Objective Methods for Measuring QoE

How well do Video Quality Indicators (VQI) correlate with Mean Opinion Scores (MOS)?

Author: Jakub Nawała, Mikołaj Leszczuk

# Correlation of the VQI with the MOS

# 

Your main task is to find out what is the correlation between the Video Quality Indicators (VQI) indicator and Mean Opinion Scores (MOS). Those should be calculated using the subjective data generated during the “Subjective Methods of Measuring QoE” classes. The goal is to assess how well an objective measure (VQI) can predict a subjective quantity (MOS). One of many ways to verify potential prediction capabilities is to calculate the Pearson correlation coefficient.

In order to find the correlation, you first need two data vectors – VQI values and MOS values for each PVS (Processed Video Sequence; in this case PVS stands for the image tested). To calculate the VQI distorted (i.e. PVS) video sequences were needed. You could find those in the video sequences tested folder. For details on how to calculate the VQI you could refer to the <https://qoe.agh.edu.pl/indicators/>.

Having the VQI for each PVS you need to compare those with MOS values. MOS values can be calculated using CSV files provided (Monday\_Group\_Result.csv and Thursday\_Group\_Result.csv). Each CSV file represents a table of values extracted from the subjective experiment.

The MOS for each video sequence can be calculated by averaging all scores given to a video sequence.

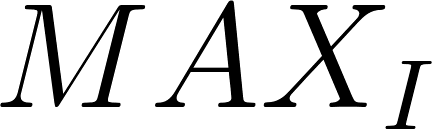
Using the vectors with VQI values and MOS values you can calculate the correlation between the two. This result (being in the range [-1, 1]) says whether there is any linear correlation between the VQI and the MOS. For details on how to calculate the Pearson correlation in Matlab please refer to the [Pearson Correlation Coefficient sub-section](#_c65qbyy3fzw2).

# Theoretical Fundamentals

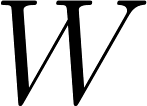
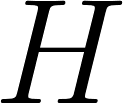
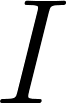
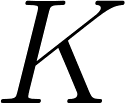
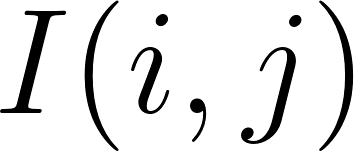
# 

This section explains two concepts crucial to deal with the objective of this laboratory. The first of them is Peak Signal-to-Noise Ratio (PSNR) and the second one is Pearson Correlation Coefficient. Both of them are described in the two following subsections.

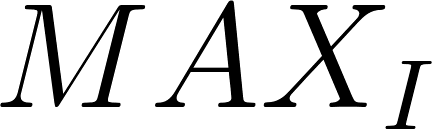
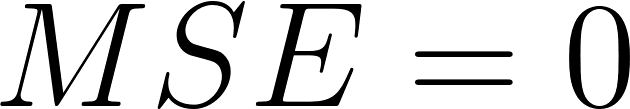
## PSNR

PSNR stands for Peak Signal-to-Noise Ratio. In the context of digital image processing it is a measure of difference between two images. It is based on pixel-by-pixel analysis of image content. To be more precise, it uses mean squared error (MSE) measure and maximum possible pixel value ([](https://www.codecogs.com/eqnedit.php?latex=MAX_I%250)). [Equation 1](#3a1wibqfs9yw) gives a formal definition of MSE and [Equation 2](#hhfcflmwqyvm) uses it to define the PSNR.

|  |
| --- |
|  |
| **Equation 1.** Mean squared error between two images I and K |

[](https://www.codecogs.com/eqnedit.php?latex=W%250) and [](https://www.codecogs.com/eqnedit.php?latex=H%250) stand for width and height of images, respectively. [](https://www.codecogs.com/eqnedit.php?latex=I%250) and [](https://www.codecogs.com/eqnedit.php?latex=K%250) are the two images being compared. [](https://www.codecogs.com/eqnedit.php?latex=I(i%2Cj)%250) stands for value of the pixel being in the i-th row and j-th column.

|  |
| --- |
|  |
| **Equation 2.** Peak Signal-to-Noise Ratio (PSNR) between two images I and K |

Importantly, [](https://www.codecogs.com/eqnedit.php?latex=MAX_I%250) has the value of 255 for an 8-bits deep image (i.e. pixels’ values expressed as 8-bits numbers). Also notice that PSNR takes the value of infinity if there is no difference between the two images (i.e. when [](https://www.codecogs.com/eqnedit.php?latex=MSE%20%3D%200%250)).

To calculate the PSNR in Matlab issue the psnr(A, ref) command. For its description please refer to the [“A List of Useful Matlab Functions and Their Descriptions” section](#_qggsqqzg9zug) and Matlab’s documentation.

## Pearson Correlation Coefficient

The Pearson correlation coefficient describes an extent of the linear correlation between two variables. It takes values from the range [-1, 1], where 1 means total positive linear correlation (when one is growing, the other is growing as well), 0 means no linear correlation and -1 means total negative linear correlation (when one is growing, the other is decreasing). [Equation 3](#bxoss68fskuq) shows a formula for the Pearson correlation coefficient for two random variables A and B. Importantly, both those random variables are assumed to have [](https://www.codecogs.com/eqnedit.php?latex=N%250) scalar observations.

|  |
| --- |
|  |
| **Equation 3.** The Pearson correlation coefficient for two random variables A and B, each having  scalar observations |

[](https://www.codecogs.com/eqnedit.php?latex=a_i%250) is the i-th observation of random variable (r.v.) A, [](https://www.codecogs.com/eqnedit.php?latex=%5Cbar%7Ba%7D%250) is the sample mean of r.v. A and [](https://www.codecogs.com/eqnedit.php?latex=s_a%250) is the sample standard deviation of r.v. A. The same goes for r.v. B.

To find the Pearson correlation coefficient in Matlab use the corrcoef(A,B) command. For its description please refer to the [“A List of Useful Matlab Functions and Their Descriptions” section](#_qggsqqzg9zug) and Matlab’s documentation.

# 

# A List of Useful Matlab Functions and Their Descriptions

# 

|  |  |
| --- | --- |
| A = imread(filename) | Loads an image to the A array. |
| imshow (A) | Displays an image from the A array. |
| imshow(A, 'initialmagnification','fit') | Displays an image from the A array in a bigger size (if the image is small). |
| [nc nl] = size(M) | Shows the size of the image M (nc - number of columns, nl - numbers of lines). |
| A1 = double (A) | Changes the type of a numerical array to double (needed for some operations). |
| A = uint8 (A1) | Changes the type of a numerical array to uint8 (8-bit unsigned integer). |
| A1 = im2double(A) | Converts the A image to the double type. If the input image is of the uint8 type (pixel values in a discrete range [0, 255]), the output image will contain pixel values in a continuous range [0, 1]. |
| subplot(m,n,p) | Divides the current figure into an m-by-n grid and creates an axes for a subplot in the position specified by p. Matlab numbers its subplots by row, such that the first subplot is the first column of the first row, the second subplot is the second column of the first row, and so on. |
| title(‘your title’) | Adds the ‘your title’ title to the plot. |
| P = psnr(A, REF) | Calculates the peak signal-to-noise ratio for the image in array A, with the image in array REF as the reference. |
| R = corrcoef(A,B) | Returns the matrix of correlation coefficients between two random variables A and B. A and B should be column vectors. |

# Additional Exercise

# 

You are encouraged to write your own Video Quality Indicator (VQI). In particular, we would like you to write an indicator correlated with exposure time (also called shutter speed). To make things easier we are providing you with an algorithm. Let us also assume that we are working with images and not videos (in practice, most of the algorithms operate frame-by-frame). Your task is to transform the algorithm into a code. Here is the algorithm:

1. Convert the input image into grayscale representation with 8-bits deep pixels.
2. Divide the image into 8x8 pixels blocks[[1]](#footnote-1).
3. Find a sum of pixels in each block.
4. Find four (4) largest and four (4) smallest sums.
5. Calculate average luminance using those 4 largest and 4 smallest sums.
   1. Please keep in mind that we are talking about a pixel-wise average. Thus, when using 8 blocks, each 8x8 pixels large, the total sum has to be divided by 512 (because 8 blocks \* (8 pixels \* 8 pixels) = 512 pixels).
6. The average luminance approximates exposure time.
   1. We can say that this average luminance is correlated with exposure time, but defining a precise relationship is much more demanding.

The task is worth **11** points and a deadline is the end of the semester. We expect a source code, which we could test with our image samples. We prefer Python 3, but accept a language of your choice if Python is not your thing.

# Copyright Notice

# 

This laboratory exercise utilises [LIVE Image Quality Assessment Database](http://live.ece.utexas.edu/research/Quality/subjective.htm). Please see the text below for its copyright notice.

Copyright (c) 2005 The University of Texas at Austin

All rights reserved.

Permission is hereby granted, without written agreement and without license or royalty fees, to use, copy, modify, and distribute this database (the images, the results and the source files) and its documentation for any purpose, provided that the copyright notice in its entirety appear in all copies of this database, and the original source of this database, Laboratory for Image and Video Engineering (LIVE, <http://live.ece.utexas.edu>) and Center for Perceptual Systems (CPS, http://www.cps.utexas.edu) at the University of Texas at Austin (UT Austin, <http://www.utexas.edu>), is acknowledged in any publication that reports research using this database.

The database is to be cited in the bibliography as:

H. R. Sheikh, Z. Wang, L. Cormack and A. C. Bovik, "LIVE Image Quality Assessment Database Release 2", <http://live.ece.utexas.edu/research/quality>.

IN NO EVENT SHALL THE UNIVERSITY OF TEXAS AT AUSTIN BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS DATABASE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF TEXAS AT AUSTIN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF TEXAS AT AUSTIN SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE DATABASE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF TEXAS AT AUSTIN HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

The following input images are from the CD "Austin and Vicinity" by Visual Delights Inc. (<http://www.visualdelights.net>) coinsinfountain.bmp, dancers.bmp, flowersonih35.bmp, studentsculpture.bmp, carnivaldolls.bmp, cemetry.bmp, manfishing.bmp, churchandcapitol.bmp, building2.bmp.

These images were modified from the original (resized) and then distorted to obtain images in the database. Permission to release these images and their distorted versions was graciously granted by Visual Delights Inc. These images may not be used outside the scope of this database without their prior permission.

The rest of the images were public domain Kodak PhotoCD images obtained from the Internet.

1. Many legacy compression algorithms compress video frames in such 8x8 pixels blocks. [↑](#footnote-ref-1)