

ReadMe:

HDR-VQM version 2 (released July. 2015)

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHORS DISCLAIM ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The software for HDR-VQM has been developed to objectively measure HDR video quality, and requires src (reference) and hrc (distorted).

For meaningful quality prediction, it requires display-processed sequences i.e. HDR video data that has been approximately scaled according to the HDR display. Such display-processing can be done in different ways, and the current version of HDR-VQM software provides implementation for one simple two approach (more details are available in the following article).

M. Narwaria, M. Perreira Da Silva, and P. Le Callet, "HDR-VQM: An objective quality measure for high dynamic range video," Signal Processing: Image Communication, vol. 35, no. 0, pp. 46-60, 2015.

If you find this software (or part of it) useful for your work, we request you to cite the above article.

Description:

This version (developed entirely in Matlab) contains the code for HDR-VQM as well for the pre-processing of the HDR videos before quality computation. Also, both the reference and distorted videos are assumed to be aligned spatially and temporally.

Note that for HDR-VQM the input src and hrc MUST be pre-processed based on the display intended to be used for viewing HDR videos.

To compute HDR video quality, following steps are needed:

1. Create a Matlab directory say C:\Users\XYZ\MATLAB\HDR-VQM\ . This is the working directory.
2. Add to Matlab search path: `addpath(genpath('C:\Users\XYZ\MATLAB\HDR-VQM\'))`
3. Download both HDR toolbox and HDRITools, and add/install them:
 - a. For reading and writing .hdr format, this version uses 'hdrimread' and 'hdrimwrite' from the HDR toolbox (available at <http://www.advancedhdrbook.com/>). HDR-VQM also uses a few other functions from this toolbox, so install it prior to using HDR-VQM.
 - b. For reading and writing .exr format, 'exrread' and 'exrwrite' from HDRITools is used (HDRITools can be downloaded from <https://bitbucket.org/edgarvhdr/hdrtools>)
4. Specify the paths where HDR data is. There should be 4 folders:

path_src_native: path of the folder with src (i.e. reference video) in the native format

path_hrc_native: path of the folder with hrc (i.e. distorted video) in the native format

path_src_emitted: path of the folder with display-referenced (i.e. display-processed) src

path_hrc_emitted: path of the folder with display-referenced (i.e. display-processed) hrc

Eg. *path_src_native* = C:\Users\XYZ\MATLAB\HDR-VQM\native_src_folder_name\

Here, 'native_src_folder_name' is the name of the folder with native src files (frames). Same convention follows for other arguments (folder paths).

Note: the folders with HDR data need not be in the working directory. Simply specify the paths to them.

5. If needed, edit the configuration file 'config_hdrvqm.m' and set all the required parameter values and options (more on this below). Default values are also included in the configuration file.

Details of configuration file (hdrvqm_config.m):

- *data*: specify either 'image' or 'video' (default)
- *do_adapt*: this option is used to specify which display-based processing is requested (2 options: 'none', 'linear')
 - 'none': no processing is requested and HDR-VQM will assume that the input data is scaled according to display (default). With this option, any values above maximum and/or below the minimum displayable luminance will be clipped.
 - 'linear': performs a linear scaling of the data based on the entire sequence (not individual frames). The details of this linear scaling are available in the article mentioned above.

- *do_parallel_loop*: whether to use parallel processing in Matlab, '0' (no parallel processing) or '1' (default). Using parallel processing obviously decreases metric run time but can be computationally expensive.
- *n_scale* and *n_orient*: no. of scales and orientations for Gabor decomposition (default values set to 5 and respectively)
- *perc*: specify the percentage for pooling (default is set to 30%)
- *rows_display*: set the no. of rows in the HDR display (default: 1080)
- *columns_display*: specify the no. of columns in the HDR display (default: 1920)
- *area_display*: specify the area of the HDR display in cm^2 (default: 6100)
- *max_display*: maximum displayable luminance in cd/m^2 (default: 4500)
- *min_display*: specify the black level, min. displayable luminance in cd/m^2 (default: 0.03)
- *viewing_distance*: set the viewing distance in cm. (default: 178)
- *frame_rate*: no. of displayed frames per second (fps) (default: 25)
- *fixation_time*: fixation duration for spatio-temporal pooling (default 600 ms)

6. Two modes can then be used for computing quality:

Mode 1:

The default mode in which the software assumes Rec. 709 for computing luminance channel, and supports formats currently handled by the HDR toolbox (.hdr, .pfm) or HDRITools (.exr). The command is:

`hdr_vqm(path_src_native, path_hrc_native, path_src_emitted, path_hrc_emitted)`

The term 'native' is used to indicate that the HDR video data is in its native format. Hence, the values may only represent luminance information up to an unknown scale. Such data is not used for subjective or objective quality assessment, and needs to be first processed according to the HDR display on which the videos will be viewed. Hence, we use the term 'emitted' to represent values that have been at least approximately scaled (calibrate) according to the display. Those emitted values are then used for objective quality measurement.

Thus, *path_src_emitted* and *path_hrc_emitted* denote the paths of empty folders where the display processed files will be put in, if 'linear' option has been selected. Quality computation will be based on these processed sequences. If the option 'none' (i.e. no display-processing requested) has been set in the configuration file, then the user will put the video data in those folders and use the command:

`hdr_vqm(path_src_emitted, path_hrc_emitted, path_src_emitted, path_hrc_emitted)`

Mode 2:

In this mode, the user can provide the frames manually. This allows to use other color spaces and/or formats, and any other processing that user may apply prior to quality computation. The command is:

```
for frame_count = 1:total_frame
    src_luminance = function_read_frame_user_defined();
    hrc_luminance = function_read_frame_user_defined();
    HDRVQM = hdr_vqm_mannual(src_luminance, hrc_luminance, frame_count, total_frame);
end
```

'src_luminance' and 'hrc_luminance' respectively denote the luminance channel of the video reference and distorted video frames (read with user-defined function), 'frame_count' is the frame index and 'total_frame' indicates the total number of frames in the sequence.

In this mode, the user should ensure that the input frames are scaled according to display luminance, as the software processes the frames without any modifications.

As described in the paper, HDR-VQM will be zero for perfect quality, and will increase with decreasing subjective quality. The software will also output a normalized HDR-VQM score which will be 1 for perfect quality and decrease as subjective quality decreases. The normalized score is derived as

$$\text{HDRVQM}_{\text{normalized}} = \frac{4}{1 + \exp(\text{HDRVQM})} - 1$$

Description of display processing:

'none':

This option specifies that the user has processed the data according to display. Hence data will be used as it is (except that any values above maximum and/or below the minimum displayable luminance will be saturated). It is therefore advisable to ensure the data represents display-referred values and not just relative luminance values.

'linear':

This will produce frames that have been linearly scaled to fit the display range depending on the format (.hdr or .exr). The said scaling factor is computed from the entire sequence. Moreover, for robustness, the scaling factor is derived from the average of 5% brightest pixels (and not just the maximum value). Details can be found in:

M. Narwaria, M. Perreira Da Silva, and P. Le Callet, "HDR-VQM: An objective quality measure for high dynamic range video," Signal Processing: Image Communication, vol. 35, no. 0, pp. 46-60, 2015.

We employed this method for rendering HDR videos using the 'HDR mode' in SIM2 HDR display for the subjective experiments reported in the above paper. Hence, the frames from 'linear' processing should display reasonably well on SIM2 display using the in-built 'HDR mode' (in this mode a factor of 179 is applied by the display in case of .hdr format). Note that with 'linear' option and .hdr format, the resulting scaled values need to be further multiplied with 179 (luminous efficacy) (and clipped if needed) to obtain approximate display constrained luminance. This is automatically done in HDR-VQM software before quality is computed. But it should be kept in mind that the scaled frames put in the folders with paths *path_src_emitted* and *path_hrc_emitted* do not represent display-referred luminance (as mentioned they should be further multiplied by 179 and possibly clipped for that). In case of .exr format, the frames will have approximate display referred luminance.

Note:

The 'linear' option described above does not take into account other factors such as PSF, coarse led sampling etc. To what extent these need to be considered in objective video quality estimation is not clear. Our initial experiments indicated that they may have less impact on the accuracy of the objective methods (more studies are needed to confirm/challenge this). Nevertheless, more accurate display processing can be achieved by other methods such as dual modulation. One such method is described in:

Narwaria et. al "Dual modulation for LED-backlit HDR displays" High Dynamic Range Video – From Acquisition to Display and Applications (ed.), Elsevier, 2015.