu-t telecommunication standardization bureau*, 2004. 43, 47, 50

[94] Zhou, Qian Yi, Jaesik Park, and Vladlen Koltun: *Open3D: A modern library for 3D data processing*. arXiv:1801.09847, 2018. 45

[95] Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, *et al.*: *Scikit-learn: Machine learning in python*. Journal of machine learning research, 12(Oct):2825–2830, 2011. 45, 47

[96] Prettenhofer, Peter and Gilles Louppe: *Gradient boosted regression trees in scikit-learn*. 2014. 47

[97] Virtanen, Pauli, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1. 0 Contributors: *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*. Nature Methods, 17:261–272, 2020. 51