

# Image and Video Processing

## Laboratory 4

### Point Cloud subjective quality assessment

---

Prof. Touradj EBRAHIMI

**Reports submission – !!! read carefully !!!** Students will have to produce and submit a report two weeks after completion of each laboratory session. An electronic version of the report is submitted via Moodle's interface for uploading assignments. There will be a hard deadline for each submission. Late submissions will have to be justified and explained to the responsible assistant by email. A structure of the report is explained below. In addition to the report, the source code produced should also be submitted in electronic form. The report and the source code should be submitted in **ONE ZIP** file using Moodle platform.

#### Structure of the reports and m-files

- the name of the final ZIP file will be: `lab_number-of-lab_your-surname.zip`
- the final ZIP file will contain
  - final m-file called `run_lab_number-of-lab_your-surname.m`
  - all created m-files and functions which are necessary to obtain the results
  - all source images and data needed for successful running of the final m-file
  - an electronic version of your report in **PDF** format
- the structure of the final m-file
  - the final m-file will contain all necessary commands and functions, by running this m-file one must get all results<sup>1</sup>
  - submitted m-files will be commented
  - each figure will be properly titled
  - the final m-file will be divided into cells<sup>2</sup> according to the example bellow
- the parts of program codes should not be included in the final report
- description of the results should be a part of the final report
- don't forget to answer all the questions in your report
- check whether your final m-file can be launched without problems - only then submit your report

<sup>1</sup>of course it is upon you whether you want to include everything in the final m-file, or create different functions which you will call in the final m-file

<sup>2</sup>those who do not know how to use the cells in MATLAB to create the program code more transparent and easier to read, they should look at <http://www.mathworks.com/demos/matlab/developing-code-rapidly-with-cells-matlab-video-tutorial.html>

## An example of final m-file

```
% Lab (number of lab) - your name and date

%% Exercise (number of exercise) - "Name of exercise"

%your own program code with your comments
a = imread('picture.tif');
%all figures will be properly titled
figure('name','Name Of The Figure')
imshow(a,[]);

%% Exercise (number of exercise) - "Name of exercise"

%your own program code with your comments
[output1, output2, ..., outputN] = function_name (input1, input2,...,inputN);
```

# 1 Introduction

## 1.1 Point cloud imaging

The significant growth of Augmented Reality (AR) and Virtual Reality (VR) applications in recent years, has resulted in an increased interest for richer imaging modalities that can better simulate real-world scenes and provide immersive experiences. Point clouds denote a 3D content representation which is commonly preferred in such applications due to high efficiency and relatively low complexity in acquisition, storage and rendering of 3D models.

**Definition** A point cloud can be defined as a collection of points in 3D space which represents the surface of a model. Each point (i.e., sample of the surface) is defined by its position, given by a triplet of X, Y, and Z coordinates. Associated attributes, such as color values, normal vectors, reflectivity or curvature, can be used in conjunction with the coordinate data, in order to provide further information that more accurately reflects the underlying surface properties.

**Acquisition** A point cloud can be acquired using several approaches that can be classified as: (a) passive, and (b) active techniques. Passive approaches do not interfere with the model. For example, a 3D model can be reconstructed from several 2D images that capture the model using stereoscopic triangulation; that is, the depth of each point is determined from the captured images using simple geometry rules, given the position and orientation of the cameras. In case the camera's position and pose is unknown, they need to be estimated, which is a problem often referred to as SLAM (Simultaneous Localisation and Mapping). Such techniques are widely used in photogrammetry. Active approaches involve emission of light, or light patterns projection in the ultraviolet, visible, or infrared part of the spectrum. For example, time-of-flight (ToF) cameras estimate the depth of a pixel using a laser projector in conjunction with a camera. The projector illuminates an object, typically using infrared light, and the sensor detects its reflection. The distance between the object and the sensor is estimated based on the speed of light and the time delay between the emitted and detected light. LiDAR (i.e., “Light Detection And Ranging”), which can be seen as analogous to RADAR (“RAdio Detection And Ranging”) involving light transmission, is a similar and widely-used technology. It should be noted that the recent availability of such technologies in low cost depth sensors, has led to a significant amount of interest to be re-drawn for point cloud imaging.

**Data structures** Independently of the acquisition technique, the geometric structure of a captured or extracted point cloud is in principle irregular. This means that the coordinates of a point cloud are real numbers of any precision that can span at any range depending on the acquisition technology and the size of the scene. However, this creates difficulties in the manipulation of the model, given that a point

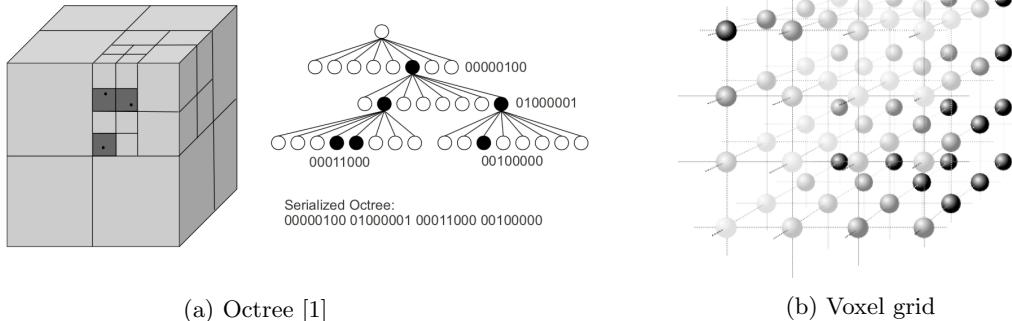


Figure 1: Popular data structures. Note that one way to obtain a voxel grid is by applying an octree.

cloud usually consists of a vast amount of points with coordinates in floating-point format. In order to deal with this overhead, suitable data structures have been proposed. The most popular are octrees and voxel grids. Octree structures are extensively exploited in point cloud representations and compression as they enable an efficient way to represent a point cloud model in binary format. In particular, a point cloud is enclosed by a minimum bounding box, and, in each level, each box is sub-divided into 8 smaller and equally sized boxes, as shown in Figure 1a. A point can be appended only in leaf nodes and all the points that are enclosed in a leaf node are collectively represented by the center of that node. Then, following a particular traversal in the tree, the point cloud can be represented as a bitstream where 1s indicate the non-empty leaf nodes. Voxelization is another commonly-used approach. A voxel  $v$  can be defined as a sample in a regularly spaced 3D grid, as shown in Figure 1b. It consists of a volumetric element of size 1, which is represented by the center of the voxel with coordinates  $(i, j, k) \in [0, 2^N - 1]^3$ , where  $N$  is the voxel bit depth. Voxelization can be defined as the process of mapping the coordinates of each point spanning in a continuous space,  $p \in R^3$ , to a discrete set of values,  $v \in Z_{>=0}^3$ . This is very similar to quantization. In both cases, duplicated entries in leaf nodes for octree and voxel centres for voxel grids are typically discarded. Information from additional attributes present in the original model is associated to the output data. In the case of color, for example, averaging of color values of duplicated entries is typically applied to obtain the color of an output data point. Note that the efficiency of these data structures come at the cost of information loss (e.g., point removal and regular displacement). In Figure 1 a typical octree and voxel grid data structure are illustrated. In this exercise, the point cloud models we provide are already voxelized.

**Compression** Since point clouds are very large data structures, effective compression algorithms are required for storage and transmission applications. These algorithms can be subdivided into two types: lossless, which compress data without any loss of information, and lossy, in which the encoding result on some loss of the information. For the second case, the main goal is to compress the data as much as possible without generating too much perceived distortion. Conventional compression algorithms can be classified as octree-based or projection-based. Algorithms from the first class represent the point clouds with an octree, and then prune the octree at a certain level of detail. Algorithms from the second class project the point cloud model into planes, generating images along with depth maps, that can be compressed with already existing image and video codecs. Besides, recent works have explored the possibility of using deep learning for point cloud compression. These works are mainly based on a family of neural network architectures called autoencoders. These schemes use convolutional neural networks that translate point clouds into vectors with smaller dimension on the encoder. Similarly, the decoder uses a symmetric architecture to generate a reconstructed point cloud from these vectors.

**Rendering** In a pioneering work, Levoy and Whitted [2] were the first to propose the use of points in computer graphics, stating that points in 3D should be viewed analogously to pixels in 2D. However, in practical terms, it is rather inefficient, to represent to use 3D points that are represented by single pixels in conventional screens. Thus, several rendering schemes have been proposed to address a trade-off between complexity and visual quality. In particular, a quite common approach is to reconstruct

a polygonal mesh from a point cloud. However, such reconstruction algorithms are costly in terms of time and computational complexity, especially for real-time applications. Thus, point-based rendering approaches are considered more suitable. The principle idea behind these techniques is to approximate the surface by assigning a primitive element (e.g., square, cube, disc or an ellipsoidal) to each point, which we call splat. In the simplest case, a point cloud is displayed by simply replacing points with splats. The size of each splat can be either fixed, or adaptive across a model. More sophisticated techniques take under consideration the normal vector of each point and its distances from nearest neighbours in order to orient and stretch each primitive element e.g., major and minor axis of an ellipsoidal, while also filling the local region accordingly. In these techniques, surface splatting and texture filtering are enabled; thus, although more complex, they result in high-quality surface approximations and water-tight models [3, 4]. For illustration reasons, in Figure 2, the point cloud is displayed using raw points, fixed-size splats, and the reconstructed polygonal mesh. In this exercise we will use a simple, yet efficient rendering solution using square splats with adaptive point size based on local neighborhoods.

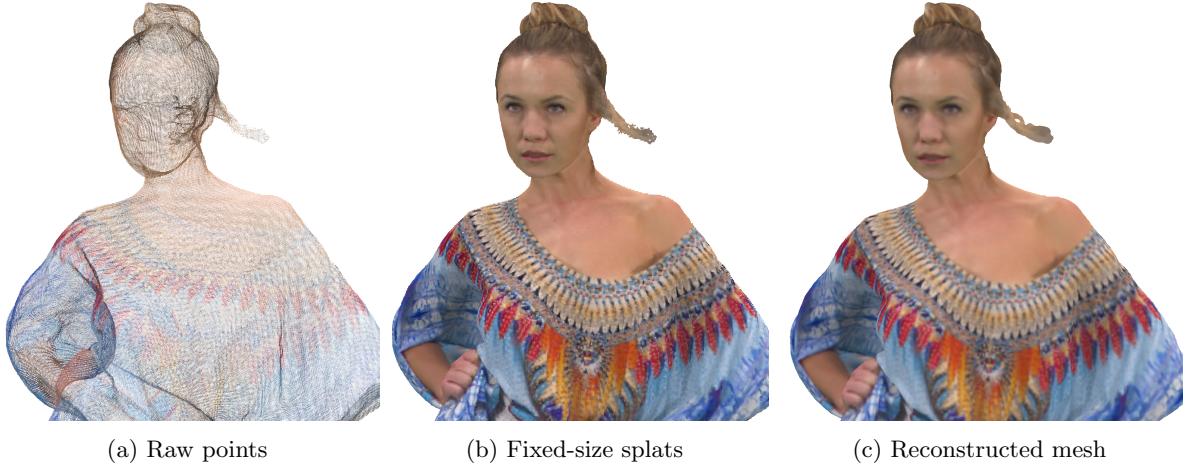


Figure 2: Point cloud using different rendering methods.

## 1.2 Subjective quality assessment

Measurement of perceived quality plays a fundamental role in the context of multimedia services and applications. Quality evaluation is needed in order to benchmark image, video, and audio processing systems and algorithms, to test end-devices performance, and to compare and optimize algorithms and their parameters setting. As human subjects usually act as end-users of digital content, subjective tests are performed, where a group of people is asked to rate the quality or the level of impairment of the multimedia material, or to submit their preference between two different versions of a multimedia content. For example,

- to assess the quality of a content, the following five-level quality scale is commonly used: 5 - *Excellent*, 4 - *Good*, 3 - *Fair*, 2 - *Poor*, 1 - *Bad*
- to evaluate the level of impairment of a content with respect to its reference version, the following five-level impairment scale is commonly used: 5 - *Imperceptible*, 4 - *Perceptible, but not annoying*, 3 - *Slightly annoying*, 2 - *Annoying*, 1 - *Very annoying*
- to identify preferences between different versions of the same content, pair-wise comparisons are typically used. In some cases, the subjects are forced to choose one option over the other, while in other cases, the subjects are allowed to choose that the visual quality of both versions is the same (i.e., tie).

The evaluation methodology and rating scale is determined depending on the scope of the experiment, while the collected human opinions denote the ground truth related to the quality level of the stimuli under assessment.

Subjective evaluations are usually performed by a limited group of test subjects. The participants should be a representative sample of the entire population of end-users for the application under analysis. The subjective results are statistically analyzed in order to understand whether it is possible to draw general conclusions which are valid for the entire population. Subjective quality assessment experiments have to be carried out with scientific rigour in order to provide valid and reliable results.

### 1.2.1 Test methodology

Several methodologies have been proposed by international standardization bodies for the subjective quality evaluation of still and moving images. For more details, please refer to ITU-R Recommendation BT.500-13 [5] and Recommendation ITU-T P.910 [6]. Within this lab session, we will use the Absolute Category Rating with Hidden Reference (ACR-HR). The subjects will be able to inspect the stimuli passively through a video of a rotating point cloud model and rate them in a scale from 1 to 5. The subjects will evaluate distorted point cloud models generated by different compression schemes and different compression levels. Along with these stimuli, there will also be one hidden reference for each content. The evaluation will be carried out on a crowdsourcing platform, and thus each subject will access it through a different device and a different screen.

### 1.2.2 Processing of subjective data

During the subjective assessment session, the scores given by all the subjects for each one of the point cloud models will be stored. The analysis of the collected data starts by computing the DMOS (Differential Mean Opinion Score) and the CI (Confidence Interval). In order to compute the DMOS, the first step is to calculate the differential viewer scores (DVs) for each distorted content. This is done by applying the formula on Equation 1, where  $V$  is the ACR score given by the subject for a distorted model and  $V(REF)$  is the ACR score given to the reference model of that same content.

$$DV = V - V(REF) + 5 \quad (1)$$

After computing the DV for each distorted model, the DMOS is obtained through Equation 2.

$$DMOS_j = \frac{\sum_{i=1}^N DV_{ij}}{N} \quad (2)$$

where  $N$  is the number of subjects and  $DV_{ij}$  is the differential score by subject  $i$  for the stimulus  $j$ .

Together with the MOS, the confidence interval (CI) of the estimated mean should also be computed. The CI provides information upon the relationship between the estimated mean values based on a sample of the population (i.e., the limited number of subjects who took part in the experiment) and the true mean values of the entire population. Due to the small number of subjects (usually around 15) the  $100 \times (1 - \alpha)\%$  CI is computed using the Student's  $t$ -distribution, as follows:

$$CI_j = t(1 - \alpha/2, N - 1) \cdot \frac{\sigma_j}{\sqrt{N}} \quad (3)$$

where  $t(1 - \alpha/2, N - 1)$  is the  $t$ -value corresponding to a two-tailed Student's  $t$ -distribution with  $N - 1$  degrees of freedom and a desired significance level  $\alpha$  (equal to 1-degree of confidence).  $N$  corresponds to the number of subjects, and  $\sigma_j$  is the standard deviation of the differential scores assigned to the stimulus  $j$ . The interpretation of a confidence interval is that if the same test is repeated for a large number of times, using each time a random sample of the population, and a confidence interval is constructed every time, then  $100 \times (1 - \alpha)\%$  of these intervals will contain the true value. **Usually, for the analysis of subjective results, the confidence intervals are computed for  $\alpha$  equal to 0.05, which corresponds to a degree of confidence of 95%.**

### 1.2.3 Outlier detection

In the majority of the experiments, outlier detection and removal is performed in order to exclude subjects whose ratings deviates drastically from the rest of the scores. In this exercise, outlier subjects will be removed according to the procedures defined in ITU-T Recommendation P.913 [7]. This algorithm is based on the correlation coefficient between the raw scores of one subject and the MOS over all subjects. For this reason, the linear Pearson correlation coefficient (PLCC) given in Equation 4 is used, where  $x$  and  $y$  are arrays of data and  $n$  is the total number of points on  $x$  and  $y$ .

$$\text{PLCC}(x, y) = \frac{n \cdot \sum_{i=1}^n x_i \cdot y_i - \sum_{i=1}^n x_i \cdot \sum_{i=1}^n y_i}{\sqrt{n \cdot \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \cdot \sqrt{n \cdot \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2}} \quad (4)$$

The PLCC can be computed on two ways: for individual stimuli (point cloud models) or for all the models for the same content. When computing for individual stimuli, the coefficient  $r_1$  is computed using Equation 5.

$$r_1(x, y) = \text{PLCC}(x, y) \quad (5)$$

On Equation 5:

- $x_i$  is the MOS of all subjects for a given stimulus  $i$
- $y_i$  is the individual score of one subject for the stimulus  $i$
- $n$  is the total number of stimuli

Moreover, the same metric (PLCC) is also computed for all the stimuli for one particular content, resulting in Equation 6

$$r_2(x, y) = \text{PLCC}(x, y) \quad (6)$$

On Equation 6:

- $x_i$  is the MOS of all subjects for all the stimuli for a given content  $i$
- $y_i$  is the MOS of one subject for all the stimuli for a given content  $i$
- $n$  is the total number of contents

A subject is considered as an outlier when both  $r_1$  and  $r_2$  fall bellow some pre-determined thresholds. In this exercise, we will use as rejection rules when  $r_1 < 0.75$  and  $r_2 < 0.8$ . Note that subjects should be discarded one at a time, beginning with the worst outlier (i.e., by averaging the amount that the two thresholds are exceeded) and then recalculating  $r_1$  and  $r_2$ .

Note: the MOS stands for Mean Opinion Score and can be calculated by the average of all the scores given by each subject to a given stimulus. It differs from the DMOS because it doesn't take into account the score given to the reference model relative to that stimulus. The MOS can be given by Equation 7, being  $m_{ij}$  the score given by subject  $i$  to a stimulus  $j$ .

$$\text{MOS}_j = \frac{\sum_{i=1}^N m_{ij}}{N} \quad (7)$$

### 1.3 Objective quality assessment

Although highly informative and reliable, subjective experiments are difficult to design and time-consuming. Furthermore, they cannot be applied when real-time in-service quality evaluation is needed. In order to reduce the effort of subjective testing and overcome its limitations, algorithms, i.e., objective quality metrics, have been developed in literature to estimate the outcome of the subjective tests. These quality metrics aim at automatically and reliably predicting the quality of the multimedia content, as perceived by the human end-user.

Objective quality metrics available in literature can be divided in three different categories according to the availability of the original, i.e., reference, signal: full reference (FR) metrics, when both original and processed signals are available; reduced reference (RR) metrics, when besides the processed signal, a description of the original signal and some parameters are available; no-reference (NR) metrics, when only the processed signal is available.

In objective quality assessment of point clouds, FR metrics are used. They can be distinguished in two main classes: (a) point-based and (b) projection-based metrics.

#### 1.3.1 Point-based objective quality metrics

Current point-based approaches can assess either geometry- or color-only distortions. Objective quality metrics for geometric-only distortions can be classified as point-to-point (po2point) point-to-plane (po2plane) [8] and plane-to-plane (pl2plane) [9] metrics. For color-only distortions, formulas from 2D imaging approaches are used.

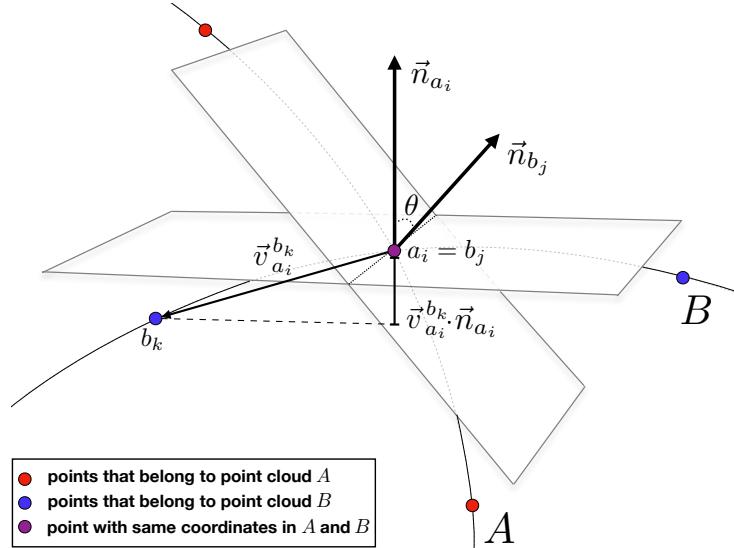


Figure 3: Point cloud objective quality metrics.

**Point-to-point metrics** The point-to-point metrics depend on geometric distances of associated points between the reference and the model under evaluation. In particular, following the notations of Figure 3, for each point  $b_k$  of the model under evaluation  $B$ , its nearest neighbor  $a_i$  from the reference point cloud  $A$  is determined. Then, to obtain an individual score, the Euclidean distance between them,  $e(b_k, a_i)$ , is computed based on Equation 8.

$$e(b_k, a_i) = \|\vec{v}_{a_i}^{b_k}\|_2 \quad (8)$$

The error value assigned to  $b_k$  reflects its geometric displacement from the reference position.

**Point-to-plane metrics** The point-to-plane metrics [8] depend on the projected error across the normal vector of an associated reference point. In particular, for each point  $b_k$  of the model under evaluation  $B$ , its nearest neighbor  $a_i$  from the reference point cloud  $A$  is determined. Then, to obtain an individual score, the projected error  $\hat{e}(b_k, a_i)$  of  $b_k$  across the normal vector  $\vec{n}_{a_i}$  of the reference point  $a_i$ , is computed based on Equation 9.

$$\hat{e}(b_k, a_i) = \vec{v}_{a_i}^{b_k} \cdot \vec{n}_{a_i} \quad (9)$$

The error value assigned to  $b_k$  indicates its deviation from the linearly approximated reference surface. Note that for the calculation of this metric, the normal vectors of the reference model should be given or estimated.

**Plane-to-plane metrics** The plane-to-plane metrics [9] depend on the angular similarity of tangent planes that correspond to associated points between the reference and the model under evaluation. In particular, for each point  $b_j$  of the model under evaluation  $B$ , its nearest neighbor  $a_i$  from the reference point cloud  $A$  is determined. Then, to obtain an individual score, the angle  $\hat{\theta}$  between the associated normal vectors  $\vec{n}_{a_i}$  and  $\vec{n}_{b_j}$  is firstly computed, and the angular similarity is obtained by keeping the minimum out of the two angles that are formed by the intersecting tangent planes  $\theta = \min\{\hat{\theta}, \pi - \hat{\theta}\}$ , as shown in Figure 3. The angular similarity bounded in the range  $[0, 1]$  is computed based on Equation 10.

$$\tilde{e}(b_j, a_i) = 1 - \frac{2\theta}{\pi} \quad (10)$$

The error value assigned to  $b_j$  indicates the similarity of the linear local surface approximations between the two models at that point. Note that for the calculation of this metric, the normal vectors of both the reference and the model under evaluation should be given or estimated. If not given, this dependency makes the algorithm being affected by the selected normal estimation algorithm and its configuration, although no specific methodology is imposed as part of it.

**Color metrics** The color-only metrics use formulas from 2D imaging, which are applied between pairs of associated points instead of matching pixels, that belong to the reference and the model under evaluation. The most widely used metric is the well-known PSNR, whose computation is adjusted for point clouds as described below. In particular, for each point  $b_j$  of the model under evaluation  $B$ , its nearest neighbor  $a_i$  from the reference point cloud  $A$  is determined. Then, to obtain an individual score, the absolute difference between the two points is computed for each color channel. These calculation can be either in RGB, or YUV color space, using potentially the ITU-R Recommendation BT.709-5 [10] for color conversion. In Equation 11, the error between a pair of associated points is indicatively given, considering only the red (R) color channel.

$$\hat{e}(b_k^R, a_i^R) = |b_k^R - a_i^R| \quad (11)$$

**Total error values** The metrics that were mentioned above provide individual error values for each point of the model under evaluation. To compute a total distortion score, a number of different pooling methods has been proposed, with the most common being the Mean Squared Error (MSE) and the Hausdorff distance<sup>3</sup>, given in Equations 12 and 13, respectively

$$d_{\text{MSE}}^{B,A} = \frac{1}{N_B} \cdot \sum_{k=1}^{N_B} \mathcal{E}(b_k, a_{i_k})^2 \quad (12)$$

$$d_{\text{HAU}}^{B,A} = \max_{\forall k \in B} \mathcal{E}(b_k, a_{i_k}) \quad (13)$$

where  $\mathcal{E}(\bullet)$  should be replaced by the error metrics defined above.

- For point-to-point metrics, the MSE and Hausdorff distance will be used in this exercise.

---

<sup>3</sup>The Hausdorff distance is defined as the maximum of the distances of each point in one set from its nearest neighbor in the other set.

- For point-to-plane metrics, the MSE and Hausdorff distance will be used in this exercise.
- For plane-to-plane metric, the MSE distance will be used in this exercise.
- For color metrics, the MSE will be used in this exercise. In particular, two versions of the PSNR calculation will be tested for the RGB colorspace. In the first version, the MSE will be computed across all three channels, with the PSNR given by Equation 14.

$$\text{PSNR}_{\text{RGB}} = 10 \cdot \log_{10} \frac{255^2}{\text{MSE}} \quad (14)$$

In the second version, the average of PSNR values from the three channels will be computed, as given by Equation 15.

$$\text{PSNR}_{\text{RGB}} = (\text{PSNR}_R + \text{PSNR}_G + \text{PSNR}_B) / 3 \quad (15)$$

Regarding the computations in the YUV colorspace, different weights will be considered for the luminance and the chrominance channels, as given in Equation 16.

$$\text{PSNR}_{\text{YUV}} = (6 \cdot \text{PSNR}_Y + \text{PSNR}_U + \text{PSNR}_V) / 8 \quad (16)$$

Finally, to obtain a total error measurement, the *symmetric error* is used. This is obtained by setting both models as reference, and estimating the corresponding error values using either the MSE or the Hausdorff distance. Then, the maximum error value is kept. **Note that for geometric distances, the higher the number the higher the error value, while for similarity distances, the higher the number, the lower the error.**

## 1.4 Comparison of objective and subjective quality scores

The ground truth data gathered through subjective tests is used to benchmark the performance of objective metrics. Usually two attributes are considered in order to compare the prediction performance of the different metrics with respect to subjective ratings:

- **Accuracy** is the ability of a metrics to predict subjective ratings with the minimum average error. It is measured by means of the Root-Mean-Square Error (RMSE) and Pearson correlation coefficient (PLCC), which quantifies the linear correlation between the MOS values and the predicted values. The Root-Mean-Square Error is estimated using the Equation 17:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (x_i - y_i)^2}{n}} \quad (17)$$

The Pearson correlation coefficient (PLCC) is given by Equation 4. In both equations,  $x$  and  $y$  are two arrays to be compared, with  $n$  the number of elements of both arrays. The value of PLCC ranges in the  $[-1, 1]$  interval, where a value close to 1 (-1) indicates the strongest positive (negative) correlation.

- **Monotonicity** measures if an increase (decrease) in one variable is associated with an increase (decrease) in the other variable, independently of the magnitude of the increase (decrease). It is measured by means of the Spearman correlation coefficient (SROCC), which quantifies the monotonicity of the mapping, e.g., how well an arbitrary monotonic function could describe the relationship between the MOS values and the predicted values. It is defined as:

$$\text{SROCC} = 1 - \frac{6 \cdot \sum_{i=1}^n (\mathcal{R}(x_i) - \mathcal{R}(y_i))^2}{n \cdot (n^2 - 1)}, \quad (18)$$

In equation 18,  $x$  and  $y$  are the two datasets of  $n$  values each, to be compared, and  $\mathcal{R}(\bullet)$  denotes a ranking relation (sorting) applied to the argument.

## 1.5 Comparison of subjective quality scores from different experiments

Subjective tests are often conducted in a laboratory environment, where the environmental factors, such as viewing conditions, light conditions, display conditions, etc. are controlled as much as possible to ensure the reproducibility and reliability of the results. The modification of environmental conditions, or the test methodologies used in the experiments, can have a severe impact on the results.

To determine whether the results obtained from subjective experiments performed using different test methodologies are similar, the correlation between the sets of results is measured using the PLCC, SROCC and RMSE indexes as computed in Equations 4, 18 and 17, respectively.

## 1.6 Privacy and data protection

Subjective evaluations produce various private data needed for further analysis (for example data about the sample of participating subjects). Such data are strictly protected, in some countries, such as Switzerland, even by law. Prior the subjective tests, a consent form must be distributed to subjects. It contains basic information about the subjective evaluation and list all relevant details (subject's rights, health risks, etc., ... ) regarding the tests. By signing such form, participants formally agree to perform the tests. Consent forms are also kept aside, and cannot be mutually linked, to raw test results. An example of such form is attached at the end of this document and will be given to you prior the subjective test and you will be asked to sign it. Finally, it should be mentioned that the IDs that are assigned to you in order to perform the experiment are dummy; this means, that upon the investigators receive your data, a random generator will be used to assign your dummy to new IDs, in order to prevent subject's identification and meet data protection requirements.

# 2 Quality assessment of point cloud compression

In this exercise, the task consists in performing a subjective evaluation experiment on a crowdsourcing platform. This exercise will help us to collect real raw scores for the point clouds under assessment. In particular, after you finish the experiment, your data will be logged. This data will be analyzed by you in the following exercises. After you finish the experiment, your data will be automatically logged on the server.

## 2.1 Experiment

In this step, you conduct the experiment.

1. Start the EPFL VPN to have access to the EPFL intranet
2. In a web browser, access the following link: [http://grebsrv7.epfl.ch/{your\\_sciper\\_number}/1](http://grebsrv7.epfl.ch/{your_sciper_number}/1)
3. Enter in fullscreen mode
4. In the screen that appears, type your age and gender
5. Read the instructions that appear and click Next
6. At this point the training session will start:
  - (a) In the screen in front of you, you can see a video of a rotating point cloud model. The task is to assess the visual quality of the model. Ensure that you take under consideration distortions in both the geometry and the color of the models. You can watch the video of the rotating model as many times as you want, but you have to watch the full video at least once. You can spend as much time as you need in order to be sure of your judgement, but you shouldn't overdo it, as there is no time limitation. Once you are convinced of the score you want to give to the model, press Next and move to the next model.

- (b) In the first example of the training session, you can see a model with excellent quality, that should be rated as Excellent.
- (c) In the second example of the training session, you can see a model with some visible color distortions. It should be rated as Fair.
- (d) In the third example of the training session, you can see a model with very bad quality, with deformed shape and strong color distortions. It should be rated as Bad.
- (e) After you finish the training session, you can start the experiment whenever you are ready. **Please make sure that once you start the test, you remain focused on identifying impairments on the models under evaluation without being distracted from external sources until the test is finished.**

## 2.2 Data description

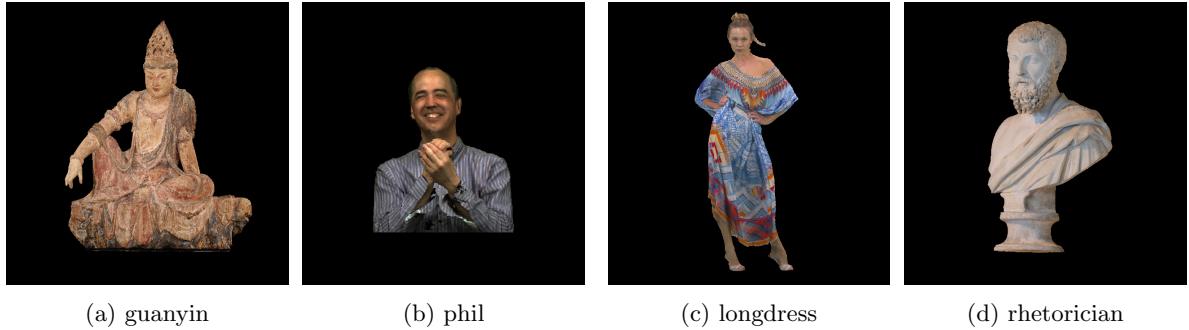


Figure 4: Contents used in this study.

The goal of this exercise is to assess the performance of 2 different point cloud encoders using both subjective and objective quality metrics. The pcc-geo-color codec was proposed in [11] and is a deep learning based codec which encodes both color and geometry using convolutional neural networks. In the other hand, CWI-PCL, proposed in [12], was used as MPEG anchor and takes advantage of an octree decomposition to compress the models.

For this purpose 4 contents have been selected, shown in Figure 4, and are encoded at 4 different degradation levels, namely, L1-L4.

Download `data.zip` from Moodle “Lab 4 - Data”. By opening the `data.mat` in MATLAB you will find all the necessary information regarding the models, codecs, degradations, number of points, and bitrate, that correspond to each filename, in table `data`. Note that the bitrate is computed as the ratio of the total number of bits divided by the number of points in the *reference* model. Moreover, you will find the `subjective_scores_dummy.csv` file, which correspond to dummy scores generated for the purposes of the lab. In this matrix, each column corresponds to the ratings of one subject and each row corresponds to a model. You’ll also find the `.ply` point cloud files relative to the reference and distorted models. Finally, you will find scripts and matrices in order to convert RGB to YUV colorspace and compute the angular similarity.

- Useful MATLAB commands: `pcread`, `pcshow`

## 2.3 Subjective quality assessment

1. Write a function that implements the algorithm described in ITU-T Recommendation P.913 in order to detect the outliers from the subjective ratings. Use this function to remove any outlier subjects that might exist in the data.
2. Write a function to compute the DMOS values and corresponding 95% CIs for each stimulus (model, codec, degradation).

3. Plot the DMOS values together with CIs against bitrates per content (i.e., on the same graph, plot 2 “DMOS vs bitrate” curves corresponding to the 2 codecs; plot one graph for each content).
4. Comment upon and justify the obtained results:
  - Do you observe similar trends between different contents?
  - Which codec performs better in terms of perceived quality?
  - Subjective scores are always improving with higher bitrates?

Note that after the subjective experiments are done (Wednesday 11/11 in the afternoon), you will be able to download the real subjective scores from Moodle “Lab 4 - Real subjective scores”. Then, you will replace these random data with the correct scores in order to produce the correct results!

- Useful MATLAB commands: `mean`, `std`, `icdf`, `errorbar`

## 2.4 Objective quality assessment

1. Write a function to compute the symmetric point-to-point metric using MSE and Hausdorff distance.
2. Estimate the normals of the models with  $k = 128$  nearest neighbors and compute the symmetric plane-to-plane metric with MSE using the given `angularSimilarity` function.  
Note: this normal computation for all the point cloud models should take several minutes. It is advised to try first with smaller value for  $k$ , and after computing the normals with  $k = 128$ , save them with the function `pcwrite`.
3. Write a function to compute the symmetric color-PSNR in the RGB colorspace using Equations 14 and 15 and in YUV colorspace using Equation 16.
4. Plot objective scores from each metric (i.e., given and computed) per content (i.e., on the same graph, plot 2 “Objective scores vs bitrate” curves corresponding to the 2 codecs; plot one graph for each combination of objective metric and content).
5. Comment upon and justify the obtained results:
  - Do you observe similar trends between different contents for the same metric?
  - Do you observe similar trends between different metrics for the same content?
  - Which codec performs better in terms of objective quality?

- Useful MATLAB commands: `pcread`, `pcnormals`, `knnsearch`, `psnr`, `ssim`

## 2.5 Benchmarking of objective quality metrics

The goal of this exercise is to benchmark the state-of-the-art objective quality metrics described in Section 1.3 using the subjective scores as ground truth. For this purpose, you will use the subjective scores and the objective measurements that are already computed.

1. Plot the DMOS values together with the CIs for each metric for all models (i.e., on the same graph, plot the “DMOS vs Objective metric” for all models; plot one graph for each metric).
2. Compute the Pearson and Spearman correlation coefficients as well as the RMSE, between the objective and subjective scores. Report these values in a table.
3. For each metric, fit the objective scores to the subjective scores, using linear and cubic fitting.
4. Repeat steps 1-2 using the fitted objective scores.
5. Comment upon and justify the obtained results:

- Which metric performs better in terms of correlation with the subjective results? Is there any particular category of metrics that is more efficient (e.g., color-only, geometry-only)?
  - What are the limitations of the metrics?
- Useful MATLAB commands: corr, polyfit

## References

- [1] J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach, “Real-time compression of point cloud streams,” in *2012 IEEE International Conference on Robotics and Automation*, pp. 778–785.
- [2] M. Levoy and T. Whitted, *The use of points as a display primitive*. University of North Carolina, Department of Computer Science, 1985.
- [3] H. Pfister, M. Zwicker, J. Van Baar, and M. Gross, “Surfels: Surface elements as rendering primitives,” in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 335–342.
- [4] M. Zwicker, H. Pfister, J. van Baar, and M. Gross, “Surface splatting,” in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 2001, pp. 371–378.
- [5] ITU-R Recommendation BT.500-13, *Methodology for the subjective assessment of the quality of television pictures*, International Telecommunications Union Std., January 2012.
- [6] ITU-T Recommendation P.910, *Subjective video quality assessment methods for multimedia applications*, International Telecommunications Union Std., April 2008.
- [7] ITU-T Recommendation P.913, *Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment*, ITU-T Recommendation P.913 Std., March 2016.
- [8] D. Tian, H. Ochiaimizu, C. Feng, R. Cohen, and A. Vetro, “Geometric distortion metrics for point cloud compression,” in *2017 IEEE International Conference on Image Processing (ICIP)*, Sep. 2017, pp. 3460–3464.
- [9] E. Alexiou and T. Ebrahimi, “Point cloud quality assessment metric based on angular similarity,” in *2018 IEEE International Conference on Multimedia and Expo (ICME)*, July 2018, pp. 1–6.
- [10] ITU-R Recommendation BT.709-5, *Parameter values for the HDTV standards for production and international programme exchange*, International Telecommunications Union Std., April 2002.
- [11] E. Alexiou, K. Tung, and T. Ebrahimi, “Towards neural network approaches for point cloud compression,” 08 2020, p. 4.
- [12] R. Mekuria, K. Blom, and P. S. Cesar Garcia, “Design, implementation and evaluation of a point cloud codec for tele-immersive video,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 828–842, Apr. 2017.