Image and Video Processing
Laboratory 4
Subjective and Objective 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[1]
  - submitted m-files will be commented
  - each figure will be properly titled
  - the final m-file will be divided into cells[2] 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

[1] 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

[2] 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

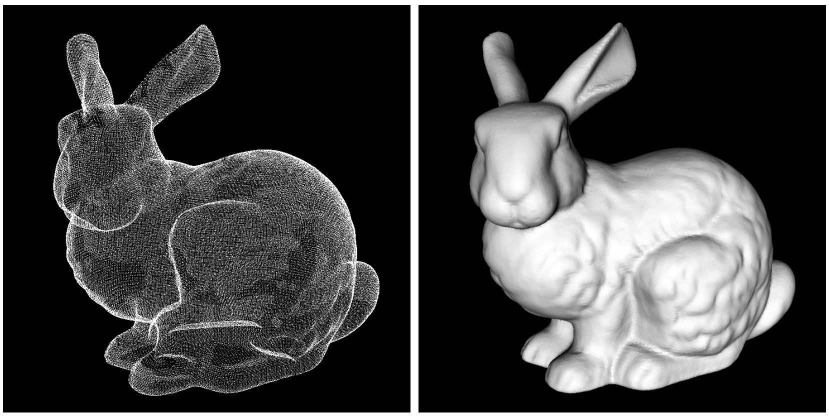
```
% Lab (number of lab) - your name and date

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

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

1 Introduction

1.1 Point clouds as an immersive imaging solution

1.1.1 Point clouds and polygon meshes

The significant growth of Augmented Reality (AR), Virtual Reality (VR), medical imaging and 3D printing applications, in recent years, resulted in an increased interest for advanced 3D representations. In this environment a common and practical way for storage and rendering of 3D models in such applications is by using point clouds. A point cloud could be defined as a collection of 3D points in space representing the external surface of an object. Each sample is defined by its position, which is obtained by the measured or reconstructed X, Y, and Z coordinates. Associated features, such as point's color, intensity, normal, are also often used in conjunction with the coordinate data to provide further information. The usage of this representation has been also remarkably increased by the recent availability of low-cost consumer market devices that provide depth estimation (e.g., Kinect, Intel RealSense) and the adoption of such technologies in handheld devices, such as phones and laptops (e.g., iPhone 10, HP Spectre x2).



(a) Point cloud (b) Polygon mesh

Figure 1: Different representations of the same 3D object (i.e., bunny from Stanford 3D Repository)

Although polygon mesh has been the prevailing form of representation for 3D models, point cloud is expected to become important in representation of 3D imaging, especially, in AR and VR real-time applications. In particular, to form a polygon mesh, in addition to point coordinates, connectivity information (i.e., mesh topology and polygons/faces) is also required. This extra information needs to be stored, encoded, delivered and decoded in a typical communication scheme, increasing the costs in terms of time and computational complexity. Polygon meshes also require sophisticated reconstruction techniques when updating the content for every additional collection of points. Using point clouds, the captured objects could be updated just by rendering the additional points.

On the other hand, using points in 3D space to represent an object leads to ambiguity regarding the perception of the underlying model, especially, when complex objects are displayed in the absence of color. The use of points in computer graphics was initially proposed back in 1985 in a pioneering work by Levoy and Whitted [1], where it was stated that points in 3D should be viewed analogously to pixels in 2D. However, it is almost impossible to visualize water-tight surfaces by using just points. This is because dense point clouds (a) cannot be acquired easily (i.e., highly expensive equipment is needed, or the same object should be scanned multiple times), and (b) they are highly inefficient due to the huge amount of data that needs to be stored, delivered and rendered. Even in the case of a highly dense point cloud, after decreasing the distance between the viewer and the object enough, "holes" will be observed. This leads to another severe issue; that is visual quality is hard to be defined for this representation.

For this purpose, different rendering approaches have been proposed. The principle idea behind the majority of these techniques is to assign a disc or an ellipsoidal on every point, whose radius or major and minor axis are estimated based on the normal of the point and its distances from nearest neighbours. The result is a content that can be perceived as water-tight [2, 3]. In the near future, GPUs are expected to provide rendering algorithms which will reconstruct on-the-fly a polygonal mesh given a point cloud as an input. In other terms, the storage, encoding, transmission and decoding of the connectivity information will be avoided, and by just feeding the captured or delivered point cloud in the GPU, the corresponding polygon mesh will be rendered. For this reason, in this lab session, we will use a quite popular surface reconstruction technique as a rendering algorithm, namely Screened Poisson surface reconstruction [4].

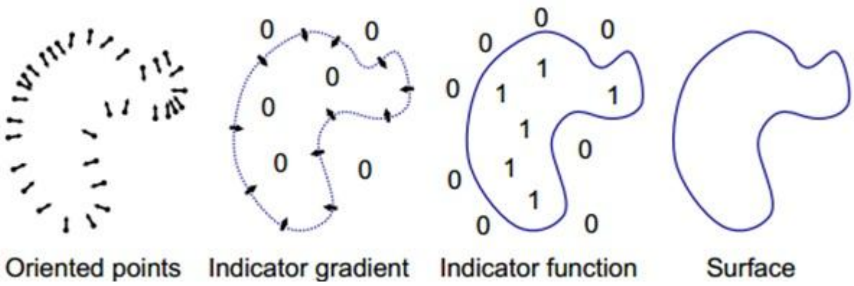


Figure 2: From oriented points to surface

Screened Poisson surface reconstruction aims to compute an indicator function (a function that encloses an area, see Figure 2) and then obtain the reconstructed surface by extracting an appropriate iso-surface. This is done, by assuming that the normal associated with a point (or, an oriented point) of the content is a sample of the gradient of the indicator function. Thus, the problem reduces in finding a function whose gradient best approximates the normals of the points, considering also constraints related to the positions of the points. For more details, please refer to [4]. What is important for our use case, is that this algorithm provides various parameters that can be set manually, which affect the construction of the output mesh. For instance, by increasing the neighborhood over which the indicator gradient is estimated, leads to lower level of details on the estimated surface. This means less points and polygons (which introduce less complexity during the rendering), but also lower visual quality. Please note that from now on, when it is mentioned that different rendering configurations are used, the use of different polygon meshes with different number of points and polygons/faces is implied.

1.1.2 Data formats

Various formats have been used for both point clouds and polygon meshes. To name a few, .pcd, .ply, .stl are used for the first and .vtk, .ply, .obj for the latter type of contents. In this exercise, we will use

.pcd[3] and .ply[4], respectively. In Figures 3 and 4, you may observe a typical structure of each adopted file format, along with comments that briefly explain the semantics.

```
# .PCD v.7 - Point Cloud Data file format
VERSION .7                                  { the PCD file version }
FIELDS x y z rgb                            { the name of each dimension/field that a point can have }
SIZE 4 4 4 4                                { define the size of each dimension in bytes, 4 bytes in this case }
TYPE F F F F                                { define the type of each dimension as a char, F represents floats }
COUNT 1 1 1 1                               { how many elements does each dimension have }
WIDTH 8                                     { the width of the point cloud dataset in the number of points }
HEIGHT 1                                    { the width of the point cloud dataset in the number of points }
VIEWPOINT 0 0 0 1 0 0 0                     { specifies an acquisition viewpoint }
POINTS 8                                    { the total number of points in the cloud }
DATA ascii                                  { the data type that the point cloud data is stored in }
0.93773 0.33763 0 4.2108e+06                { start of list of points }
0.90805 0.35641 0 4.2108e+06
0.81915 0.32 0 4.2108e+06
0.97192 0.278 0 4.2108e+06
0.944 0.29474 0 4.2108e+06
0.98111 0.24247 0 4.2108e+06
0.93655 0.26143 0 4.2108e+06
0.91631 0.27442 0 4.2108e+06
```

Figure 3: PCD format

```
ply
format ascii 1.0           { ascii/binary, format version number }
comment made by Greg Turk  { comments keyword specified, like all lines }
comment this file is a cube
element vertex 8           { define "vertex" element, 8 of them in file }
property float x           { vertex contains float "x" coordinate }
property float y           { y coordinate is also a vertex property }
property float z           { z coordinate, too }
element face 6             { there are 6 "face" elements in the file }
property list uchar int vertex_index { "vertex_indices" is a list of ints }
end_header                 { delimits the end of the header }
0 0 0                      { start of vertex list }
0 0 1
0 1 1
0 1 0
1 0 0
1 0 1
1 1 1
1 1 0
4 0 1 2 3                  { start of face list }
4 7 6 5 4
4 0 4 5 1
4 1 5 6 2
4 2 6 7 3
4 3 7 4 0
```

Figure 4: PLY format

Such files can be stored either in ASCII or in binary format. The provided contents have been stored in ASCII, so they can be opened using for instance WordPad, in order to verify the aforementioned structure. In addition, you can notice the number of points and polygons (faces) of every content.

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 of the multimedia material, the overall multimedia experience, or the level of impairment by using the same rating scale.

A commonly used scale to evaluate quality is the five-level quality scale:
5 - *Excellent*, 4 - *Good*, 3 - *Fair*, 2 - *Poor*, 1 - *Bad*

A commonly used scale to evaluate level of impairment is the five-level impairment scale:
5 - *Imperceptible*, 4 - *Perceptible, but not annoying*, 3 - *Slightly annoying*, 2 - *Annoying*, 1 - *Very annoying*

The limited group of subjects should be a representative sample of the entire population of end-users for the application under analysis. The subjective results are then 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 rigor in order to provide valid and reliable results.

[3]http://pointclouds.org/documentation/tutorials/pcd_file_format.php
[4]http://paulbourke.net/dataformats/ply/

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, you will follow Single Stimulus with Impairment Scale (SSIS) and simultaneous Double Stimulus with Impairment Scale (DSIS), both using Hidden Reference. We should emphasize that in Single Stimulus methods, the Absolute Category Scale (ACR) is mainly used. In this case though, since you will compare the subjective scores from two different test methods, it is important to use the same rating scaling in both experiments.

1.2.2 Processing of subjective data

In order to interpret the results of subjective quality assessment experiments, the scores assigned by all the observers taking part to a test are averaged in order to obtain the mean opinion score (MOS) for each stimulus:

$$\mathrm{MOS}_j = \frac{\sum_{i=1}^{N} m_{ij}}{N} \tag{1}$$

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

Apart from the MOS, another quantity is also computed, which is the confidence interval (CI) of the estimated mean. 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:

$$\mathrm{CI}_j = t(1-\alpha/2, N) \cdot \frac{\sigma_j}{\sqrt{N}} \tag{2}$$

where $t(1-\alpha/2, N)$ is the t-value corresponding to a two-tailed Student's t-distribution with $N-1$ degrees of freedom and a desired significance level α (equal to 1-degree of confidence). N corresponds to the number of subjects, and σ_j is the standard deviation of the 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 α 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-R Recommendation BT.500-13 [5]. Initially, it is determined whether the distribution of scores for a test sequence is normal or not. Specifically, for each test sequence, if the kurtosis coefficient of the subject scores is between 2 and 4, the distribution can be considered as normal. Then, to reject a subject, a confidence interval is defined with the following procedure: if scores are distributed normally, for each score larger than $2 \cdot \sigma$ from the mean of the scores (upper limit) of a test sequence i, a counter P_i is incremented. For each score smaller than $2 \cdot \sigma$ from the mean of the scores (lower limit) of the test sequence i, a counter Q_i is incremented. In case of non-normal distributions, the upper and lower limits are set as $\sqrt{20} \cdot \sigma$ from the mean of the scores of a test sequence. Assuming a total of N_s number of stimuli, the scores of a subject are removed if the following conditions are met:

$$\frac{\sum_{i=1}^{N_s}(P_i + Q_i)}{N_s} > 0.05 \text{ and } \left| \frac{\sum_{i=1}^{N_s}(P_i - Q_i)}{\sum_{i=1}^{N_s}(P_i + Q_i)} \right| < 0.3 \tag{3}$$

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 metrics, have been developed in literature to estimate the outcome of the subjective tests. These quality metrics try to automatically and reliably predict 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 and are based on the relative geometrical differences between the points of the processed and the reference contents. The state-of-the-art objective metrics for geometric distortions can be classified as point-to-point and point-to-plane [7]. Let us assume a collection of points a_j, with $j = 1, 2, ..., N_A$ that form the the reference point cloud A, and a collection of points, b_i, with $i = 1, 2, ..., N_B$ that belong to the the content under evaluation B. For each point b_i of the point cloud under evaluation B, the nearest neighbor in the reference point cloud A is found and the distance between the two points is noted as $d^{B,A}(i)$

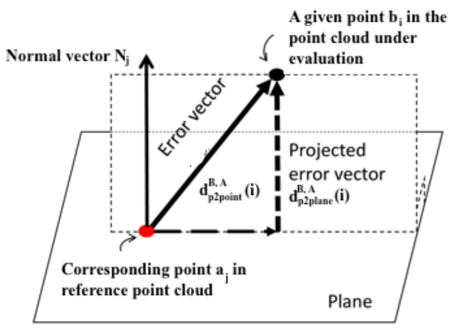


Figure 5: Point cloud metrics

1. Point-to-point error is calculated by connecting each point b_i of the content under evaluation B to the closest point a_j that belongs to the reference point cloud A. In Figure 5, point-to-point error is marked as $d^{B,A}_{\text{p2point}}(i)$.

2. For the calculation of the point-to-plane error, the normal N_j of each point a_j of the reference point cloud should be given or estimated. Point-to-plane error of a point b_i of a content under evaluation B is calculated by measuring the projected error along the normal N_j of the closest point a_j that belongs to the reference point cloud A. This applies when both stimuli have identical number of points. Otherwise, every point b_i of the content under evaluation B is associated with a non-overlapping nearest neighborhood of points that belongs to the reference A. Then, an average normal vector of the points classified into the same neighborhood is computed. Finally, the point-to-plane error of a point b_j under assessment is calculated by measuring the projected error along the associated averaged normal vector. In Figure 5, point-to-plane error is marked as $d^{B,A}_{\text{p2plane}}(i)$, which equals $||d^{B,A}_{\text{p2point}}(i) \cdot N_j||$.

Geometric errors between the original and the processed point clouds can be estimated either using the mean Euclidean distance defined in Equation 4, or the Hausdorff distance defined in Equation 5 for both point-to-point and point-to-plane cases (using $d^{B,A}_{\text{p2point}}(i)$ and $d^{B,A}_{\text{p2plane}}(i)$ instead of $d^{B,A}(i)$, correspondingly).

$$d_{\mathrm{Euc}}^{B,A} = \frac{1}{N_B} \cdot \sum_{i=1}^{N_B} d^{B,A}(i) \tag{4}$$

$$d_{\mathrm{H}}^{B,A} = \max_{\forall i \in B} d^{B,A}(i) \tag{5}$$

To calculate objective scores in quality assessment of point clouds, it is rather common to use the symmetric error distance, which is obtained by setting both contents as reference and estimating the absolute values of error in both cases. Then, the maximum value is considered, as shown in Equations 6 and 7.

$$d_{\mathrm{Euc}}^{\mathrm{sym}} = \max\{d_{\mathrm{Euc}}^{B,A}, d_{\mathrm{Euc}}^{A,B}\} \tag{6}$$

$$d_{\mathrm{Haus}}^{\mathrm{sym}} = \max\{d_{\mathrm{Haus}}^{B,A}, d_{\mathrm{Haus}}^{A,B}\} \tag{7}$$

Finally, such absolute values do not provide meaningful results when contents of different dimensions are assessed. For this purpose, the Peak-to-Signal Noise Ratio (PSNR) is proposed. In the literature, it is defined as the squared distance of the diagonal of the minimum bounding box, d_{BB}, divided by the squared error value (i.e., squared average Euclidean distance or squared Hausdorff using Equations 6 and 7 in 8, correspodingly). In this lab, though, the contents are already scaled down to fit in a minimum bounding box of 1 and, thus, PSNR will not be used.

$$d^{\mathrm{PSNR}} = 10 \cdot \log_{10} \left(\frac{d_{\mathrm{BB}}}{d^{\mathrm{sym}}} \right)^2 \tag{8}$$

1.4 Comparison of objective and subjective results

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 (CC), which quantifies the linear correlation between the MOS values and the predicted values. The Root-Mean-Square Error is estimated using the formula:

  $$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (x_i - y_i)^2}{n}} \tag{9}$$

  while the Pearson correlation coefficient is defined as:

  $$\mathrm{CC} = \frac{n \cdot \sum_{i=1}^{n} x_i \cdot y_i - \sum_{i=1}^{n} x_i \cdot \sum_{i=1}^{n} y_i}{\sqrt{n \cdot \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2} \cdot \sqrt{n \cdot \sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2}} \tag{10}$$

  being x_i, y_i, $i = 1, \dots, n$ the two datasets of n values each, to be compared.

  The value of CC 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 (RC), 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:

$$\mathrm{RC} = 1 - \frac{6 \cdot \sum_{i=1}^{n}(\mathcal{R}(x_i) - (\mathcal{R}(y_i))^2}{n \cdot (n^2 - 1)}, \tag{11}$$

being x_i, y_i, $i = 1, \dots, n$ 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 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 Pearson and Spearman correlation coefficients, as calculated by Equations 10 and 11 in Section 1.4.

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. Subjective raw scores and individual results are always randomized, so even the investigators don't have information connecting individual participants with concrete answers. 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 Familiarized with point clouds, polygon meshes and GUI

At this point, the main goal is to get the data provided to you and visualize point clouds and polygon meshes in order to understand the positive and negative aspects of using each one of these representations. At first, download `data.zip` from moodle "Lab 4 - Data". After you unzip this folder, you will find one sub-folder with the executables (`software`), one sub-folder with the 3D models (`contents`), one sub-folder with matlab files (`mat`) and an excel file with an alphabetical assignment of IDs (`dummy_IDs`). **Do not change the order of enclosing folders!** Open the `contents` folder and check the naming convention of the models. Recall that point clouds are stored as .pcd files and polygon meshes as .ply (you can use WordPad to open them and check how information is stored). On the file names you will notice an increasing index. As the index is increasing, more points and polygons are used to represent the object, which usually implies better visual quality. To visualize the contents:

1. Open a terminal
2. Go to the `software` folder where the executables are stored
3. Run the software using the following comand:

   `visualizer.exe xxx.yyy`

   where xxx denotes the name of the content (e.g., bunny_Q2 - check `contents` folder) and yyy the extension (.pcd or .ply for point clouds and polygon meshes, respectively)

4. If you want to visualize more than one contents simultaneously use the following command:

   visualizer.exe $\texttt{xxx}_1$.$\texttt{yyy}_1$ $\texttt{xxx}_2$.$\texttt{yyy}_2$ -multiview 1

Please, spend some time at this step in order to get familiarized with the 3D representations and the interaction part through the given GUI (e.g., rotate the object, zoom in and out). Of course, you can maximize the window to see the objects more clearly. This GUI will be also used in the experiments.

Please note that there will be cases where the polygons will disappear and you will be able to visualize just the underlying points! This has to do with hardware limitations along with non-optimal use of the rendering of objects as it is performed by this library. By releasing the cursor, the polygons will appear again! At this step there is no problem against this behaviour (except of being a bit annoying). **Please make sure during the experiment that you will rate the visual quality of contents based on polygons and not based on points!** A simple tactic would be to rotate or zoom the contents at your will in order to find the desirable viewpoint, release "quickly" the cursor, and then compare the objects!

3 Participation on the subjective evaluation of point clouds

Without this task you will not be able to finish the lab 4. In this exercise, the task consists in performing two short subjective experiment in semi-controlled environment. This exercise will help us to collect real raw scores from subjective evaluation. In particular, after you finish one experiment, a .txt file with your data will be generated in the `software` folder, including in the file name the dummy ID that is given to you (check in `dummy_IDs` excel file your ID). This data will be analyzed by you in the following exercises. **It is critical to use the correct dummy ID that is assigned to you and send the generated .txt file to the lab assistants, before the end of the lab session**. The lab assistants will instruct you about when each of the experiments (i.e., DSIS and SSIS) shoud be conducted. The plan is to perform the DSIS on Wednesday 22/11 and the SSIS on Thursday 23/11.

3.1 Tasks

1. Log in one of the computer in CO4/5 (you need to setup the resolution of your display to maximum: 2560x1440 pixels in CO5 and 1920 x 1200 in CO4).
2. Open a terminal
3. Go to the `software` folder where the executables are stored
4. Run the software for **training** as follows:

   ```
   training_DSIS.exe -id YOUR_DUMMY_ID
   ```

   In this step, you will get familiarized with the artifacts that are introduced due to different rendering configurations. This step will provide you references in order to understand how you should rate the contents under evaluation during the test. While running this executable, it is critical to read simultaneously the document that is given to you, because it involves details related to how the evaluation should be done.
5. **After finishing the training**, run the software for **testing** as follows:

   ```
   testbed_DSIS.exe -id YOUR_DUMMY_ID
   ```

   In this step, you will actually do the experiment. **Please make sure that once you start the test, you remain focused on identifying impairments on the contents under evaluation without being distracted from external sources until the test is finished.** Remember that after you finish the experiment, a .txt file with your data will be found on the `software` folder and you should send it to the lab assistants before the end of the lab session.

You will repeat the same procedure in order to complete the SSIS experiment.

4 Comparison of different rendering configurations

The goal of this exercise is to compare the performance of 5 different rendering configurations that have been applied on the selected surface reconstruction algorithm on 5 different contents, as specified in Table 1, for each test methodology (i.e., DSIS and SSIS).

1	*Bunny*
2	*Cube*
3	*Dragon*
4	*Sphere*
5	*Vase*

(a) Contents

1	*Q1*
2	*Q2*
3	*Q3*
4	*Q4*
5	*Q5*

(b) Configurations

Table 1: Material used in the subjective experiment

4.1 Tasks

1. After download `data.zip` from moodle "Lab 4 - Data", open the `mat` folder. You will find the following .mat files:

   - `content_lut.mat`: content lookup table (25×1 vector)
   - `configuration_lut.mat`: configuration lookup table (25×1 vector)
   - `subj_scores_METHOD.mat`: raw subjective scores using the corresponding test method (i.e., DSIS or SSIS), each column correspond to the ratings of one subject (25×20 matrix). These scores are randomly generated in range 1-5 for purposes of the lab 4, so you are able to prepare your scripts. After the subjective experiments described in Section 3 are done (Thursday 23/11 in the afternoon), you will be able to download the real subjective scores from moodle "Lab 4 - Data: Real scores". Then, you will replace these random data with the correct scores in order to produce the correct results! Do not re-scale the scores!!
   - `obj_metrics.mat`: contains four vectors `METRIC_values`, each having results of one of the corresponding objective metric introduced in Section 1.3 (4 metrics, each is 25×1 vector).

   Each row corresponds to a specific condition, i.e., a specific combination of content and rendering configuration, which can be determined using the lookup tables and Table 1.

   Example: `content_lut(i) = 3`, `configuration_lut(i) = 2` corresponds to content *Dragon* and configuration *Q_2* for rendering. The corresponing values for each of the objective metrics is given by `METRIC_values(i)`. The corresponing subjective scores are given in `subj_scores_METHOD(i,:)` (following MATLAB notation).

2. Write a function to compute the MOS values and corresponding 95% CIs for each test condition (content, configuration). You should have two 25×1 vectors as result.

3. Plot the MOS values together with CIs for each content (i.e., on the same graph, plot 5 "MOS vs configuration" curves corresponding to the 5 contents).

4. Plot objective scores for each content (i.e., on the same graph, plot 5 "Objective scores vs configuration" curves corresponding to the 5 contents; plot one graph for each objective metric).

5. Comment upon the obtained results.

   - Are subjective scores affected by the selection of content?
   - Are objective scores affected by the selection of content?

6. **When the subjective evaluation, in which you have to take part, is done (Thursday 23/11 in the afternoon), you will be able to find the correct subjective scores on moodle. Don't forget to replace the random data by the real subjective scores to provide correct results within your report!!!**

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

Examples of the graphs required as a results of this exercise can be found in [8].

5 Benchmarking of objective metrics

The goal of this exercise is to benchmark the different objective metrics described in Section 1.3 using the subjective scores as ground truth. For this purpose, you will use again the subjective scores and the objective measurements that are already computed. The data is composed as explained in Section 4.

5.1 Tasks

1. All necessary data are available in the given `mat` folder (check 1st task of Section 4.1).
2. Plot the MOS values together with CIs for each metric for all sequences (i.e., on the same graph, plot the "MOS vs objective metric" for all contents; plot one graph for each metric).
3. For each metric, fit the objective measures to the subjective scores, using linear and cubic fitting.
4. Plot the fitted curves on the graphs.
5. Measure the Pearson and Spearman correlation coefficients, as well as the RMSE, between the fitted scores and the actual subjective scores and report these values in a table.
6. Comment upon the obtained results:
   - Which metric performs better in terms of correlation with the subjective results?
   - What is the influence of the fitting process?
   - Do you observe any influence of the selection of contents on the performance of the objective metrics?

   Useful MATLAB commands: corr, polyfit

Examples of the graphs required as a results of this exercise can be found in [8].

6 Comparison between test methodologies

The goal of this exercise is to compare the influence of applying different test methodologies in subjective quality assessment, using the subjective scores of the DSIS as ground truth. For this purpose, you will use the subjective scores that are given to you. The data is composed as explained in Section 4.

6.1 Tasks

1. All necessary data are available in the given `mat` folder (check 1st task of Section 4.1).
2. Plot the MOS values together with CIs of DSIS against the MOS values along with CIs of SSIS for all sequences (i.e., on the same graph, plot the "MOS of DSIS vs MOS of SSIS" for all contents).
3. Fit the subjective scores of SSIS to the subjective scores of DSIS, using linear and cubic fitting.
4. Plot the fitted curves on the graphs.

5. Measure the Pearson and Spearman correlation coefficients, as well as the RMSE, between the fitted scores and the subjective scores of DSIS and report these values in a table.

6. Comment upon the obtained results:

   - To your opinion, which test method performs better?
   - Do you observe any specific tendency on the way the contents are rated using either of both methods?
   - Is there any influence of the test method adopted on the content under evaluation? (i.e., is there any content that is rated differently by the two test methods?)

   Useful MATLAB commands: corr, polyfit, herrorbar

Examples of the graphs required as a results of this exercise can be found in [8].

References

[1] M. Levoy and T. Whitted, *The use of points as a display primitive.* University of North Carolina, Department of Computer Science, 1985. [Online]. Available: https://www.cs.princeton.edu/courses/archive/spring01/cs598b/papers/levoy85.pdf

[2] 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.

[3] 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.

[4] Kazhdan, Michael and Hoppe, Hugues, "Screened poisson surface reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 3, p. 29, 2013.

[5] ITU-R Reccomendation BT.500-13, *Methodology for the subjective assessment of the quality of television pictures*, International Telecommunications Union Std., January 2012.

[6] ITU-T Reccomendation P.910, *Subjective video quality assessment methods for multimedia applications*, International Telecommunications Union Std., April 2008.

[7] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro, "Evaluation metrics for point cloud compression," ISO/IEC JTC m74008, Geneva, Switzerland, January 2017.

[8] E. Alexiou and T. Ebrahimi, "On the performance of metrics to predict quality in point cloud representations," in *Applications of Digital Image Processing XL*, vol. 10396. International Society for Optics and Photonics, 2017, p. 103961H.

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

INFORMED CONSENT FORM FOR PARTICIPANTS IN RESEARCH STUDIES

Project: "***Quality assessment of point clouds geometry***"

Project supervisor: Prof. Dr. Touradj Ebrahimi
Investigator: Evangelos Alexiou, Irene Viola (PhD candidates)
EPFL STI IEL GR-EB, ELD 225 (Bâtiment ELD), Station 11,
CH-1015, Lausanne
evangelos.alexiou@epfl.ch

Thank you for your interest in taking part in this research. Before you agree to take part, the person organizing the research must orally explain to you the objectives and context of the project.

If you have any questions arising from the oral explanations given to you, please ask the researcher before you decide to join in. You will be given a copy of this Consent Form to keep and refer to at any time.

The questions asked during the subjective evaluations have the goal to facilitate the processing of the ratings that will be recorded. Your answers will be treated in a strictly confidential manner.

Last name, first name: ……………………………………………………………...

You are ☐ male ☐ female

Your date of birth (day/month/year) ……………………………………………………………….

By signing this form, you attest to be at least 18, not to be under the tutelage or legal guardianship, to have well understood the experiments goals and the task that you will be asked to perform. You also freely agree to take part in this study. Your signature also attests that the bank information you provided in order to proceed to the payment of the compensation are correct.

- The experimenter has informed me orally about the purposes of this study.
- I have read and understood the information about the study. I have received sufficient answers to questions about my participation in this study. On written request, I will receive a copy of the present folder (information, consent form).
- I voluntary take part in this study. I can quit this study at any moment without explaining my reasons and without any consequences.
- I accept that the data from the experiment is recorded and processed in scientific purposes only and by respecting the confidentiality, in accordance with the CH Federal law on data protection ("Loi fédérale sur la protection des données"- RS 235.1). I also accept that scientific publications are realized from the obtained results.
- I have been informed that possible damage to my health, which is directly related to the above study and is demonstrably the fault of EPFL, is covered by the general liability insurance of EPFL (insurance policy no. 30/5.006.824 of the Baloise Insurance). However, beyond the before mentioned, my health insurance and/or accident insurance will apply.
- I had enough time to think before taking my decision.

Check here if agree:

☐ I accept that my picture is used in an eventual scientific publication.
☐ I have read the text above and I agree to take part in this experiment.

Subject's signature: …………………………… Date: …………………… Time: ……………………