MSU Video Quality Measurement Tool (VQMT) Documentation

PRO & Premium version 14.1

Jun, 2022
COMPRESSION.RU Team
CS MSU G&M Lab Video Group
video-measure@compression.ru
https://www.compression.ru/vqmt/

Table of Contents

Overview

Brief Description

MSU Video Quality Measurement Tool (MSU VQMT) is professional software that is used to perform deep comparative objective analysis of video quality. The main functionality of this software is to calculate objective quality metrics for digital multimedia content (video or image) using reference (when comparing one or several processed/compressed/distorted video sequences to original one) or non-reference (when analyzing content and getting mark of its quality) types of analysis.

MSU VQMT is developed by COMPRESSION.RU Team with participants **MSU Graphics & Media Lab. Video Group** (G&M Lab.). Project leader: dr. Dmitry Vatolin.

To use MSU VQMT with Python refer to MSU VQMT Python Interface.

See also: MSU VQMT online help on https://videoprocessing.ai.

Main support email: video-measure@compression.ru.

Main application areas

Codec developers Modern metrics are close to human perception. Control quality loss with VQMT.

Television industry. Be sure in picture quality.

Studios. Control quality in automatic or manual mode on stage of production

Medical equipment developers. Analysis quality and parameters of medical devices signal.

Video providers. Reduce storage and transmission cost.

Research groups & Universities. Use VQMT to control arbitrary parameters of video and images.

Individuals. Writers, bloggers, journalists in video segment.

AV system manufacturers. For camera, video compressors, transmitters developers.

We use MSU VQMT in own projects, for example:

- Moscow State University, CS MSU Graphics & Media Lab. Annual Codecs Comparison https://www.compression.ru/video/codec_comparison/index_en.html.
- MSU Benchmarks. https://videoprocessing.ai/benchmarks/
- See also x264 Codec Capabilities analysis from YUVsoft Corp. http://yuvsoft.com/pdf/x264_parameters_comparison.pdf.

Editions & API

FREE license is an edition for individuals. It supplies only GUI edition for Windows and has limitations.

DEMO license is an edition for testing PRO/Premium features. This version is useless - computed values are incorrect. However it can be used to test VQMT features and performance.

Trial license is an edition for companies considering purchasing VQMT. This edition can be requested on VQMT website. The standard trial period is 2 weeks.

PRO license is an edition for companies and individuals, who wants to use all abilities of MSU VQMT GUI and CLI interfaces on Windows or Linux. It helps them to carry out massive comparisons of their technologies using batch processing of big numbers of files and flexible options for measurements without technical limitations. One PRO verison license grants access to one Windows and one Linux platform simultaneously. We have very attractive volume discounts.

PREMIUM license is an edition for companies, who actively uses MSU VQMT. It doesn’t limit installations, supports automatical activation, using in clouds, on virtual machines and more.

SDK license is a special expensive license for companies that want to incorporate VQMT into their workflow.

System Requirements

Supported Windows systems (GUI and CLI):

- Microsoft Windows 7 x64 SP 1 or later, with installed updates: KB2999226, KB976932, KB2533623.
- Microsoft Windows 8 x64, 10 x64 or later.

Supported Linux systems (command-line interface only):

- Debian family (Ubuntu 16.04 or later)
- Red Hat family (CentOS 7 or later)
- Support for other standalone Linux systems – by request.

Hardware requirements:

- 2 GHZ x86-64 Processor
- Processor with AVX support required on Linux
- 4 GB of RAM
- 200 Mb free disc space

We recommend:

- 4+ core processor with AVX support
- 8 GM RAM
- Graphical adapter with at least 1 GB video RAM.

We recommend Premium version for usage in clouds, containers, virtual machines and with other virtualization technologies. The PRO activation is guaranteed to work only on standalone workstations, laptops.

Main Features

Three types of User Interface:

1. Command-Line Interface (Windows & Linux, Professional only);
2. Graphic User Interface (Windows only, Free & Professional);
3. Python Interface. Can be installed via PyPI package.

Various input video formats:

- Video files (*.avi; *.avs; *.dat; *.divx; *.f4v; *.flv; *.h265; *.m2ts; *.m2v; *.m4v; *.mkv; *.mov; *.mp4; *.mpg; *.mts; *.mxf; *.ogm; *.ogv; *.qt; *.ts; *.vob; *.wmv; *.265; *.3g2; *.3gp; *.3gpp and others);
- Raw files with 1-16 bit integer color depth or 32-bit floats, different colorspaces and subsamplings;
- Image files and image sequences: *.bmp, *.bw, *.cut, *.dds, *.exr, *.g3, *.gif (only static images), *.hdp, *.hdr, *.ico, *.iff, *.j2c, *.j2k, *.jif, *.jng, *.jp2, *.jpe, *.jpeg, *.jpg, *.jxr, *.lbm, *.mng, *.pbm, *.pcd, *.pct, *.pcx, *.pgm, *.pic, *.pict, *.png, *.ppm, *.psd, *.ras, *.rgb, *.rgba, *.sgi, *.targa, *.tga, *.tif, *.tiff, *.wap, *.wbm, *.wbmp, *.wdp, *.webp, *.xbm, *.xpm;
- Full-HD, 4K, 8K and other resolutions support;
- TIFF images with 16 and 32 bit (floating point) color depth;
- AviSynth scripts – it can be very useful when using VOB, WMV and other files as input for MSU VQMT;
- Video files via AviSynth scripts auto generation;

Base metrics are included:

1. PSNR;
2. HDR PSNR;
3. MSE;
4. MSAD;
5. Delta;
6. Delta ICtCp;
7. SSIM (superfast, fast, precise, OpenCL, CUDA);
8. HDR SSIM;
9. MSSSIM (superfast, fast, precise, OpenCL, CUDA);
10. HDR MS-SSIM;
11. 3SSIM (CPU, OpenCL, CUDA);
12. stSSIM (temporary excluded);
13. VQM;
14. HDR VQM;
15. NIQE (no-reference);
16. VMAF;
17. SI (no-reference)
18. TI (no-reference)

Additional objective quality metrics from MSU:

- MSU Brightness Flicking Metric (with source code);
- MSU Brightness Independent PSNR (with source code);
- MSU Drop Frame Metric (with source code);
- MSU Noise Estimation Metric (with source code);
- MSU Scene Change Detector (with source code);
- MSU Blocking (no-reference);
- MSU Blurring (no-reference);

In command-line mode metrics can be calculated simultaneously. Also metrics supports:

- Sets of color components;
- ROI;
- Saving results to CSV of JSON files
- Visualization;
- Different averaging modes;
- Plug-ins interface with SDK that gives user possibility to create their own metrics;
- Possibility to compare several files in one comparison – typical situation when comparing original uncompressed file with several compressed files with different codecs or presets;
- Possibility to save “bad” frames for every comparison with flexible options – it can be very useful when comparing two (or more) codecs and user wants to see frames where these codecs have maximum difference, the lowest/highest metric value, etc.

Changelog

14.1 (June 2022). Python wrapper

Main features:

- + Python wrapper as pip package msu_vqmt
- + New color policy: specifying exact colorspace for metric calculation
- * Modern decoders from FFmpeg 5
- * Better Linux support - more systems supported
- * Free policy updated - no resolution limitation

Other features:

- * New VQMT logo
- * VQMT SDK updated
- + Specifying custom file path for visualization
- + More json output
- * Fixed color components order in GUI
- * Bugfixes

14.0 BETA (Jan 2022). SSIM, MS-SSIM Superfast metrics

Main features:

- + Support for big files up to 1'000'000 frames and more
- + GUI: Possibility to select any number of files
- + GUI: You can save & load results, including plots
- + GUI: Measure history. Open results of resent measurement
- + Metrics: new metrics SSIM Superfast, MS-SSIM Superfast: speedup - 10 times
- + New documentation

Optimization:

- + Initialization speedup
- + Memory consumption redused while processing big files: up to 20 times
- + All CPU metrics speedup about 5-10%

GUI:

- + Full path in result window
- + Pagination in result table
- + Dragging any number of files to GUI
- + Setting up process priority
- + Arrow showing better values on the plot
- + New tab Summary. Appears when multiple files were processed.
- * Improved scrollbars behaviour

CLI:

- + Possibility to set skipping mode
- + New section Summaty in the output.
- + Help will be displayed in separate sections
- + Output of JSON list of devices
- + Output of saved files in performace JSON output

+ Added option -v: output general info in readable or JSON fromat

Metrics:

+ SSIM-metrics can compute weighted average over Y, U, V
+ SSIM-metrics can build visualization of all components in one image
+ PSNR could take into account subsampled U, V components
+ Confidence interval length in VMAF now is tunable
* ci95_low and ci95_high in VMAF output renamed to ci_low and ci_high
* MSSSIM renamed to MS-SSIM

Other:

* Output: Fixed names of autogenerated files
* Changed field names in VQMT JSON configuration
* Bugfixes

13.1 (May 2021).

+ VMAF now supports "No enhancement gain" (NEG) features
+ VMAF calculation on CPU — 1.5 times faster
+ VMAF supports JSON models
+ VMAF legacy mode
+ Optimization of OpenCL metrics: 1-4 times faster than 13.0
+ PIPE mode: read any numbers of inputs from arbitrary commands output
* Bugfix: settings were not restored
* Bugfix: rebuilding index each time
* Bugfix: flood of FFmpeg messages
* Bugfix: some formats supports
* Bugfix: temporal metrics in Preview
* Bugfix: crashes
* Other bugfixes

13.0 BETA (Feb 2021). Colorspaces, HDR metrics, AV1

Main features:

+ FFmpeg 4 in VQMT - improved navigation and format support
+ Support for AV1
+ NIQE now supports OpenCL (~40 fps on FullHD on tested devices)
+ Native support for EXR files

+ Colorspaces and color standards support in VQMT
+ Visualization pallets and gamma correction

Visualization & Preview:

+ Support for 11 pallets for metrics visualization
+ Support gamma correction for metrics visualization
+ Turning visualization parameters right in Preview window
+ New visualization formats: PNG, JPEG, BMP, fixed H264
+ Visualization video will be saved with correct FPS
+ Preview by channel, any colorspace

+ Separate colorspace setup for render and values

Colorspaces:

+ Fixed L-LUV metrics
+ Conversion between different colorspaces and models
+ Support for CIE XYZ, CIR LUV, ICtCp, LMS, Perceptual uniform encoding
+ Manually specifying input file colorspace and automatic detection
+ Manually specifying sample range for floating-point input files
+ Setting of display maximum luminance, that affects GDR-metrics calculation

New metrics:

+ HDR VQM
+ HDR PSNR
+ HDR SSIM
+ HDR MS-SSIM
+ Delta ICtCp

Other:

+ Using arbitrary external codec as input
+ VQMT don't save the index to the input file folder anymore
+ Different verbosity levels
+ Setting implicit output CSV file path
+ New command line arguments: -cs, -csv-default-file, -csv-file, -display-luminance, -float-range, -vis-colormap, -vis-gamma
* Bugfixes in online metrics and online visualization
* Oher bugfixes

Known issues:

! Speed of SSIM metrics on OpenCL could be slower
! Results after RGB → YUV conversion may differ from VQMT 12 ones
! Meaning of L-LUV channel changed, results for L colorspace will differ from VQMT 12 ones

12.1 (Feb 2021). VMAF on OpenCL, Non-binary masks

Main features:

+ VMAF now supports OpenCL (~40 fps on FullHD on tested devices)
+ VQMT now can be configured with JSON-file instead of command line
+ Online help for VQMT
+ Support for non-binary masks

Masks:

+ Mask-support fixed and refactored
+ Support the simple setting of rectangular masks
+ Native mask support in some metrics (PSNR, Delta, MSE, MSAD, SSIM-Fast, SSIM-Precise, SSIM-OpenCL, MSSSIM-Precise, VMAF, VMAF-OpenCL, NIQE)
+ Mask logging

Performance:

+ "Lazy" metrics' loading, VQMT initialization speedup.

New metrics:

+ Timeshift

Visualization in metrics:

+ TI
+ VMAF

Other:

+ JSON-configuration from STDIN and command line
* Fixed AviSynth on Windows
* Other bugfixes

12.0 BETA r12434 (Jan 2020).

* VMAF freezing on platforms with a big amount of logic units
* VMAF wrong results on platforms with a big amount of logic units if run for multiple files at once
* Freezes of Preview window in some cases
* Crashes in the case of unaligned subsampled picture type (i. e. odd yuv420)
* AviSynth and AVIFile support fixed (Windows)
* Mask support fixed
* JSON fixes - removed trailing None in legacy mode
* CUDA error correct handling
+ Newer OpenCL driver
+ Saving properties of the Result window

12.0 BETA (Nov 2019). Online metrics, acceleration

+ Online metric calculation
+ Visualization for specific frames
+ Computed metric values in Preview
* VQMT will not stop if some files, but not all, are done
* Improved multi-core support - acceleration up to 3 times
+ Automatic color selection in command line possible
* Color specification in command line changed
* Now colors named by single letter, i. e. Y instead if YYUV
+ PSNR metric now supports single value for RGB and YUV
* Single PSNR metric instead of 4
+ Processing of multiple colors simultaneously possible in GUI
+ Preview window: zoom
+ Preview window: timestamp & frame
+ Preview window: no freezing when changing frame
+ Preview window: pixel indicator & pixel information
+ Preview window: individual pixel values
+ Preview window: clickable miniview
+ Preview window: hotkeys improved
+ Preview window: vertical comparison mode
+ Preview window: slider mode
+ Preview window: information in fullscreen mode

+ NIQE: visualization
* NIQE will output common mean and NIQE mean as separate accumulated values
+ More average values: statistical data, specific metric values
* New JSON format: column-vise JSON values
+ Bitrates in JSON
* New CSV format
* Legacy mode to bring VQMT closer to VQMT 11
+ Subsampling mode: skipping frames to improve performance
+ Performance settings in GUI, new performance setting . metric parallelism
+ Sample conversion setting
+ More RGB ↔ YUV tables
* Changed value range in the following metrics: Delta, MSE, MSAD
* Option "-subsampling" renamed to "-no-upscale-uv"
+ Considerably improved precision for SSIM Fast and MSSSIM fast metrics
+ Corrected precision in SSIM-CUDA, SSIM-OpenCL, Blocking
+ Some metrics optimized
+ VQMT internally optimized
+ Results plot window optimized
+ Completely refreshed VQMT log output
+ Information in console window: FPS, estimated time, etc.
+ Average values in console output
+ Additional information in log: average FPS, exit status, etc
+ More accurate FPS during calculation
+ Consider correct sample range in all metrics
+ New bad frames features

11.2 (Nov 2019).

+ CentOs Linux now is supported
+ Added FFmpeg scaling algorithms
* Fixed PSNR average value
* Fixed OpenCL metrics (SSIM, 3SSIM, MSSSIM)
* Fixed incorrect values in bi_psnr metric
* Fixed BFM metric
* Fixed NIQE metric
* 3SSIM-CUDA bug fixed
* Error reporting in CUDA metrics, accuracity in CUDA metric fixed for compatibility with OpenCL, GPU_Id metrics
* CSV fixes - correct escaping, line endings
* RGB48 pixel format support fix
* Other bugfixes

11.1 (May 2019). Geometry correction

+ Comparison on different resolution (geometry correction)
+ New metrics (SI, TI)

+ Saving bitrates
* And bugfixes

11.0 (Feb 2019). 1400+ raw formats, wider HDR

+ Support for 1400+ new RAW formats
+ Support for standard names for RAW formats
+ Support new RGB ↔ YUV tables
+ Support HDR videos and more HDR image formats: 3 more times video formats now can be lossless
+ Result table with metric values inside VQMT
+ Automatic command-line generation from GUI
+ Plot of bitrate
+ Actual VMAF support: modern models, 4k models, confidence intervals
+ New blurring metric
+ New Noise Estimation metric
+ Linux command line considerably improved
+ Better performance on some cases
+ New quality control standards
+ Measurement on subsampled U and V
+ Latest CUDA engine
* And bugfixes
* Only x64-bit edition from this version

10.2 (Feb 2019).

+ PREMIUM licensing introduced
* And bugfixes

10.1 (Apr 2018). New metrics: VMAF, NIQE

+ Support of perspective and modern VMAF metric. Many settings guarantee full metric support.
+ Now it's possible to determine quality even without a reference via new NIQE metric.
+ Improved usability: Drag & Drop files and changing of file order by single click.
+ The -stdin parameter allows you to transmit data directly from the output of another program, such as FFmpeg, without saving multi-gigabyte files. Acceleration and simplification (PRO and DEMO only)
+ New formats with support for HDR: extended support for TIFF and PXM family. (HDR only in PRO and DEMO)
+ Better usability of command line - a lot of improvements while maintaining full backward compatibility (PRO and DEMO only)
+ Built-in profiler. It will show which operations take the most time and allow you to accurately measure the performance of VQMT (PRO and DEMO only)
+ Better JSON output

+ TIFF as format for bad frames

+ More functions in Linux version: more metrics, plug-ins support, OpenCL support (PRO and DEMO only)
* And bugfixes
* Now we recommend specifying original input using new key '-orig' instead of traditional '-in'

10.0 BETA (Apr 2017). New main window, Linux support

+ New look of main window
+ Saving and loading VQMT projects
+ Status bar with diagnostic in main window
+ Report generation
+ Saving result plot in arbitrary image formats
+ New clear VQMT folder structure
+ Improved using vqmt console in PRO version
+ Command line Linux utility

9.1 (Apr 2017).

* Fixed incorrect results while using second processed video
+ Added legend to result plot
+ Saving log from GUI
* Fixed copying plot to clipboard
* Using *.YUV bug fixed: first frame could be duplicated
* Improved h265 support
* Index file building bug fixed

9.0 BETA (Dec 2016). New result window

+ New look of result window
+ Using interface while calculation is processing and while viewing results
+ Viewing multiple results at the same time
+ Switch between calculation log and results plot
+ Extended information about progress: count of processed frames, FPS, elapsed time, estimated time
+ Pretty and configurable result plot
+ Saving CSV after calculation finished
+ Saving JSON result file in GUI
+ Saving results' plot as SVG file

8.1 (Nov 2016).

+ Added checking for updates.
+ Using single image as source do not truncate all other inputs to one frame.
* Fixed crashes when use plug-in metric.
* Improved video previewing, fixed opening visualization video inside VQMT.
* Fixed possible incorrect negative PSNR.
* Fixed mask behavior.

+ Recognize patterns 720p etc. in the name of RAW file to detect resolution automatically.
+ Hotkeys in Preview and Fullscreen window considerably improved.
+ Allow to change video source and frame inside Fullscreen window.
* Bugfixes in Preview window performed.
* Fixed green frame in some types of file, incorrect reading of some color spaces.

8.0 BETA (Oct 2016). Preview window

+ Added new visualization methods: Lossless Video, Lossy Video and TIFF files.
+ Preview window replaced.
+ Full-screen preview, display selection.
+ Side-by-side preview.
+ Inspecting visualization inside VQMT.

7.1 (Oct 2016).

* Crashes fixed.
* Improved command line output in PRO version.
* Preview display fix.
* Seeking and offsetting YUV files fixed.

7.0 BETA (Jun 2016). OpenCL support, Metrics reorganized

+ Added support for OpenCL device interface. Efficient calculation on different devices.
+ Speedup of PSNR and SSIM metrics.
+ Similar metrics joined.

6.2 (Jun 2016). JSON support

+ Added support for JSON output in console.
* Incorrect results with RGB metric on RGB video fixed.

6.1 BETA (Apr 2016).

* Fixed bug: hangs in console while using files in network directories.
* Fixed incorrect behavior of .Y4M files in GUI.
* Fixed crashes when the length of all files was not able to be detected."

6.0 BETA (Mar 2016). Acceleration

+ VQMT became about 3 times faster.
+ 16 and 32 bpp floating point TIFF files supported (PRO version).
+ Added "-threads" command line argument to PRO version, which allows to control CPU usage.
* Advanced mask processing - allowed shifts from exact black color for indication black area.
* Fixed crashes and incorrect results while using mask.

5.2 (Apr 2016).

* Fixed bug: hangs in console while using files in network directories.
* Files incorrect behaviour of .Y4M files in GUI."

5.1 (Mar 2016). Ranges and offsets support

+ Support comparing ranges of input videos.
* Fixed crashed while using YUV files with more than 8bpp.
* Corrected RGB <-> YUV conversion procedures.
* Fixed incorrect transformation for some input formats.
* 10, 14 and 16 bpp RGB format plain order changed to R, G, B.

5.0 BETA (Sep 2015). Open wizard, Command line refactor

+ Added Open File Wizard to help with opening and previewing files;
+ Use wizard or file picker for selection of input file;
+ Wizard supports drag&drop mechanism;
+ Now user can select mode for opening file (FFmpeg, AviFile, automatic AviSynth, RAW file, image or image sequence and others);
+ Special mode to compare the results of opening file using different modes;
+ Automatic selection of the best open mode for specified file;
+ More settings for opening process customization;
+ Added automatic generation of video index file that helps file to be opened and correctly previewed;
+ Number of supported codecs available for opening and previewing increased (supported for new versions of video codecs);
+ Added support for images as input files: *.jpg, *.png, *.tif, more formats of *.bmp and many others;
+ Added support for using image sequences as input video, automatic detection of sequence;
+ Previewing raw files (*.yuv, etc.) "on the fly";
+ Detection of resolution of raw files from file name;
* PRO console interface is more user friendly with full compatibility to previous version;

4.4 (Sep 2015). GPU support improved

+ Number of files available for metric calculation increased;
+ The number of supported devices for running CUDA metrics increased;
* Unable to run metric for single file bug fixed;
* Unable to view results if some minor error occurred bug fixed;
* CUDA metrics crash fixed;

4.3 BETA (May 2015).

+ Number of files available for metric calculation increased. Now metric can be calculated for the files that not available for preview;
+ Speed up of file opening in metric calculation process for some types of files;
+ Standard VQMT plug-ins is now supported;
* Memory leak fixed in VQMT.

4.2 BETA (Apr 2015).

+ Added native support for *.mkv, *.flv and some other containers and codecs;
* Stability fixes.

4.1 BETA (Mar 2015). Native reading video files via FFmpeg

+ Added FFmpeg file reading support, the number of supported formats greatly increased. Using AviSynth is not recommended;
* Fixed x64 crashed;
* Stability fixes.

3.2 (Feb 2015). AviSynth support improved

* Existence of AviSynth determining fixed;
+ AviSynth for VQMT as standalone installer;
* AviSynth plug-in opening fixed;
* Fixed unsuccessful file opening in AviSynth mode;
+ AviSynth mode now supported in console;
+ All dependencies now are immediately in installer, no more redistributable packages needed;
+ Main menu and desktop labels fixed to determine Pro, Free and Pro Demo license;
+ Executable file metadata errors fixed;
+ Cosmetic fixed in Interface and file naming: revision number added to naming;
* Fixed crashes in 64-bit on multiple platforms;
* Fixed crashes and hangs after: the press of Process button, viewing of analyses result, other events.

3.1 (Nov 2012).

+ Changed to CUDA 5.0 toolkit, added Kepler support (Compute Capability 3.0)
* Stability fixes
* Fixed major bug with masking

3.0 (Jun 2011). GPU support, 64-bit version

+ Added stSSIM metric
+ Added ".y4m" raw video internal support
+ Added Autoupdate feature for free (our PRO customers receive updates automatically)
+ Added GPU realization for SSIM-based metrics (SSIM, 3-SSIM, MS-SSIM. Requires CUDA-capable device. See metrics info page for additional info)

* Added subjective comparison for the most popular metrics (see metrics info)
* Added 64-bit of MSU VQMT (up to 10% speedup)

! Program crashes due memory lack when -metr ALL specified with large (i.e. 1280x720) video frames.

2.7.3 (Oct 2010).

* Fixed bug with MSSIM metric causing source frame change

* Fixed some metrics inaccuracy causing different metric values by enabling\disabling visualization
! Program crashes due memory lack when -metr ALL specified with large (i.e. 1280x720) video frames

2.7.2 (Aug 2010).

* Fixed bug causing incorrect metric values, when using 3SSIM and MSSSIM simultaneously
* Fixed bug causing incorrect PSNR metric values in CSV files
* Fixed bug causing no metric calculation for large (>4gb) files
* Not existing directory specified in "-cod" parameter will be created now and processing will not cancel.
! Program crashes due memory lack when -metr ALL specified with large (i.e. 1280x720) video frames

2.7.1 (Jun 2010).

* Fixed bug in CVS file generation. Sometimes first frame metric value was empty
* Fixed bug causing incorrect MSE metric values after calculating SSIM metric

2.7 (May 2010). MSSSIM, 3SSIM metrics added

+ MSSSIM (fast and precise) metric implemented
+ 3SSIM metric implemented
* Fixed bug in calculation of VQM metric under Windows 7
* Fixed bug during program launch on some machines

2.6 (Jan 2010). Windows Vista & Windows 7 support

* Fixed bug in Scene Change Detection plug-in when working under Windows Vista or Windows 7
* Fixed bug in saving visualization video when running on Windows Vista or Windows 7
* Fixed dependency with vcomp.dll

2.5 (Nov 2009).

+ Video with any resolution is now supported by all metrics. Video with resolution which is not appropriate for some metric is now expanded (via data duplication, separately for each metric) to make resolution acceptable
+ 1.95 times speed up of command line tool multiple metrics calculation on average (PRO only)
+ YUV files with size more than 2Gb are supported now
+ Output directory for *.CSV and visualization files is automatically set to folder of last specified reference file
* Fixed bug in processing of *.YUV files with non-standard resolution
* Fixed bug in loading the mask from *.YUV files
* Fixed bug in masking of L (LUV colorspace) component
* Fixed bug in processing of non-standard resolution *.AVS files
* Fixed bug in calculation of SSIM (precise) for second reference file

* Fixed bug in conversion from RGB32 to YUV color spaces for video with non-standard resolutions (affects calculation of metric for *.AVI files)

2.01 beta (Apr 2009). Masking added

+ 1.5 times speed up of command line tool multiple metrics calculation on average (PRO only)
+ Masking is added
* Fixed bug in 4:2:2 raw files with more than 8 bits per component support

2.0 BETA (Mar 2009). HDTV support, Deep RAW files

+ *.MOV, *.VOB, *.WMV, *.MP4, *.MPG, *.MKV, *.FLV formats support simplified
+ HDTV support (PRO only)
+ Raw files with more than 8 bits color depth per component are supported (PRO only)
+ Alternative SSIM and PSNR are added for compatibility with other implementations
+ New of *.CSV files with average metric values (PRO only)
+ Minor acceleration
+ Preview buttons are added
+ Options save is improved
+ All MSU plug-ins are renamed (names are now more correct in GUI and simpler to call from PRO console)
* MSU Noise Estimation plug-in bug with incorrect (identical) values for some videos is fixed
* MSU Noise Estimation and MSU BI-PSNR plug-ins provide correct information about their home pages now
* MSU BI-PSNR plug-in crash during visualizing a metric for video sequences with dimensions less than 255 is fixed
+ Now it is possible to compress visualization

1.0 (May 2006).

+ More YUV file types are supported, including YV12, YUY2, YUV
+ Supports unicode
+ Visualization dialog was extensively reworked
* Interface is more user-friendly

Command-line Help

Alphabet index

Command	Category	Short description
-activate	Misc.	running activation wizard
-bad	Bad frames	turning on/off saving bad frames (frames with the worst metric values)
-bad-dir	Bad frames	specifying directory for bad frames
-bad-format	Bad frames	specifying bad frames format
-bad-name	Bad frames	specifying scheme for naming bad frame files
-bad-num	Bad frames	specifying how many bad frames should be saved
-bad-radius	Bad frames	specifying the minimum distance between bad frames
-bad-threshold	Bad frames	specifying threshold to ignore bad frame for some metric values
-bad-type	Bad frames	specifying the type of bad frames
-cng	Output	specifying default name scheme for CSV, JSON files
-config	Misc.	specifying a JSON configuration file
-config-line	Misc.	specifying JSON configuration as command line argument
-config-stdin	Misc.	using STDIN for reading configuration JSON
-cs	Input	overriding input file colorspace
-csv	Output	turning on/off saving CSV file with results of metrics
-csv-default-file	Output	requesting writing to the default CSV file
-csv-dir	Output	specifying directory for default CSV and JSON files
-csv-file	Output	specifying an exact file for CSV output
-ct	Output	specifying csv data delimiter
-dev	Metrics	specifying a device to perform the metric on
-display-luminance	Sample interpretation	specifying display luminance for HDR metrics and colorspaces
-float-range	Input	specifying value range for float sampled files
-forced-length	Performance	specifying expected file length

Command	Category	Short description
-fpd	Output	setting floating point for CSV files
-gui	Misc.	running GUI (only Windows)
-h, -?, --help (1)	Help, information	requesting CLI help
-h, -?, --help (2)	Help, information	request information about a group of options
-h, -?, --help (3)	Help, information	requesting help for metric
-h, -?, --help (4)	Help, information	requesting help for VQMT config format
-idx	Indexing	specifying index mode for reading input files
-idx-file	Indexing	specifying a custom scheme for an index file
-json	Output	turning on json output
-json-default-file	Output	requesting writing to the default JSON file instead of stdout
-json-file	Output	specifying a file for JSON output
-json-fmt	Output	specifying legacy json format
-legacy-mode	Sample interpretation	setting up legacy mode
-list	Help, information	Requesting additional information
-log	Output	specifying logfile
-mask	Masking	specifying mask file
-mask-type	Masking	specifying mask type and configuration
-metr, <metric name>	Metrics	specifying metric and (optionally) color component
-metr-config	Metrics	setting configuration line for plugin metrics
-metric-parallelism	Performance	setting the number of instances of each metric that can work simultaneously
-no-upscale-uv	Input	enabling of disabling upscaling for chroma components (UV) of subsampled files
-orig, -in	Input	specifying an input file to apply metrics

Command	Category	Short description
-performance	Performance	requesting information about VQMT performance to stdout
-project	Misc.	specifying a project file, saved from GUI
-quiet	Misc.	setting up silent mode
-range, <range>	Input	specifying frame range for the last file
-resize	Preprocessing	turning on geometry correction of input video - scale and crop, configuring it
-ryt	Sample interpretation	specifying the default color standard for files
-sample-conversion	Sample interpretation	changing integer sample interpretation
-set	Metrics	setting additional parameter for metric of file reader configuration
-skip-frames	Performance	setting up frame skipping for increasing performance
-slots	Performance	setting the number of slots for simultaneous image processing
-stdin, -stdin-orig	Input	specifying stdin as input yuv or y4m file
-summary-accum	Metrics	setting accumulator for summary output for a metric
-summary-col	Metrics	setting column for summary output for a metric
-terminal	Misc.	running CLI
-threads	Performance	specifying the number of threads for basic VQMT operations
-v, --version	Help, information	getting information about this VQMT copy
-vis	Visualization	turning visualization on/off
-vis-caption-pos	Visualization	specifying the position of caption labeled over the visualization
-vis-colormap	Visualization	specifying colormap for visualization
-vis-dir	Visualization	specifying directory for visualization
-vis-gamma	Visualization	specifying gamma correction for visualization
-vis-preset	Visualization	specifying preset for selected visualization format

Command	Category	Short description
-vis-print-frame	Visualization	enabling or disabling printing frame number on visualization
-vis-print-value	Visualization	enabling or disabling printing values of the metric on visualization
-vis-type	Visualization	specifying visualization format
-vng	Visualization	specifying visualization filename scheme
<open mode>	Input	specifying mode for reading a file
<picture type>	Input	specifying picture type for YUV file(s)
<resolution>	Input	specifying resolution for YUV file(s)

Help, information

Getting help on command-line options, information about VQMT features, lists of metrics, supported formats and devices.

Usage	Description
-h -? --help	Request help information about VQMT command-line section
-h *<help section>* -? *<help section>* --help *<help section>*	Request information about specified group of options
-h *<metric>* -? *<metric>* --help *<metric>*	Request extended information about the specified metric
-h config -? config --help config	Request VQMT configuration JSON description. Configuration file completely describes VQMT calculation process. You can configure VQMT via -config option
-v [json] --version [json]	Get information about VQMT version, edition, available license, build date, etc. It is possible to get this information in JSON format for automatical processing.
-list -list all -list [json] metrics -list modes -list [json] colorspaces -list [json] devices -list raw -list visualizations	Request additional information about available: • metrics (metrics) • file reading modes (modes) • picture types for YUV files (raw) • computing devices: CPU, GPU, etc. (devices) • visualization formats (visualizations) • color standards (colorspaces) • all above (all) Some lists supports json output.

Input

Specifying video input files, streams, pipes, or image sequences. Setting options, colorspaces and formats for opening these files.

Usage	Description
-orig *<file>* -in *<file>*	Use the specified file for metrics. Specify original file using -orig, distorted or other files using -in
-stdin *<mode>* -stdin-orig *<mode>*	Use when a source file (YUV) or Y4M file should be read from standard input. *<mode>* can be: • raw • y4m Use -stdin-orig for original file in case of reference metric
<picture type>	Use after file or stdin specification to specify picture type for RAW (.yuv) input file. The last specified type also will be used as the default one. Use before the first file or stdin specification to set forced default picture type for all RAW files and pipes. View the list of available picture types using -list raw.
*<number>*x*<number>*	Use after file or stdin specification to specify resolution (width and height) for RAW (.yuv) input file. The first declaration becomes the default for all other files. Note: resolution can be determined automatically from the file name, it will be preferred over the default.
-range *<offset type> <range>* -range *<range> <range>*	Use to manually specify and limit frame range for last the input file. <range> is one of : • <number>-<number> : specifying the first and last frame of range; • <number>-: set range from the first frame to the end of the video. <offset type> is one of : • seek: go to the first frame immediately, if possible; • skip: read and skip some frames from the beginning of input file(can be slow for big offsets); • auto: (default) automatically choose between seeking and skipping.

Usage	Description
-float-range *<min>*..*<max>*	Use after a file specification to manually set minimal (black) and maximum (white) value for the file with floating point samples. Applicable to EXR, YUV and other formats for which the range is not specified in the metadata. Float range by default is 0..1. *<min>* and *<max>* could be floating point numbers. For HDR metrics and colorspaces, the maximum color value is interpreted as the display maximum luminance. Note: this range will have no effect if range specified in the file metadata
-cs auto -cs *<colorspace>*	Use after file specification to manually set colorspace that will be considered the colorspace of the input file values. This setting always overrides the color space, even if it does not match the picture type. You can find the list of available colorspaces typing -list colorspaces. Use -cs auto for automatic colorspace deducing (default). In this case VQMT will consider file metadata. If the metadata is not enough to determine the exact colorspace, the default color standard from -ryt setting will be used to choose the colorspace.
<open mode>	Specify mode for reading last the file. This will override the default reading mode. To see all available modes use -list modes;
-no-upscale-uv -no_upscale_uv yes -no_upscale_uv no	Enable or disable upscale of chrome components for subsampled files (Y422, Y422 formats for example). If no (default), inputs will be upscaled to the resolution of Y-plane.

Metrics

Setting up metrics to apply to the input files. Setting color components for analysis and configuring metrics. Selection of devices.

Usage	Description
-metr *<metric name>* [over *<color component list>*] *<metric name>* [over *<color component list>*]	Add a metric to compute overall specified files (original file vs. all other files for reference metric or each file separately for no-reference metric). *<color component list>* is a component or comma-separated list of components, for example, Y,U,V. If over <color component list> is not specified, VQMT will use default component or components according to the following algorithm: • If a metric supports only one color component, it will be selected. • If a metric supports none of Y, U, V, it will be calculated over supported channels from R, G, B. • If a metric supports none of R, G, B, it will be calculated over supported channels from Y, U, V. • If all input files for metric (1 or 2) are in YUV color space, colors from the set Y, U, V available for this metric will be used. • If all input files for metric (1 or 2) are in RGB color space, colors from the set R, G, B available for this metric will be used. • If input files have different colorspaces, an error is generated. Since VQMT 12 additional channels YUV and RGB added(supported by PSNR metric). To specify 3 separate channels, tell Y,U,V. The list of available color components: Y, U, V, L, R, G, B, YUV, RGB. Use in metric list only components, supported by the specified metric. *<color components>* can be ALL - it means all supported components for this metric. Use -list metrics to see all variants.
-metr-config *<config line>*	Set metric configuration line (applicable for plugin metrics only)
-set *<key>*=*<value>*	Use after file to set a parameter for file reader, or after metric to set a parameter for the metric. Use -h <metric> to see parameters available for specified metric.
-summary-accum *<accum name>*	Set default accumulator for outputing summary values. If multiple values are obtained for this metric, files in the output section SUMMARY will be ordered according to this accumulator.

Usage	Description
-summary-col *<column id>*	Set default column id for outputing summary values. If multiple values are obtained for this metric, files in the output section SUMMARY will be ordered according to the column with this id.
-dev *<device>*	Specify a device to perform the last specified metric on. This is possible only if the metric supports this device. *<device>* is device id or a part of the device name. Get a list of available devices using -list devices. To see, what devices are supported by the metric use -h <metric name>

Output

Setting up the output to JSON, CSV, stdout, log and other files that will contain the results of the analysis.

Usage	Description
-json	Use to turn on JSON output to stdout
-json-file *<file>*	Specify a file to write JSON to. This option will enable JSON output. Note: *<file>* is the full path to the only JSON file will be written while VQMT is running. If it exists, will be silently overridden.
-json-default-file	Use to set up saving JSON to file without specifying a filename. Default filename will be used. The default filename includes information about the specified metric(s) and file(s).
-json-fmt *<version>*	Set json legacy format. *<version>* could be: • 11, • 12 (default);
-csv -csv yes -csv no	Use to turn one saving CSV file with results of metrics. A default file will be used. If exists, the file will be silently overridden. Use -csv no to turn CSV saving off.
-csv-file *<file>*	Specify a file to write CSV to. This option will enable CSV output. Note: *<file>* is the full path to the only CSV file will be written while VQMT is running. If it exists, will be silently overridden.
-csv-default-file	Use to set up saving CSV to file without specifying a filename. Default filename will be used. The default filename includes information about the specified metric(s) and file(s).
-cng PREFIX -cng POSTFIX -cng CUSTOM *<filename>*	Specify scheme for default CSV or JSON file name: • PREFIX - metric before files • POSTFIX - files before metrics (default) • CUSTOM - manually specify filename (without extension)
-csv-dir *<dir name>*	Set directory for saving CSV and JSON files (for default files). Default - original file directory
-log *<file name>*	Specify a file to write information about VQMT process
-ct , -ct ;	Set CSV file fields delimiter (default: ,)
-fpd . -fpd ,	Set CSV file floating point (default: .)

Visualization

Setting the output of per-pixel metric values. Visualization will generate an image for each metric and each processed frame.

Usage	Description
-vis -vis yes -vis no	Turn on/off saving visualization for each frame and each running measure. The visualization shows pixel-wise metric values and can be saved to a sequence of images or video file
-vng PREFIX -vng POSTFIX -vng CUSTOM *<file mask>*	Set the visualization file name scheme. A distorted file name and metric will be included in the filename. Specify one of: • PREFIX - metric before the file • POSTFIX - file before the metric (default) • CUSTOM - manually specify filename (without extension). Use %d for specifying a placeholder for frame number if needed. You can also use environement variables in it.
-vis-type *<type>*	Set visualization format (image of video). Use -list visualizations to see all available formats
-vis-preset *<preset>*	Set visualization method preset for the specified visualization format. Use -list visualizations to see all presets.
-vis-caption-pos *<pos>*	Set the position of caption labeled over the visualization where *<pos>* is top or bottom. You can turn off the caption by turning off its components in VQMT PRO
-vis-print-value yes -vis-print-value no	Enable or disable printing value of the metric on visualization frames (default: yes)
-vis-print-frame yes -vis-print-frame no	Enable or disable printing frame number on visualization frames (default: yes)
-vis-gamma *<float value>*	Specify gamma correction for the visualization of the last specified metric. Use before the first metric specification for setting the default gamma correction for all the metrics. You can change the visualization contrast by varying this value. Visualization will still use the same colormap, but the colors of that colormap will be distributed the other way. *<float value>* is gamma correction value between 0.001 and 1000, default 1.

Usage	Description
-vis-colormap *<colormap>*	Specify a colormap for the last metric visualization. Use before the first metric specification for setting the default colormap for all the metrics. Colormap is a pallete, which colors will be used for a single-channel visualization of a metric. This will not affect metrics with 3-channel visualization. *<colormap>* is one of: • vlag • vlag-inversed • icefire • icefire-inversed • rocket-mako • mako-rocket • vqmt-traditional • vqmt-traditional-inversed • segment • segment-inversed • black-white • white-black Default: icefire
-vis-dir *<directory>*	Set directory where visualization will be saved. Default - original file directory

Masking

Setting region of interest (ROI) to indicate areas which the metric should consider with bigger or lesser weight. Can be an input video/image or set by parameters.

Usage	Description
-mask *<file name>*	Set mask file and enable masking. Mask sets interesting areas of an image where the metric will be considered with a bigger weight. The file is specified in the same way as described for -in parameter. Notes: • Only the first color component is taken into account: Y for YUV, R for RGB • Mask can also be a single image if it is static over frames • In the case of a binary mask, each pixel either considered or not while metric calculation. The value will be considered if the corresponding mask color over 0.5 of the maximum color value • A mask value of zero does not mean that the pixel will be completely ignored • Some metrics have native mask support and some don't have. Native support means that the metric will consider pixels that do not match the mask to be non-existent. Metric without native support will consider pixels to be the same for original and distorted but can produce incorrect values.
-mask-type *<mask type> <mask config>*	Set mask type, where *<mask type>* and *<mask config>* could be: • bin black - binary mask, black for ignored areas • bin white - binary mask, white for ignored areas • 8bit black - 8-bit mask, black for ignored areas • 8bit white - 8-bit mask, white for ignored areas • incl-rect *<rect spec>*, excl-rect *<rect spec>* - mask consists of rectangle (included or excluded). *<rect spec>* is ;-separated list of measurements in the order: top, right, bottom, left. Each measurement can be pixel value or percent value (floating point) if followed by '%' sign. A negative value can be used to measure from the opposite side.

Bad frames

Requesting output of frames of input video/image with extreme metric values. VQMT will save the requested number of images containing input frames.

Usage	Description
-bad -bad yes -bad no	Turn on/off saving bad frames - frames with the worst metric values. Saving of bad frames is done after all computations are completed by reopening and rereading input files
-bad-dir *<directory>*	Set directory to save bad frame. Default - original file dir
-bad-type *<type>*	Set type of bad frames, where *<type>* is one of: • ORIGPROC : save frames with the max difference between first and second files; • PROCPROC : save frames with the max difference between second and third files (usually these files meant to be processed files to compare); • FIRSTBETTER : save frames where the second file is better than the third; • SECONDBETTER : save frames where the third file is better than the second;
-bad-format *<format>*	Set an image format for bad frames. *<format>* can be one of: TIFF (default) or BMP
-bad-num *<number>*	Set the maximum number of bad frames to be saved
-bad-radius *<number>*	Specify the radius of bad frames - minimum distance between bad frames
-bad-name *<naming scheme>* -bad-name CUSTOM *<naming scheme>*	Specify scheme for naming bad frames. Where *<naming type>* is one of • frame - start file names from frame (default); • file - start file names from source file name; • CUSTOM - specify a custom name scheme;
-bad-threshold [*<lower threshold>*]..[*<upper threshold>*]	Specify thresholds to ignore bad frames for some metric values. Specify at least one threshold. A lower threshold means that metric must be better than this threshold, upper threshold means that metric must be worse than it (lower threshold can be bigger than upper one if lower values of metric means better quality)

Preprocessing

Correction of images before applying metric. Setting of the scaling to bring the images to the same size or reduce the resolution.

Usage	Description
-resize [[*<implementation>*] *<mode>* [A]] to *<dst>* [crop]	Set up geometry correction of input files or downsample for increasing performance. This is necessary if input files have different resolutions. *<mode>* can be: • crop • nearest • linear • cubic (default) • lanczos and *<implementation>* is the preferred algorithm implementation and can be: • ffmpeg (default) • intel and *<dst>* is destination size and can be: • orig - original size or 1-st input size • 1/2 orig • 1/4 orig • min - the size of input with the smallest resolution • max - the size of input with the biggest resolution • <width>x<height> A means antialiasing, can not be used with crop and nearest. If you specified crop after *<dst>*, the scaling will preserve the ratio, and then the crop will be done. You can not specify crop after *<dst>* if *<mode>* is crop

Indexing

Preliminary analysis of input streams to improve the quality of seeking and accurate information about the progress of the analysis.

Usage	Description
-idx *<index type>*	Specify index mode, which is preferable for reading some types of the input file (especially, coded streams). One of: • no - do not create an index file (default); • needed - create an index file only if needed; • forced - always create an index file; • needed_mem - build an index in memory if needed, no files created; • forced_mem - always build an index in memory (no files created); Default: needed; Note: Index is needed if the source file cannot be correctly opened for seeking. Index file creation has a sense only if FFmpeg or Auto mode specified. While the calculation the file will be read frame by frame, and no index file will be created. The index file can be created when VQMT attempts to save bad frames
-idx-file *<file scheme>*	Set scheme for naming index file. A scheme can use special VQMT variables. Possible if index mode is one of: • needed, • forced. Default: $(VQMTCacheDir)/FileIndexes/$(FileHash).vqmtidx

Sample interpretation

Additional setting of working color standards, transformations of input samples, screen luminance for HDR images.

Usage	Description
-ryt *<color standard>*	Specify color standard and quantization range (PC or TV) that will be used to find exact colorspace for files, where metadata not sufficient to specify exact colorspace. For example, for an integer-sampled file with YUV picture type, the quantized YUV colorspace from specified standard will be used. Use color standards with PC suffix to use PC range for quantized data. The default value is BT.709. Use -list colorspaces to see all variants.
-display-luminance *<luminance value>*	Specify the display luminance for some HDR colorspaces and metrics. This will be treated as the luminance of a maximum white pixel of an input file. This value will be also used in some metrics as maximum. *<luminance value>* is integer in the range 1..10000
-sample-conversion shift -sample-conversion repeat	Set integer samples interpretation. In the mode shift (default for VQMT 11 and earlier) 8-bit value 0xFF of the input file will be interpreted as 255/256. In the mode repeat (default since VQMT 12) 8-bit value 0xFF of the input file will be interpreted as 1.
-legacy-mode *<number>*	Turn on legacy mode with specified VQMT version. This will affect the metric calculation, input interpretation, and conversion. Reproducibility of results of earlier VQMT versions is not guaranteed. Note: setting of other parameters of this group can break reproducibility. Currently supported versions: 11

Performance

Adjusting CPU usage and memory consumption, skipping frames. Requesting output detailes information of VQMT performance after the analysis.

Usage	Description
-skip-frames [*<skip mode>*] fps *<fps>* -skip-frames [*<skip mode>*] each *<N>*	Set up frame skipping to increase performance. You can specify target *<fps>* and VQMT will try to achieve this fps, or you can set up processing if each *<N>*'th frame, skipping N-1 out of every N. <skip mode> is one of : • skip - fully decode skipped frames (default), • seek - try to seek directly to the target frame.
-slots *<number>*	Set the number of slots that be allocated for each metric to process files simultaneously. 0 - default value. Increasing this value will slightly increase memory consumption, but can improve concurrency
-metric-parallelism *<number>*	Set the metric parallelism level - how many instances of a single metric can work simultaneously. 0 - default value. Set small values to reduce memory consumption. High metric parallelism is quite useless if you computing many metrics simultaneously
-threads *<number>*	Set number of threads for basic VQMT operations such as: metric calculation, input reading, input conversion, mask applying, geometry correction, etc. Note: VQMT can spawn additional threads, especially for reading files. This is a number means how many operations can be performed simultaneously
-forced-length *<number>*	Manually set the expected number of frames (0 to disable). Note: this parameter affects only progress bar, to limit frames count length use -range 0-<last frame> after the file specification
-performance	Use this to request information about VQMT performance which will be printed in JSON format to stdout after VQMT finishes computation. This will suppress the common VQMT output. Performance information will contain info about performed operations

Misc.

Performing VQMT activation or reactivation. Interface selection. Specifying a project file or JSON configuration file. Configuring output profile.

Usage	Description
-quiet	Use to suppress printing common VQMT information to stdout. This will not affect JSON output if specified to stdout or performance information
-terminal	Use with any VQMT executable to run in terminal mode (CLI)
-gui	Use with any VQMT executable to run in graphical mode (GUI). This is supported only on Windows systems
-activate	Use to manually run the activation wizard for registering Pro, Premium, or trial licence. Activation wizard allows you to change the active licence for VQMT. Running in this mode will ignore the existing license. This command is allowed for GUI mode
-project *<file>*	Use to load a project, saved from GUI. This command allowed for GUI mode
-config *<file>*	Specify file to use settings from. This file will override all other settings. Configuration file fully describes all VQMT settings. It has JSON format. Unlike -project this cannot be used in GUI mode, because CLI settings are wider than GUI. You can use autogenerated config from JSON output of VQMT to repeate the completed calculation, or use customly-generated JSON. Config of compatible versions will be automatically converted the current version format
-config-line *<json configuration>*	Specify this option and JSON configuration as the next CLI argument. See help for command -config for more information about VQMT JSON configuration
-config-stdin	Specify this option to use STDIN for reading configuration JSON. VQMT will read exactly one JSON value from STDIN in ASCII encoding. See help for command -config for more information about VQMT JSON configuration

JSON configuration description

The configuration is a JSON file that fully specifies VQMT computation settings. It specifies the set of files, the set of metrics, and calculation settings. You can use the configuration file instead of a classical CLI configuration (for any computational purpose, excluding getting help, specification and activation). Using the configuration file is possible in command via the specifying -config option. Also, it is possible to read the configuration JSON from STDIN using -config-stdin. The configuration is available only in the CLI interface in PRO/Premium edition.

- **version** - This configuration file version (optional)
  Syntax: [*integer,integer*]
  First integer for major version, second one for minor. Supported versions:
  - 12.1
  - 13.0
  - 13.1
  - 14.0
  - 14.1

  Default: [14,1]
- **files** - List of all input files. The first will be treated as the original one for reference metrics
  Syntax: [A, ...]. A possible values: One of:
  - "*string*"
  - ["*string*", {*object*}]

  The string is always file name. Possible keys for the object:
  - **mode** - (string) - explicit mode for reading a file. See also List of open modes. Default: "auto"
  - **index** - (string) - index mode
  - **index_file** - (string) - index file. For some index modes will be skipped
  - **props** - (object) - additional properties, specific for mode and file (see List of open modes)
  - **range** - (array) - specify array of one element - [N] to skip N frames. Specify array of two elements [N, M] to select frames from N to M inclusive
  - **stdin** - (bool) - specify true, if the file should be read from stdin
  - **offset_type** - (string) - if range specified, the way for seeking target frame
- **metrics** - List of metrics to perform
  Syntax: [A, ...]. A possible values: {*object*}
  - **metric** - Metric to perform
    Syntax:
    One of:
    - "*string*"
    - {"name":"*string*", "device":"*string*"}
    - {"name":"*string*", "variation":"*string*"}

    Run VQMT with -list json metrics arguments to see the list of all available metrics. Each metric can be uniquely identified by it's name and variation
  - **color** - The color component to perform metric. You can specify only one component per element.
    Syntax: "*string*"
    Run VQMT with -list json metrics arguments to see the list of all available metrics with supported color components
  - **config** - Metric configuration (optional)
    Syntax: {*object*}
    Run VQMT with -list json metrics arguments to see the list of all available metrics with available configuration settings

    Default: {}
    - **vis-gamma** - Gamma correction value for visualization (optional)
      Syntax: null or A, where A possible values: *float*
      Default: null
    - **vis-colormap** - Colormap for visualization (optional)
      Syntax: null or A, where A possible values: “*string*”
      Default: null
- **output** - Settings for writing JSON ans CSV files
  Syntax: {*object*}
    - **enabled** - Turn on/off saving output
      Syntax:
      One of:
        - true
        - false

      Default: true
    - **output_directory** - Output directory for saving CSV and JSON (supports VQMT environment variables)
      Syntax: “*string*”
      Default: “?(InputOriginalDir)”
    - **name_generation_type** - Scheme for default naming of output files (CSV, JSON)
      Syntax:
      One of:
        - “prefix”
        - “postfix”
        - “custom”

      Default: “postfix”
    - **custom_name** - Custom name for output files (supports VQMT environment variables). Only considered if the scheme is CUSTOM
      Syntax: “*string*”
      Default: “”
    - **csv_separator** - Separator for starting new cell in the output CSV file
      Syntax:
      One of:
        - ”;”
        - ”,”

      Default: “,”
    - **fp_separator** - Separator for starting fractional part in floating point numbers for the output CSV file
      Syntax:
      One of:
        - ”.”
        - ”,”

      Default: “.”
    - **save_csv** - Turning on/off sabing CSV
      Syntax:
      One of:
        - true
        - false

      Default: false
    - **save_json** - Turning on/off saving JSON
      Syntax:
      One of:
        - true

        - false

    Default: false
    - **json_legacy** - Switching to JSON format of older VQMT releases. Currently supported: 11
      Syntax: *integer*
      Default: 14
    - **json_file** - Explicit name of the JSON file (supports VQMT environment variables). If set and not empty, has a priority over other specifications. It will override custom_name if set
      Syntax: "*string*"
      Default: ""
    - **csv_file** - Explicit name of the CSV file (supports VQMT environment variables). If set and not empty, has a priority over other specifications. It will override custom_name if set
      Syntax: "*string*"
      Default: ""
    - **json_stdout** - Enabling/disabling saving JSON to STDOUT instead of a file. Do not use with json_file
      Syntax:
      One of:
        - true
        - false

      Default: false
- **vis** - Settings for writing visualization
  Syntax: {*object*}
    - **enabled** - Turn on/off saving visualization for all metrics
      Syntax:
      One of:
        - true
        - false

      Default: false
    - **output_directory** - Output directory for saving visualization (supports VQMT environment variables)
      Syntax: "*string*"
      Default: "?(InputOriginalDir)"
    - **name_generation_type** - Scheme for default naming of visualization files
      Syntax:
      One of:
        - "prefix"
        - "postfix"
        - "custom"

      Default: "postfix"
    - **compressor** - Compressor name for visualization files. See also the list of all compressors
      Syntax: "*string*"
      Default: "h264"
    - **preset** - Preset for setting compressor mode
      Syntax: "*string*"
      Default: "mq"
    - **caption_position** - Place for printing caption over the visualization (if some data are printed)
      Syntax:
      One of:
        - "left-top"

        - “left-bottom”
        - “right-top”
        - “right-bottom”

      Default: “left-top”
    - **write_value** - Turning on/off printing metric value on each visualization frame
      Syntax:
      One of:
        - true
        - false

      Default: true
    - **write_frame** - Turning on/off printing frame number on each visualization frame
      Syntax:
      One of:
        - true
        - false

      Default: true
    - **gamma** - The default gamma correction value for visualization. Can be overwritten by each metric individually by specifying the metric setting
      Syntax: *float*
      Default: 1.0000000000000000
    - **colormap** - The default color table for visualization. Can be overwritten by each metric individually by specifying the metric setting
      Syntax: “*string*”
      Default: “”
- **conversion** - Settings for interpreting samples of input files
  Syntax: {*object*}
    - **standard_name** - Color standard that will be used to find exact colorspace for files, where metadata not sufficient to specify exact colorspace
      Syntax: “*string*”
      Default: “BT.709”
    - **full_range** - Is quantization range PC (full range, 0..255 in case of 8-bit samples). Quantization range will be used to find exact colorspace representation for files, where metadata not sufficient to specify coded data quantization range
      Syntax:
      One of:
        - true
        - false

      Default: false
    - **display_lum** - Display luminance for some HDR colorspaces and metrics. This will be treated as the luminance of a maximum white pixel of an input file. This value will be also used in some metrics as maximum. Value is integer in the range 1..10000
      Syntax: *float*
      Default: 10000.000000000000
    - **sample_conv** - The way an integer sample will be converted to continuous
      Syntax:
      One of:
        - “shift” - 0xFF means 0.996 (255/256)
        - “repeat” - 0xFF means 1.0

      Default: “repeat”
    - **legacy_mode** - Emulation of an old VQMT mode. It affects some metrics and conversion algorithm. Do not use with other settings of this section

    Syntax: *integer*
    Default: 0
  - **disable_UV_upscale** - Disabling upscaling of chroma components (U, V for YUV pixtures types) to Y size for subsampled formats (YUV420, YUV422, etc.)
    Syntax:
    One of:
    - true
    - false

    Default: false
- **bad_frames** - Settings for bad frames
  Syntax: {*object*}
  - **enabled** - Turn on/off saving visualization for all metrics
    Syntax:
    One of:
    - true
    - false

    Default: false
  - **dir** - Directory for saving bad frames (suports VQMT environment variables)
    Syntax: "*string*"
    Default: "?(InputOriginalDir)"
  - **count** - Total (maximum) number of bad frames to be saved
    Syntax: *integer*
    Default: 10
  - **distance** - The minimum distance that bad frames can be (frames)
    Syntax: *integer*
    Default: 0
  - **type** - The criterion for the selection of bad frames
    Syntax:
    One of:
    - "origproc"
    - "procproc"
    - "firstbetter"
    - "secondbetter"

    Default: "origproc"
  - **format** - Image format for saving bad frames
    Syntax:
    One of:
    - "BMP"
    - "TIFF"

    Default: "TIFF"
  - **high_threshold_enabled** - Is high threshold enabled
    Syntax:
    One of:
    - true
    - false

    Default: false
  - **low_threshold_enabled** - Is low threshold enabled
    Syntax:
    One of:
    - true
    - false

    Default: false
  - **high_threshold** - High threshold. A frame with a metric value equal to or better than this will never be considered bad

  Syntax: *float*
  Default: 0.00000000000000000
- **low_threshold** - High threshold. A frame with a metric value equal to or worse than this will never be considered bad
  Syntax: *float*
  Default: 0.00000000000000000
- **naming_type** - Scheme for default naming of bad frames image files
  Syntax:
  One of:
  - "frame"
  - "file"
  - "custom"

  Default: "frame"
- **custom_name** - Custom name for bad frames image files (supports VQMT environment variables). Only considered if the scheme is CUSTOM
  Syntax: "*string*"
  Default: ""

- **geometry** - Settings for geometry correction
  Syntax: {*object*}
  - **enabled** - Turn on/off geometry correction. If off, all files mush have the same resolution. Also, geometry correstion can be used to downscale inputs for increasing performance
    Syntax:
    One of:
    - true
    - false

    Default: false
  - **size** - Target size for resizing or cropping input images
    Syntax:
    One of:
    - "orig"
    - "1/2 orig"
    - "1/4 orig"
    - "min"
    - "max"
    - "custom"

    Default: "orig"
  - **width** - Target width if size is custom
    Syntax: *integer*
    Default: 1920
  - **height** - Target height if size is custom
    Syntax: *integer*
    Default: 1080
  - **strategy** - Strategy, that specifies resampling, cropping and ratio preserving
    Syntax:
    One of:
    - "scale"
    - "scale+crop"
    - "crop"

    Default: "scale"
  - **algorithm** - Image resampling algorithm
    Syntax:
    One of:
    - "nearest"

        - "bilinear"
        - "bicubic"
        - "lanczos"

      Default: "bicubic"
    - **antialiasing** - Is antialiasing while resampling enabled
      Syntax:
      One of:
        - true
        - false

      Default: false
    - **prefer_ffmpeg** - Use FFmpeg scaling algorithm if available. If false, Intel algorithm will be always used
      Syntax:
      One of:
        - true
        - false

      Default: true
- **mask** - Mask specification for indication of the most important areas of the image
  Syntax: {*object*}
    - **mask_type** - Type of mask to be aplied
      Syntax: ["*string*", "*string*"]
      Possible types:
        - bin black - binary mask, black for ignored areas
        - bin white - binary mask, white for ignored areas
        - 8bit black - 8-bit mask, black for ignored areas
        - 8bit white - 8-bit mask, white for ignored areas
        - incl-rect *<rect spec>*, excl-rect *<rect spec>* - mask consists of rectangle (included or excluded). *<rect spec>* is ;-separated list of measurements in the order: top, right, bottom, left. Each measurement can be pixel value or percent value (floating point) if followed by '%' sign. A negative value can be used to measure from the opposite side.

      Default: ["no","mask"]
    - **mask_file** - A mask file if maskType considers a file. Could be static image or video file
      Syntax: One of:
        - "*string*"
        - ["*string*", {*object*}]

      The string is always file name. Possible keys for the object:
        - **mode** - (string) - explicit mode for reading a file. See also List of open modes. Default: "auto"
        - **index** - (string) - index mode
        - **index_file** - (string) - index file. For some index modes will be skipped
        - **props** - (object) - additional properties, specific for mode and file (see List of open modes)
        - **range** - (array) - specify array of one element - [N] to skip N frames. Specify array of two elements [N, M] to select frames from N to M inclusive
        - **stdin** - (bool) - specify true, if the file should be read from stdin
        - **offset_type** - (string) - if range specified, the way for seeking target frame

      Default: ""
- **subsampling** - Skipping frames for increasing performance
  Syntax: {*object*}
    - **enabled** - Is subsampling enabled. Subsampling could be used for some

metrics to skip frames for increasing performance
Syntax:
One of:
        - true
        - false

    Default: true
    - **do_target_fps** - Whether subsampling takes into account target_fps setting. If false, value for every_nth_frame will be taken for subsampling
      Syntax:
      One of:
        - true
        - false

      Default: false
    - **every_nth_frame** - Number of frames that will be skipped after each processed frame plus one. This value will be taken into account only if do_target_fps is False. Value 1 means no subsampling (each 1st frame is processed)
      Syntax: *integer*
      Default: 1
    - **target_fps** - Processing FPS, that VQMT will try to achieve skipping frames
      Syntax: *float*
      Default: 25.000000000000000
    - **enable_seek** - Allow VQMT seek in the input files. If false, VQMT will read all intermediate frames
      Syntax:
      One of:
        - true
        - false

      Default: false
- **performance** - Performance tuning and information
  Syntax: {*object*}
    - **threads_number** - Number of threads for performing VQMT tasks. VQMT can spawn additional threads
      Syntax: *integer*
      Default: 0
    - **slots_number** - Number of slots that be allocated for each metric to process files simultaneously. 0 - default value. Increasing this value will slightly increase memory consumption, but can improve concurrency
      Syntax: *integer*
      Default: 0
    - **metric_parallelism** - How many instances of a single metric can work simultaneously. 0 - default value. Set small values to reduce memory consumption. High metric parallelism is quite useless if you computing many metrics simultaneously
      Syntax: *integer*
      Default: 0
    - **show_performance** - Whether information about VQMT performance will be displayed instead of results
      Syntax:
      One of:
        - true
        - false

      Default: false
    - **forced_length** - The expected number of frames (0 to disable). This parameter affects only progress bar

Syntax: *integer*
Default: 0

- **verbosity** - Verbocity level. Affects on amount of messages, generated by VQMT
  Syntax:
  One of:
  - “silent”
  - “default”
  - “all”

  Default: “default”

Metric’s reference

Principle quality metrics

PSNR (Peak signal-to-noise ratio)

General info

Metric type	full-reference image metric
Value range	(completely different) 0..100 (similar to original)
Value interpretation	bigger is better quality
MSU VQMT implementations	cpu multithreaded
MSU VQMT visualization	pixel-wise
Available colorspaces	R, G, B, Y, U, V, L, RGB, YUV
Output values	metric value
Aggregated values	standard set , total PSNR
MSU VQMT usages	-metr psnr [over <color components>]
External references	Wikipedia

Algorithm description

Metric depends only on difference of original and distorted, and more preciesly, only on L_2-norm of this difference (see MSE). Unlike MSE, metric has logrithmic scale and can be calculated using the following formula:

$$\mathrm{PSNR} = 10 \cdot \log_{10} \frac{\mathrm{MaxErr}^2 \cdot w \cdot h}{\sum\limits_{i=1,j=1}^{w,h} (x_{i,j} - y_{i,j})},$$

where MaxErr – maximum possible absolute value of color component (MaxErr=1 in VQMT), w – video width, h – video height.

Total PSNR is aggregated value, that considers all processed frames as a single huge image and then calculates PSNR. Total PSNR takes into account sum suqared distortion on all frames, and doesn’t distinguish situation where all distorion on one frame from situation where it is distributed across all frames. While arithmetic mean aggregated value depends from geometric mean of MSE’s of each frames, Total PSNR depends on arithmetic mean of them.

In MSU VQMT you can calculate PSNR for all YUV and RGB components and for L component of LUV color space. Also, since VQMT 12 you can calculate over all YUV space or all RGB space, achieving a single value for 3 components. PSNR metric is easy and fast to calculate, but sometimes it is not appropriate to human’s perception.

Benchmark

Please note: values can vary depending on system configuration, input format and other factors.

Benchmark on VQMT 14.0 BETA r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores

GPU: NVIDIA CUDA/GeForce GTX 660 Ti

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	multithreaded	Y	HD 720p	548.93	0.003
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	271.15	0.005
VQMT 14.0 default	multithreaded	Y	4K 2160p	58.47	0.019
VQMT 14.0 default	multithreaded	YUV	HD 720p	406.74	0.004
VQMT 14.0 default	multithreaded	YUV	FullHD 1080p	194.49	0.007
VQMT 14.0 default	multithreaded	YUV	4K 2160p	44.09	0.026
VQMT 14.0 default	singlethreaded	Y	HD 720p	178.93	0.007
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	88.4	0.012
VQMT 14.0 default	singlethreaded	Y	4K 2160p	22.49	0.047
VQMT 14.0 default	singlethreaded	YUV	HD 720p	134.8	0.008
VQMT 14.0 default	singlethreaded	YUV	FullHD 1080p	65.15	0.016
VQMT 14.0 default	singlethreaded	YUV	4K 2160p	22.9	0.047

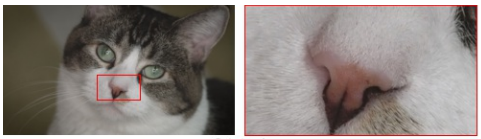
Benchmark on VQMT 14.0 BETA r12792 PRO for Linux.
CPU: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64 cores
GPU: NVIDIA CUDA/TITAN RTX

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	multithreaded	Y	HD 720p	1231.0	0.002
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	669.22	0.004
VQMT 14.0 default	multithreaded	Y	4K 2160p	98.17	0.019
VQMT 14.0 default	multithreaded	YUV	HD 720p	1174.16	0.002
VQMT 14.0 default	multithreaded	YUV	FullHD 1080p	461.01	0.005
VQMT 14.0 default	multithreaded	YUV	4K 2160p	136.2	0.019
VQMT 14.0 default	singlethreaded	Y	HD 720p	135.22	0.008
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	62.2	0.016
VQMT 14.0 default	singlethreaded	Y	4K 2160p	22.67	0.045
VQMT 14.0 default	singlethreaded	YUV	HD 720p	107.83	0.009
VQMT 14.0 default	singlethreaded	YUV	FullHD 1080p	48.46	0.021
VQMT 14.0 default	singlethreaded	YUV	4K 2160p	17.85	0.058

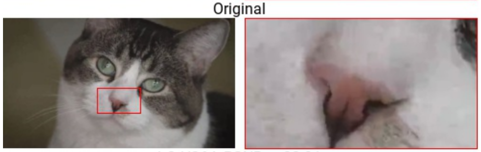
Benchmark on VQMT 13.1 r12792 PRO for Windows.

CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

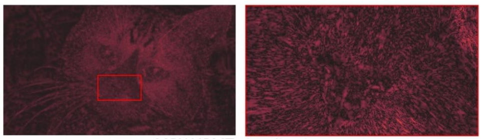
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 default	multithreaded	Y	HD 720p	563.49	0.003
VQMT 13.1 default	multithreaded	Y	FullHD 1080p	257.21	0.005
VQMT 13.1 default	multithreaded	Y	4K 2160p	54.08	0.02
VQMT 13.1 default	multithreaded	YUV	HD 720p	416.93	0.003
VQMT 13.1 default	multithreaded	YUV	FullHD 1080p	187.71	0.006
VQMT 13.1 default	multithreaded	YUV	4K 2160p	41.67	0.027
VQMT 13.1 default	singlethreaded	Y	HD 720p	181.1	0.006
VQMT 13.1 default	singlethreaded	Y	FullHD 1080p	88.76	0.012
VQMT 13.1 default	singlethreaded	Y	4K 2160p	23.1	0.045
VQMT 13.1 default	singlethreaded	YUV	HD 720p	135.52	0.008
VQMT 13.1 default	singlethreaded	YUV	FullHD 1080p	64.92	0.016
VQMT 13.1 default	singlethreaded	YUV	4K 2160p	22.18	0.048



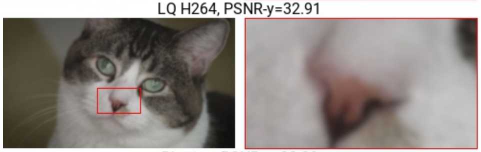
Original



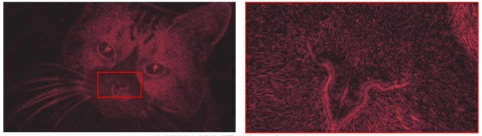
LQ H264, PSNR-y=32.91



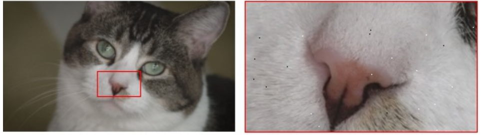
MSU VQMT visualization



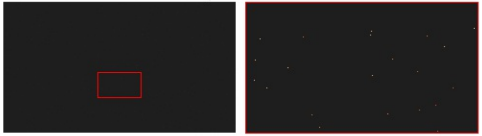
Blurring, PSNR-y=32.32



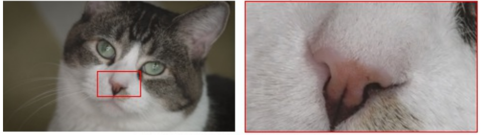
MSU VQMT visualization



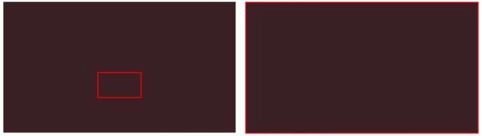
Random points, PSNR-y=37.91



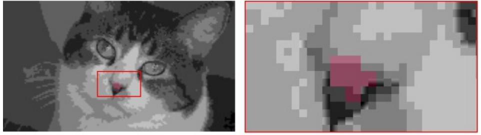
MSU VQMT visualization



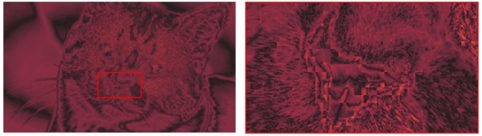
Luminance shift, PSNR-y=42.11



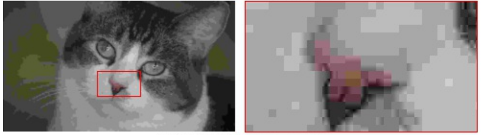
MSU VQMT visualization



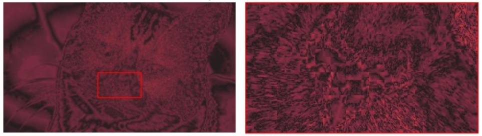
JPEG Q=2, PSNR-y=26.65



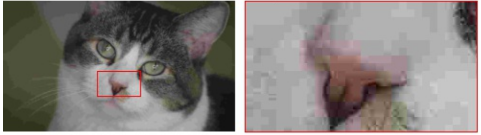
MSU VQMT visualization



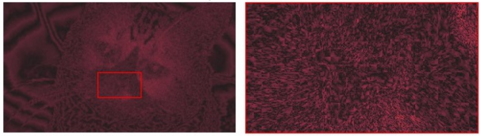
JPEG Q=5, PSNR-y=29.47



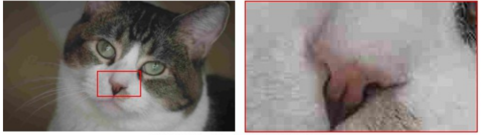
MSU VQMT visualization



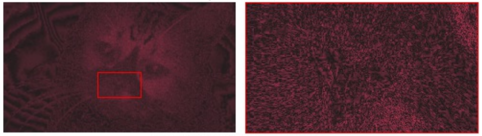
JPEG Q=10, PSNR-y=32.97



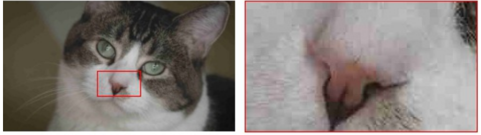
MSU VQMT visualization



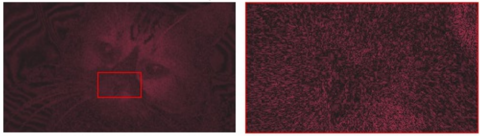
JPEG Q=15, PSNR-y=34.72



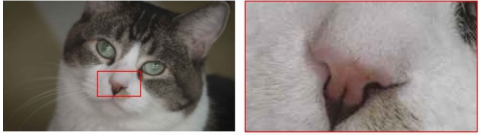
MSU VQMT visualization



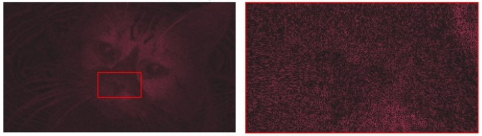
JPEG Q=20, PSNR-y=35.99



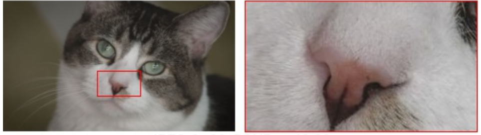
MSU VQMT visualization



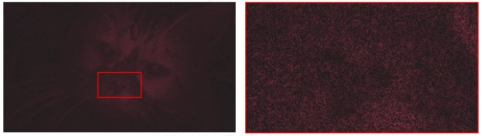
JPEG Q=40, PSNR-y=38.82



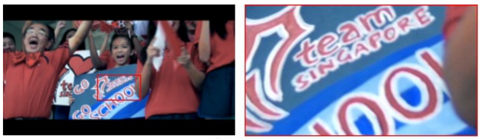
MSU VQMT visualization



JPEG Q=80, PSNR-y=43.29

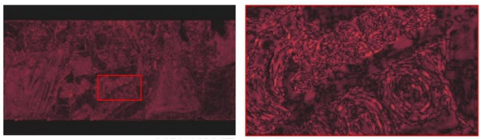


MSU VQMT visualization

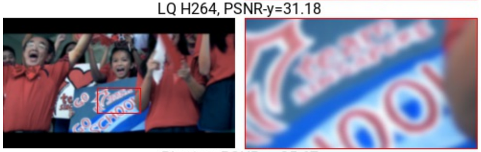


Original

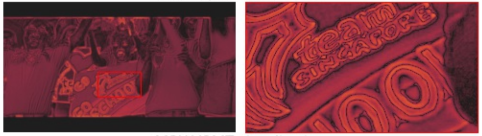
LQ H264, PSNR-y=31.18



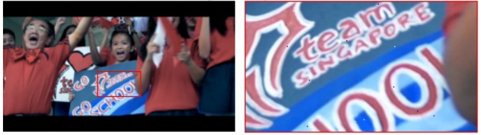
MSU VQMT visualization



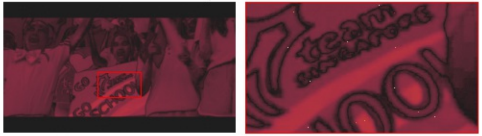
Blurring, PSNR-y=25.67



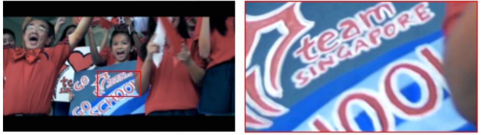
MSU VQMT visualization



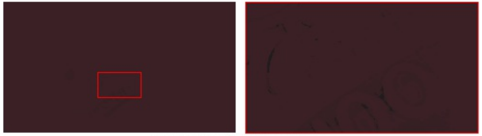
Random points, PSNR-y=29.20



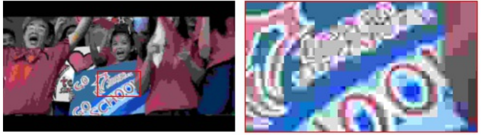
MSU VQMT visualization



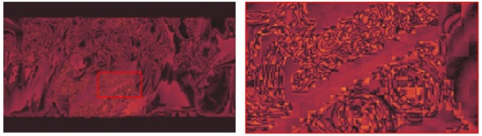
Luminance shift, PSNR-y=42.11



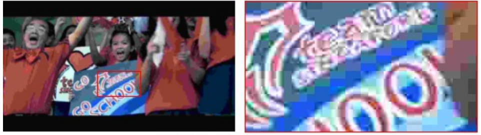
MSU VQMT visualization



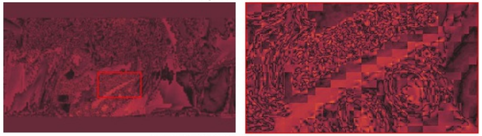
JPEG Q=2, PSNR-y=25.05



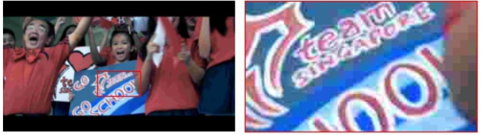
MSU VQMT visualization



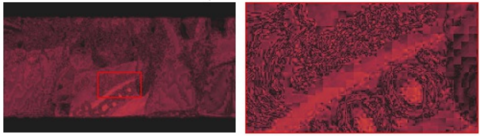
JPEG Q=5, PSNR-y=26.58



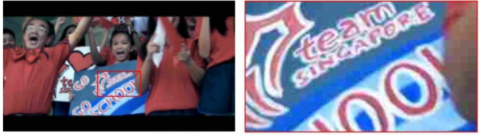
MSU VQMT visualization



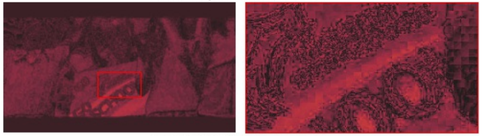
JPEG Q=10, PSNR-y=28.67



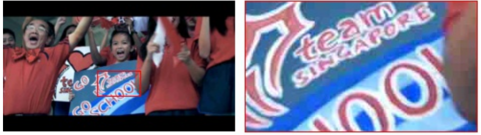
MSU VQMT visualization



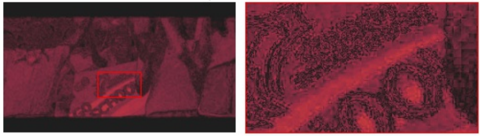
JPEG Q=15, PSNR-y=29.10



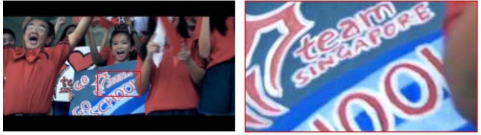
MSU VQMT visualization



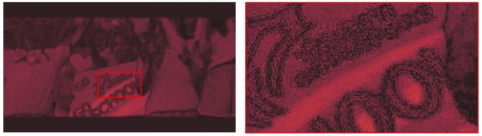
JPEG Q=20, PSNR-y=29.32



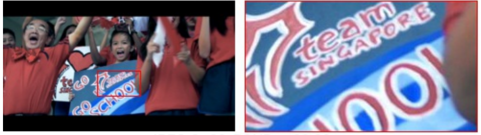
MSU VQMT visualization



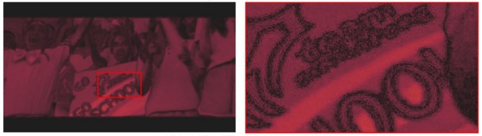
JPEG Q=40, PSNR-y=29.67



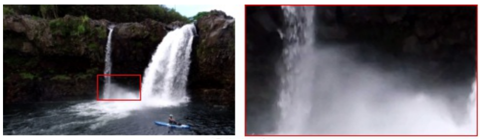
MSU VQMT visualization



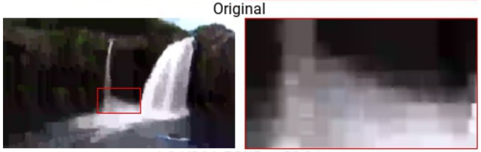
JPEG Q=80, PSNR-y=29.89



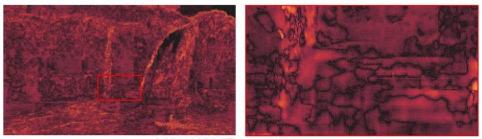
MSU VQMT visualization



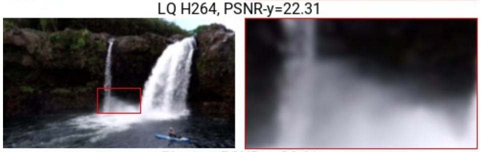
Original



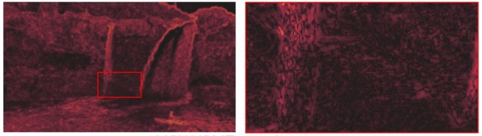
LQ H264, PSNR-y=22.31



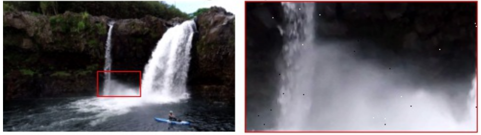
MSU VQMT visualization



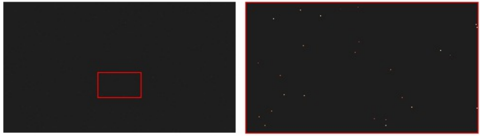
Blurring, PSNR-y=26.61



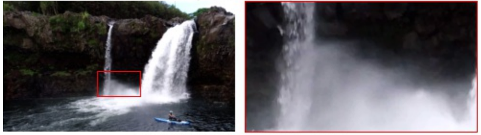
MSU VQMT visualization



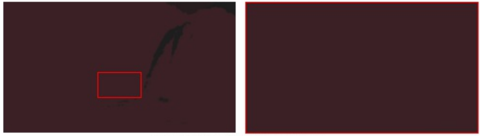
Random points, PSNR-y=36.74



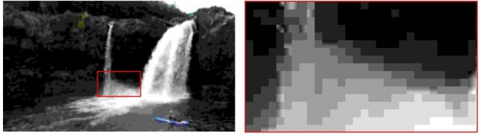
MSU VQMT visualization



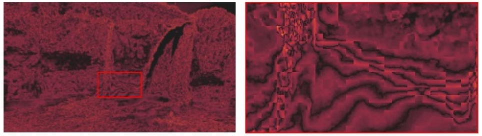
Luminance shift, PSNR-y=42.29



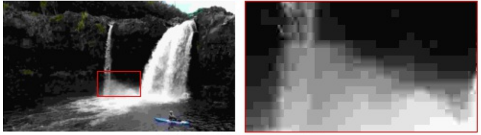
MSU VQMT visualization



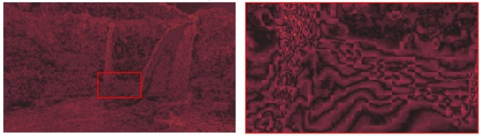
JPEG Q=2, PSNR-y=24.60



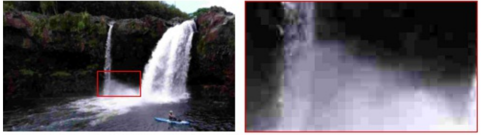
MSU VQMT visualization



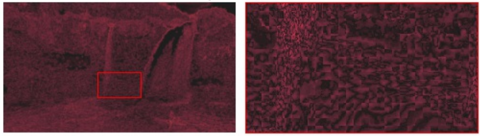
JPEG Q=5, PSNR-y=27.04



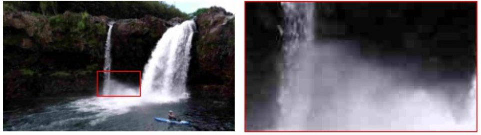
MSU VQMT visualization



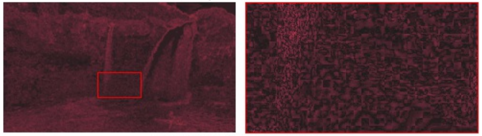
JPEG Q=10, PSNR-y=29.75



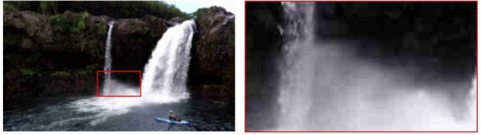
MSU VQMT visualization



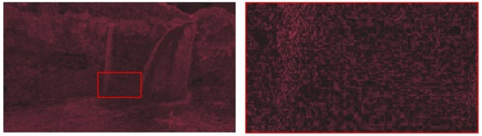
JPEG Q=15, PSNR-y=31.31



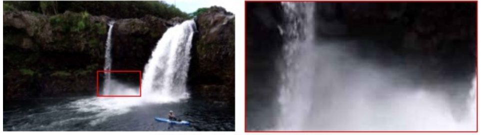
MSU VQMT visualization



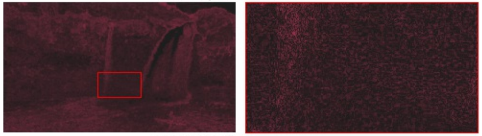
JPEG Q=20, PSNR-y=32.32



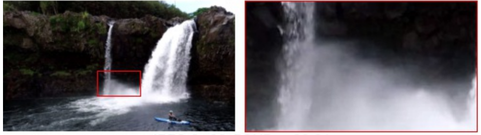
MSU VQMT visualization



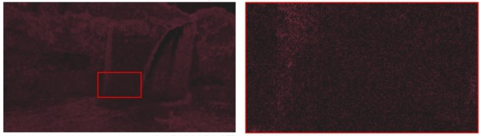
JPEG Q=40, PSNR-y=34.79



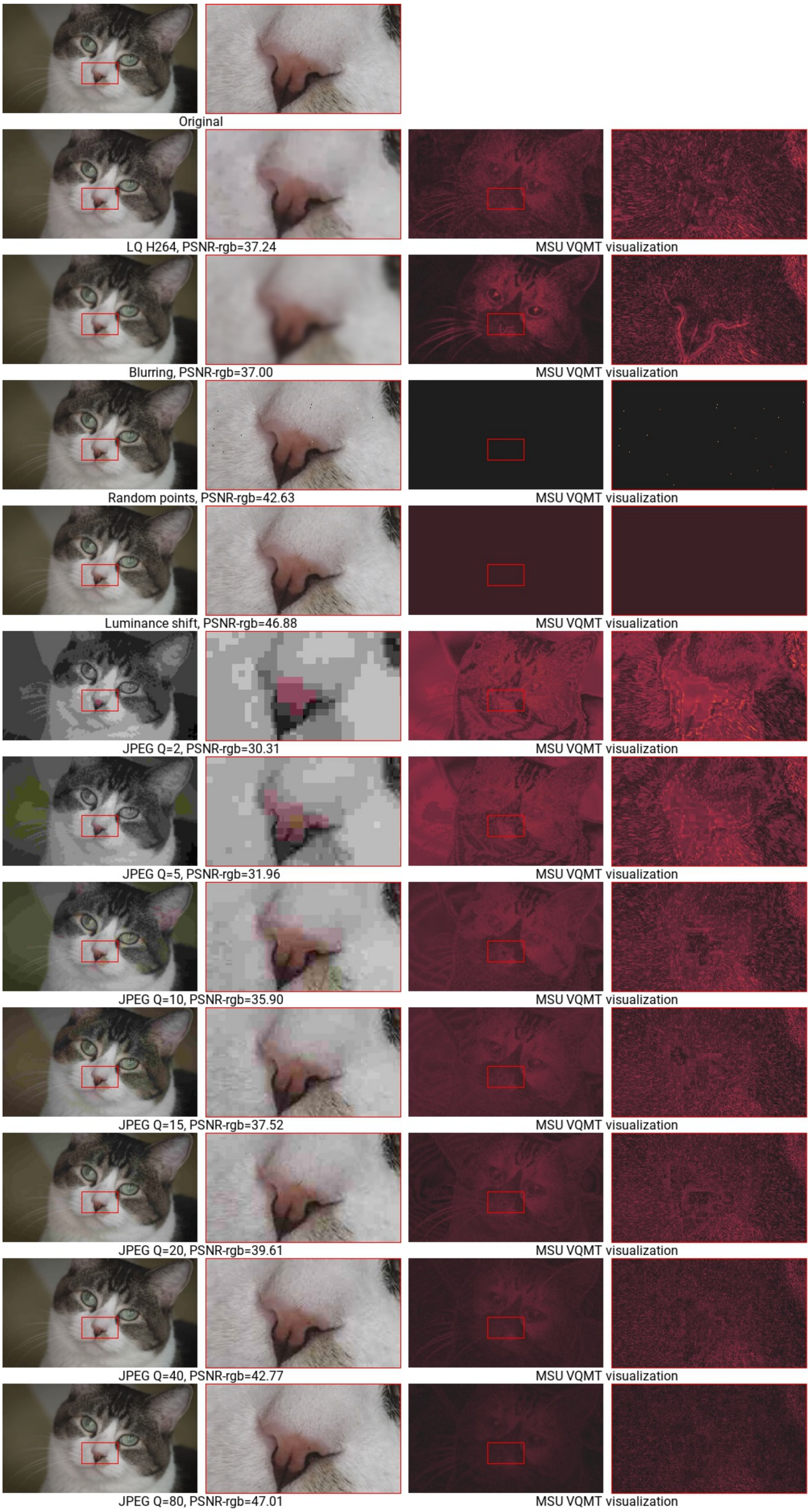
MSU VQMT visualization



JPEG Q=80, PSNR-y=39.21



MSU VQMT visualization



Original

LQ H264, PSNR-rgb=37.24

MSU VQMT visualization

Blurring, PSNR-rgb=37.00

MSU VQMT visualization

Random points, PSNR-rgb=42.63

MSU VQMT visualization

Luminance shift, PSNR-rgb=46.88

MSU VQMT visualization

JPEG Q=2, PSNR-rgb=30.31

MSU VQMT visualization

JPEG Q=5, PSNR-rgb=31.96

MSU VQMT visualization

JPEG Q=10, PSNR-rgb=35.90

MSU VQMT visualization

JPEG Q=15, PSNR-rgb=37.52

MSU VQMT visualization

JPEG Q=20, PSNR-rgb=39.61

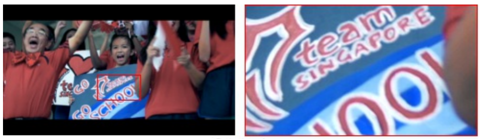
MSU VQMT visualization

JPEG Q=40, PSNR-rgb=42.77

MSU VQMT visualization

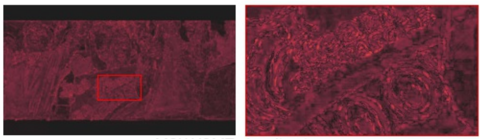
JPEG Q=80, PSNR-rgb=47.01

MSU VQMT visualization

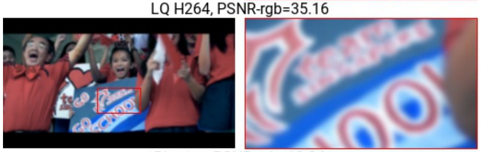


Original

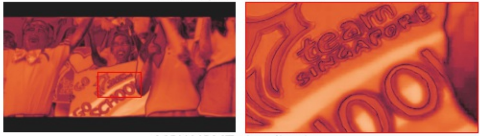
LQ H264, PSNR-rgb=35.16



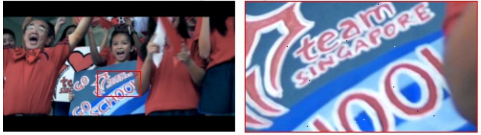
MSU VQMT visualization



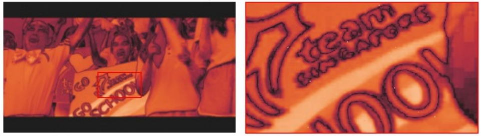
Blurring, PSNR-rgb=19.34



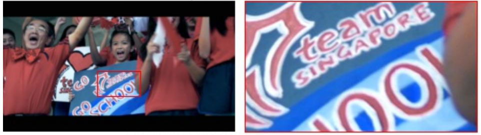
MSU VQMT visualization



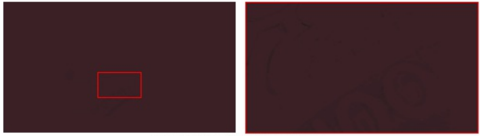
Random points, PSNR-rgb=19.39



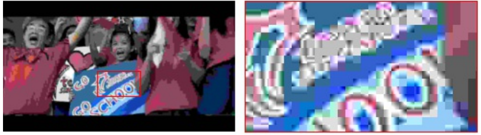
MSU VQMT visualization



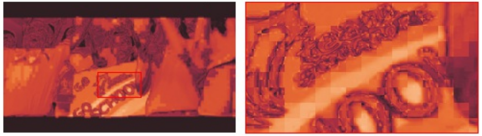
Luminance shift, PSNR-rgb=46.87



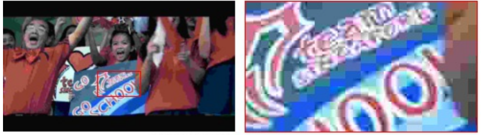
MSU VQMT visualization



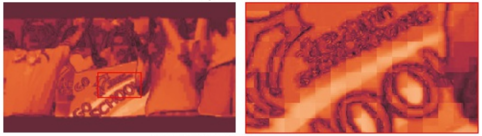
JPEG Q=2, PSNR-rgb=19.87



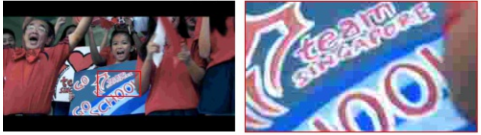
MSU VQMT visualization



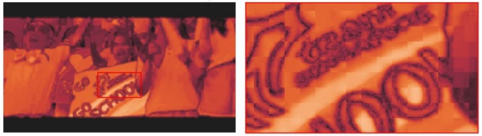
JPEG Q=5, PSNR-rgb=19.28



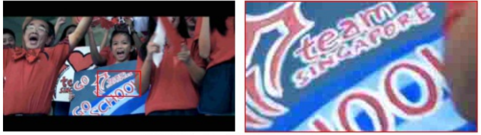
MSU VQMT visualization



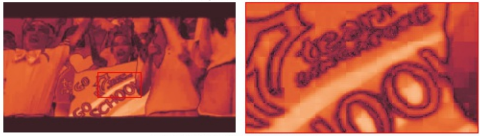
JPEG Q=10, PSNR-rgb=19.46



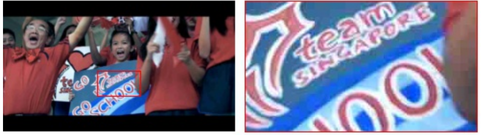
MSU VQMT visualization



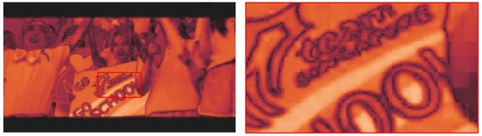
JPEG Q=15, PSNR-rgb=19.43



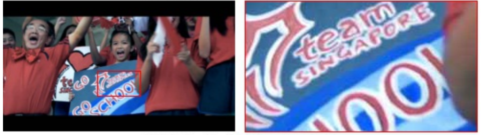
MSU VQMT visualization



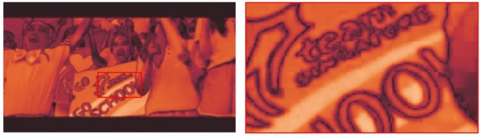
JPEG Q=20, PSNR-rgb=19.45



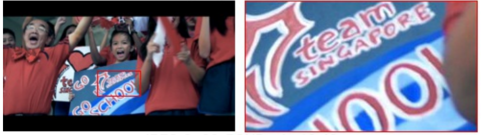
MSU VQMT visualization



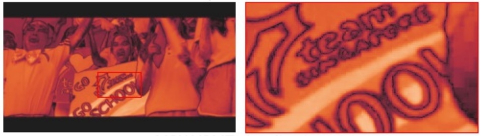
JPEG Q=40, PSNR-rgb=19.44



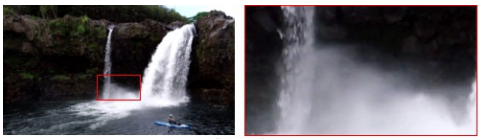
MSU VQMT visualization



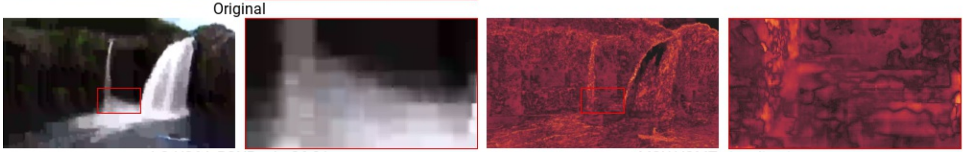
JPEG Q=80, PSNR-rgb=19.42



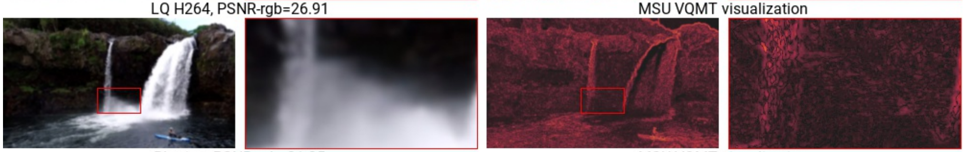
MSU VQMT visualization



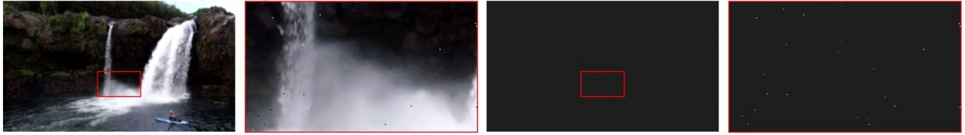
Original



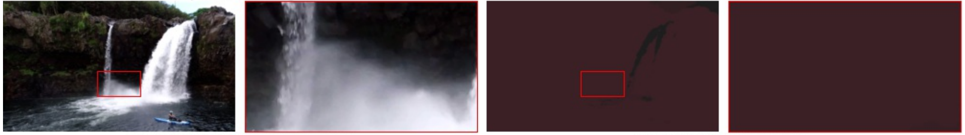
LQ H264, PSNR-rgb=26.91 MSU VQMT visualization



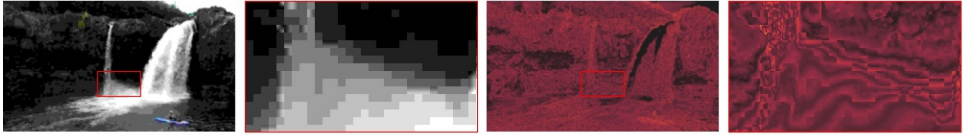
Blurring, PSNR-rgb=31.35 MSU VQMT visualization



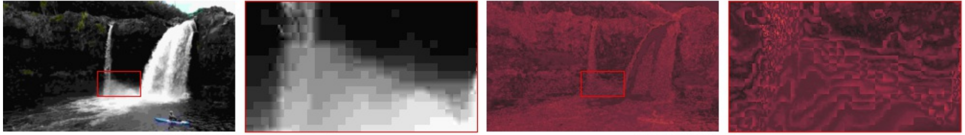
Random points, PSNR-rgb=41.50 MSU VQMT visualization



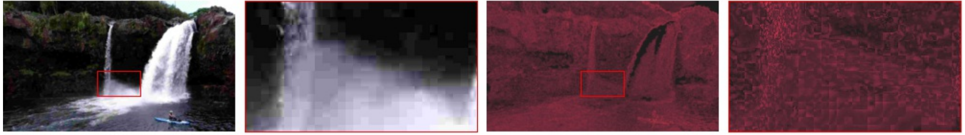
Luminance shift, PSNR-rgb=47.06 MSU VQMT visualization



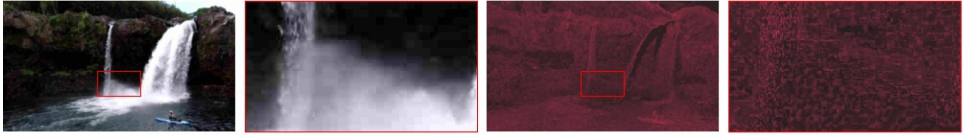
JPEG Q=2, PSNR-rgb=28.74 MSU VQMT visualization



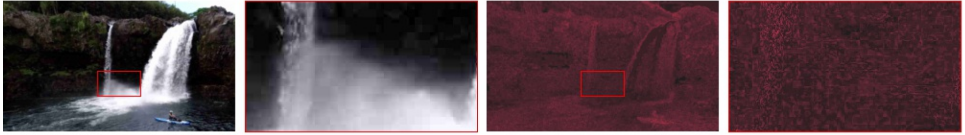
JPEG Q=5, PSNR-rgb=30.87 MSU VQMT visualization



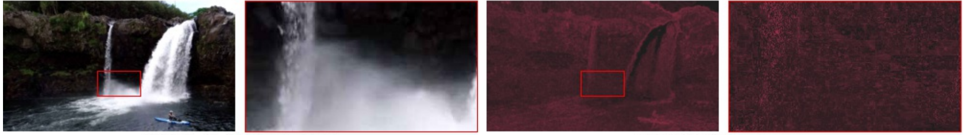
JPEG Q=10, PSNR-rgb=33.43 MSU VQMT visualization



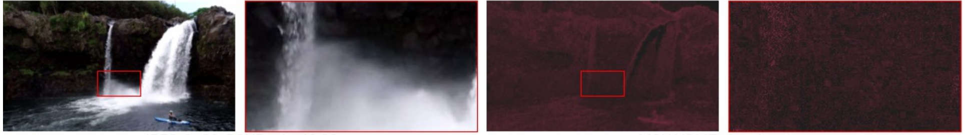
JPEG Q=15, PSNR-rgb=35.30 MSU VQMT visualization



JPEG Q=20, PSNR-rgb=36.45 MSU VQMT visualization



JPEG Q=40, PSNR-rgb=38.96 MSU VQMT visualization



JPEG Q=80, PSNR-rgb=43.23 MSU VQMT visualization

Netflix VMAF (Video Multimethod Assessment Fusion)

General info

Metric type	full-reference temporal metric
Value range	(completely different) 0..100 (similar to original) -∞..∞ if truncation to 0..100 is off *dependent on model* in case of custom model
Value interpretation	bigger better quality, value 100 does not mean that the images match pixel by pixel
MSU VQMT implementations	CPU multithreaded OpenCL (since VQMT 13)
MSU VQMT visualization	block-wise (for VMAF visualization), pixel-wise (for ADM, VIF, ANSNR visualisation)
Available colorspaces	Y
Output values	metric value, bagging values (if on), confidence intervals (if on), values of elementary features (if on)
Aggregated values	standard set
MSU VQMT usages	-metr vmaf [-dev <OpenCL device>]
External links	Original paper

Algorithm description

VMAF is modern reference metric developed by Netflix in cooperation with the University of Southern California. VQMT has full support of VMAF with multiple configuration switches. MSU VQMT support the following VMAF models: VMAF 0.60, VMAF 0.61 (2k, 4k), VMAF 0.62 (2k, 4k), VMAF 0.63 (2k), also, you can compute phone model and elementary features of VMAF. You can use custom model in pkl format with VQMT.

VMAF consist of 4 features (ADM, VIF, Motion, ANSNR) and 35 elementary features, but VMAF models uses only 6 of them: adm2, motion2, vif_scale0, vif_scale1, vif_scale2, vif_scale3. VMAF applies an SVM model to this set of features, which depends on current settings. After applying SVM, the value is clipped to interval 0..100 by default. Motion feature is the only temporal feature, it consider adjacent frames. To calculate VMAF value for current frame it is needed to use the previous frame and the next frame.

VMAF can also compute confidence intervals by applying multiple models and calculating standard deviation of result. VMAF has models, that aimed for 4k and 2k. By default, VQMT will automatically select the correct model by the resolution of input video.

Since VQMT 13 VMAF has a real block-wise visualization, which computes individual VMAF value for each 16x16 block of image. You also can visualize every feature besides motion (ADM, VIF, ANSNR).

Benchmark

Please note: values can vary depending on system configuration, input format and other factors.

Benchmark on VQMT 14.0 BETA r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 OpenCL	OpenCL	Y	HD 720p	78.35	0.014
VQMT 14.0 OpenCL	OpenCL	Y	FullHD 1080p	38.95	0.028
VQMT 14.0 OpenCL	OpenCL	Y	4K 2160p	9.94	0.113
VQMT 14.0 default	multithreaded	Y	HD 720p	17.68	0.06
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	8.03	0.134
VQMT 14.0 default	multithreaded	Y	4K 2160p	1.86	0.616
VQMT 14.0 default	singlethreaded	Y	HD 720p	5.93	0.177
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	2.78	0.38
VQMT 14.0 default	singlethreaded	Y	4K 2160p	0.66	1.567

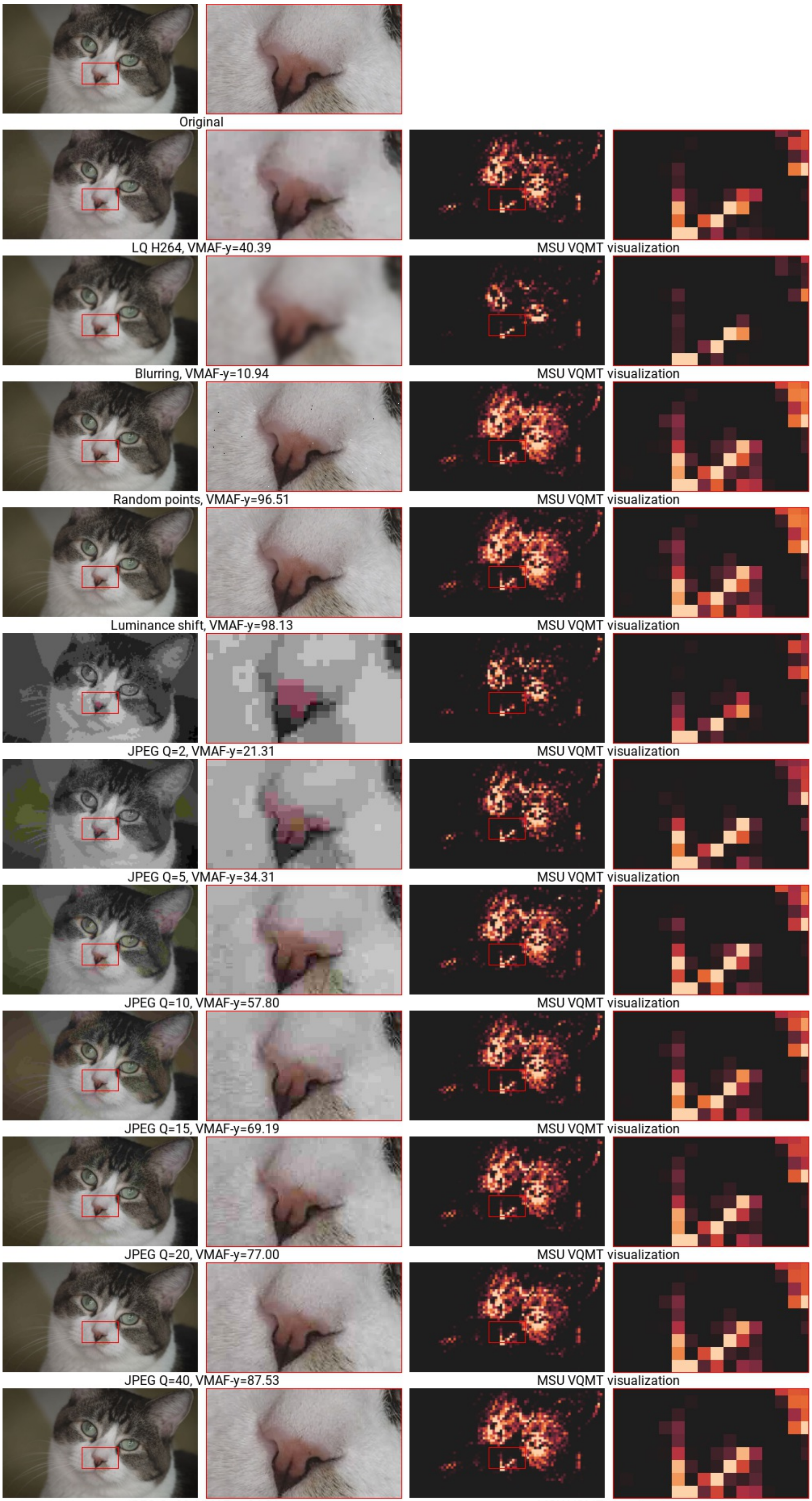
**Benchmark on VQMT 14.0 BETA r12792 PRO for Linux.
CPU: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64 cores
GPU: NVIDIA CUDA/TITAN RTX**

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 OpenCL	OpenCL	Y	HD 720p	291.17	0.005
VQMT 14.0 OpenCL	OpenCL	Y	FullHD 1080p	132.3	0.011
VQMT 14.0 OpenCL	OpenCL	Y	4K 2160p	32.23	0.048
VQMT 14.0 default	multithreaded	Y	HD 720p	192.66	0.007
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	42.9	0.032
VQMT 14.0 default	multithreaded	Y	4K 2160p	10.34	0.189
VQMT 14.0 default	singlethreaded	Y	HD 720p	6.56	0.157
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	3.01	0.339
VQMT 14.0 default	singlethreaded	Y	4K 2160p	0.8	1.267

**Benchmark on VQMT 13.1 r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti**

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 OpenCL	OpenCL	Y	HD 720p	80.5	0.013
VQMT 13.1 OpenCL	OpenCL	Y	FullHD 1080p	40.25	0.027
VQMT 13.1 OpenCL	OpenCL	Y	4K 2160p	10.12	0.11
VQMT 13.1 default	multithreaded	Y	HD 720p	17.96	0.058
VQMT 13.1 default	multithreaded	Y	FullHD 1080p	8.03	0.132

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 default	multithreaded	Y	4K 2160p	1.91	0.593
VQMT 13.1 default	singlethreaded	Y	HD 720p	6.23	0.166
VQMT 13.1 default	singlethreaded	Y	FullHD 1080p	2.91	0.356
VQMT 13.1 default	singlethreaded	Y	4K 2160p	0.69	1.484



Original

LQ H264, VMAF-y=40.39

MSU VQMT visualization

Blurring, VMAF-y=10.94

MSU VQMT visualization

Random points, VMAF-y=96.51

MSU VQMT visualization

Luminance shift, VMAF-y=98.13

MSU VQMT visualization

JPEG Q=2, VMAF-y=21.31

MSU VQMT visualization

JPEG Q=5, VMAF-y=34.31

MSU VQMT visualization

JPEG Q=10, VMAF-y=57.80

MSU VQMT visualization

JPEG Q=15, VMAF-y=69.19

MSU VQMT visualization

JPEG Q=20, VMAF-y=77.00

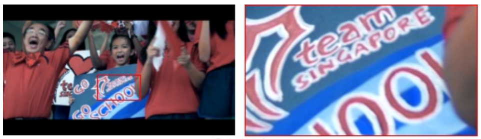
MSU VQMT visualization

JPEG Q=40, VMAF-y=87.53

MSU VQMT visualization

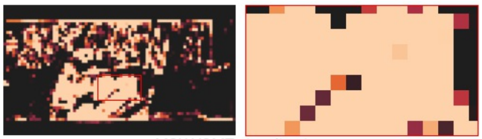
JPEG Q=80, VMAF-y=94.31

MSU VQMT visualization

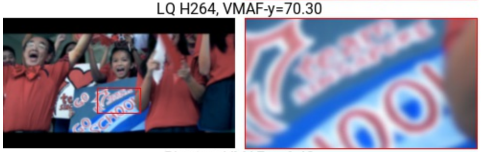


Original

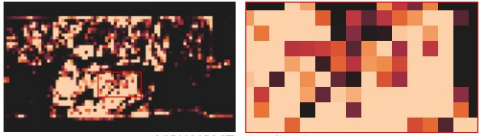
LQ H264, VMAF-y=70.30



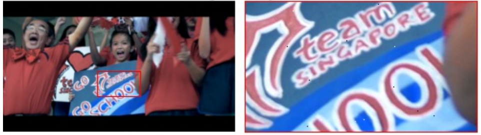
MSU VQMT visualization



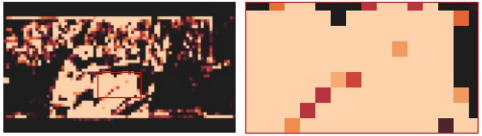
Blurring, VMAF-y=6.43



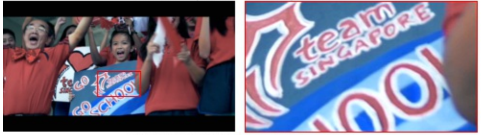
MSU VQMT visualization



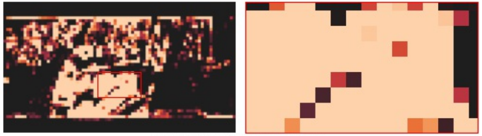
Random points, VMAF-y=95.32



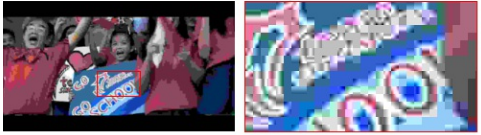
MSU VQMT visualization



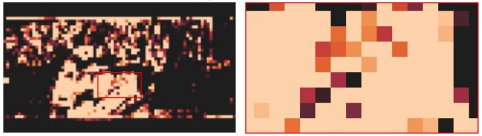
Luminance shift, VMAF-y=100.00



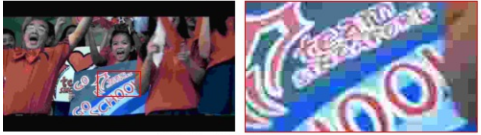
MSU VQMT visualization



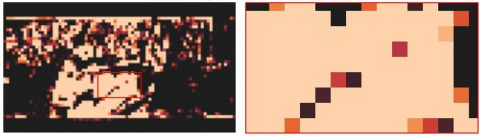
JPEG Q=2, VMAF-y=48.68



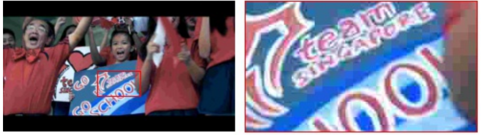
MSU VQMT visualization



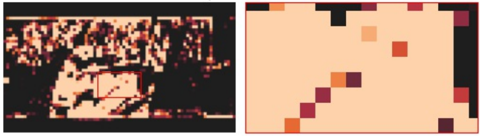
JPEG Q=5, VMAF-y=60.95



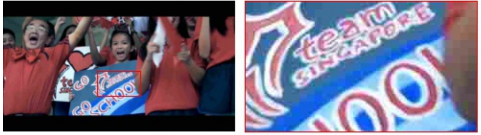
MSU VQMT visualization



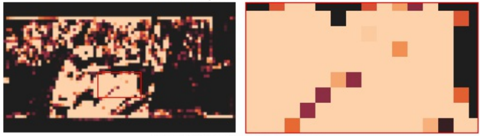
JPEG Q=10, VMAF-y=76.68



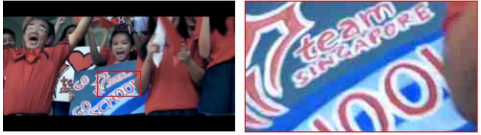
MSU VQMT visualization



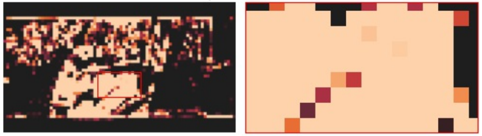
JPEG Q=15, VMAF-y=82.54



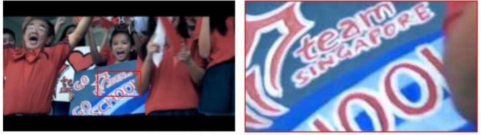
MSU VQMT visualization



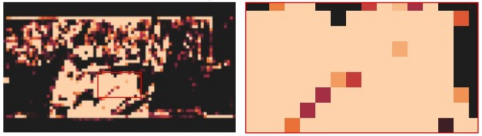
JPEG Q=20, VMAF-y=85.99



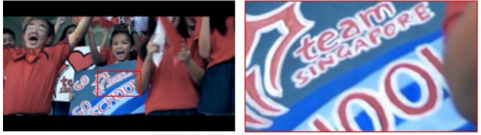
MSU VQMT visualization



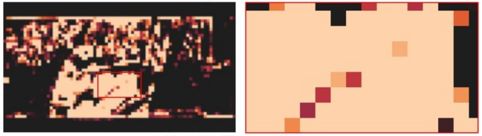
JPEG Q=40, VMAF-y=90.84



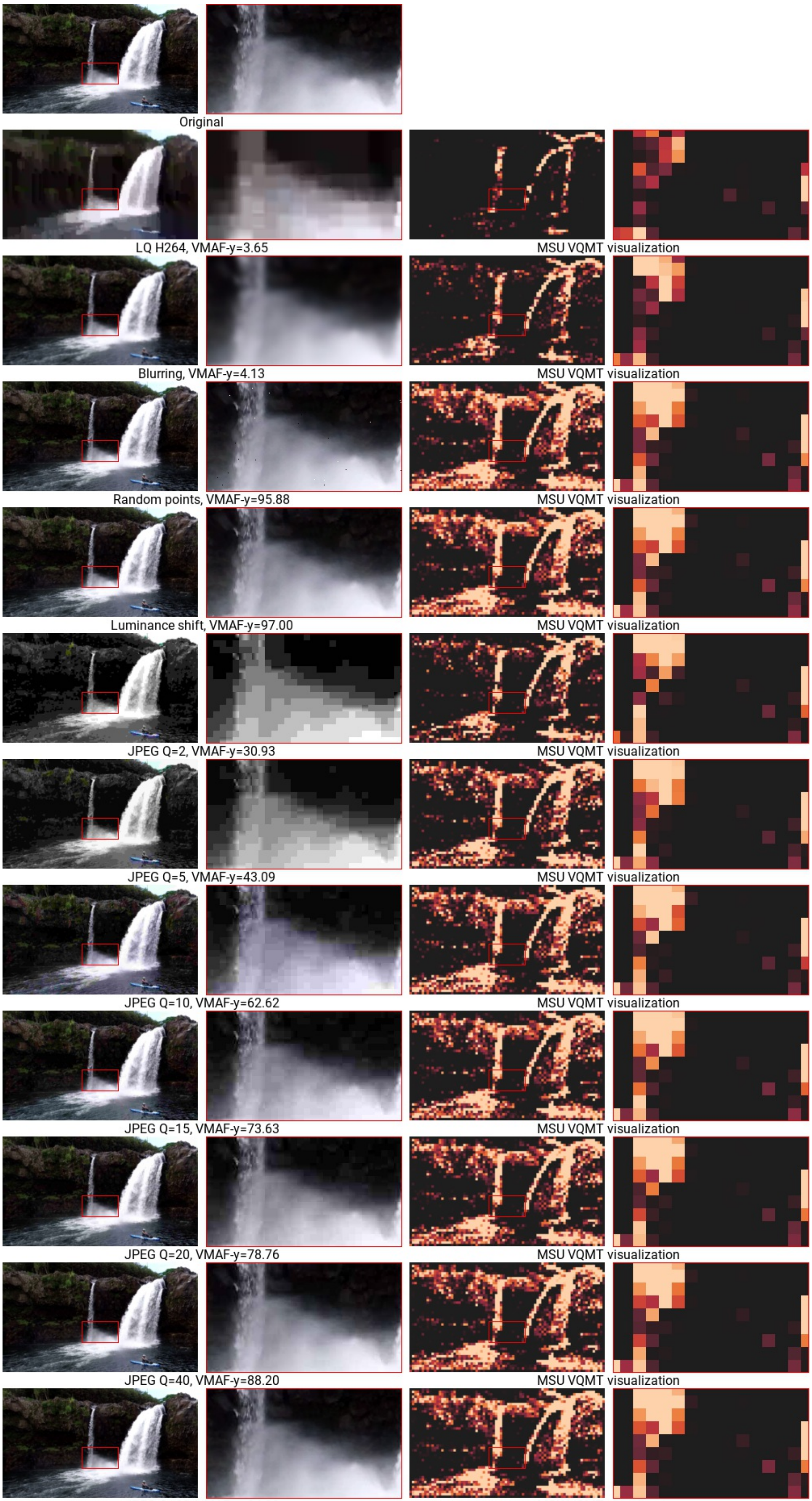
MSU VQMT visualization



JPEG Q=80, VMAF-y=93.91



MSU VQMT visualization



Original

LQ H264, VMAF-y=3.65

MSU VQMT visualization

Blurring, VMAF-y=4.13

MSU VQMT visualization

Random points, VMAF-y=95.88

MSU VQMT visualization

Luminance shift, VMAF-y=97.00

MSU VQMT visualization

JPEG Q=2, VMAF-y=30.93

MSU VQMT visualization

JPEG Q=5, VMAF-y=43.09

MSU VQMT visualization

JPEG Q=10, VMAF-y=62.62

MSU VQMT visualization

JPEG Q=15, VMAF-y=73.63

MSU VQMT visualization

JPEG Q=20, VMAF-y=78.76

MSU VQMT visualization

JPEG Q=40, VMAF-y=88.20

MSU VQMT visualization

JPEG Q=80, VMAF-y=94.32

MSU VQMT visualization

NIQE (Naturalness Image Quality Evaluator)

General info

Metric type	no-reference image metric
Value range	(very natural) 0..∞ (not natural)
Value interpretation	lesser metric values - better quality (naturalness) normal metric values are in range about 3..20. Also, there can be values NAN and 0, which are abnormal and should be considered as symptom of not natural image
MSU VQMT implementations	CPU multithreaded OpenCL (since VQMT 13)
MSU VQMT visualization	block-wise
Available colorspaces	Y
Output values	metric value
Aggregated values	standard set , NIQE mean
MSU VQMT usages	-metr niqe [-dev <OpenCL device>]
External links	original paper (A. Mittal, R. Soundararajan and A. C. Bovik), MSU paper

Algorithm description

NIQE performs feature extraction for every 96x96 block of image on 2 scales. Than it computes correlations of features between all blocks and applies leaned model. For more details, please refer to original paper. Since version 13 VQMT can build a block-wise visualization showing contribution of each block to the final result.

This metric has a special aggregated value NIQE mean, it takes a weighted mean filtering abnormal metric values and taking suspicies values with low weight. You can turn this mode using metric settings. Also, please see our paper.

This metric only applicable to filmed scenes. Metric can produce inadequate result if some graphics is on it (including credits, subtitles, etc.). Please, run NIQE excluding all rendered scenes, they can fatally spoil average value. Also, this metric can produce bad result on scenes containing noisy objects, like sand or grass, however on scenes with big constant areas, like monotonic sky. In common case normal metric results lies in the interval 3..20.

Sometimes, metric shows better result for the compressed image and this correlates with human perception. Compressed image not always is perceived as worse. It can occur for example in case of noisy images (if the noise has non-compression nature). This is only metric in VQMT now that can detect increasing of subjective quality in comparison to original.

Sometimes, codec can allow geometry transformation (like shift of heterogeneous objects in frame), that not critical for subjective perception. Objective-reference metrics are very perceptive to such transformation, and in this cases no-reference metric can show result closer to subjective score.

Benchmark

Please note: values can vary depending on system configuration, input format and other factors.

**Benchmark on VQMT 14.0 BETA r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores**

GPU: NVIDIA CUDA/GeForce GTX 660 Ti

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 OpenCL	OpenCL	Y	HD 720p	89.39	0.014
VQMT 14.0 OpenCL	OpenCL	Y	FullHD 1080p	41.94	0.027
VQMT 14.0 OpenCL	OpenCL	Y	4K 2160p	10.61	0.107
VQMT 14.0 default	multithreaded	Y	HD 720p	47.76	0.023
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	21.35	0.051
VQMT 14.0 default	multithreaded	Y	4K 2160p	4.83	0.243
VQMT 14.0 default	singlethreaded	Y	HD 720p	12.42	0.084
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	5.56	0.189
VQMT 14.0 default	singlethreaded	Y	4K 2160p	1.38	0.769

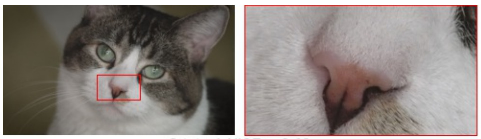
Benchmark on VQMT 14.0 BETA r12792 PRO for Linux.
CPU: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64 cores
GPU: NVIDIA CUDA/TITAN RTX

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 OpenCL	OpenCL	Y	HD 720p	4.29	0.293
VQMT 14.0 OpenCL	OpenCL	Y	FullHD 1080p	4.24	0.3
VQMT 14.0 OpenCL	OpenCL	Y	4K 2160p	4.22	0.325
VQMT 14.0 default	multithreaded	Y	HD 720p	136.31	0.009
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	38.66	0.034
VQMT 14.0 default	multithreaded	Y	4K 2160p	5.21	0.279
VQMT 14.0 default	singlethreaded	Y	HD 720p	14.17	0.071
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	7.67	0.131
VQMT 14.0 default	singlethreaded	Y	4K 2160p	2.19	0.467

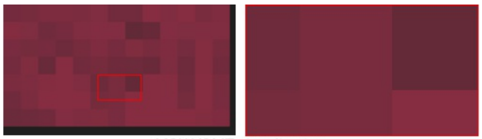
Benchmark on VQMT 13.1 r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 OpenCL	OpenCL	Y	HD 720p	88.87	0.014
VQMT 13.1 OpenCL	OpenCL	Y	FullHD 1080p	42.24	0.027
VQMT 13.1 OpenCL	OpenCL	Y	4K 2160p	10.68	0.104
VQMT 13.1 default	multithreaded	Y	HD 720p	47.71	0.022

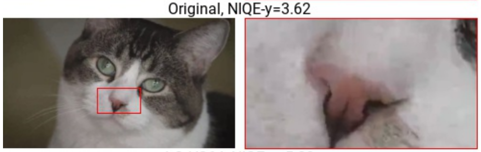
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 default	multithreaded	Y	FullHD 1080p	21.1	0.051
VQMT 13.1 default	multithreaded	Y	4K 2160p	4.8	0.241
VQMT 13.1 default	singlethreaded	Y	HD 720p	12.46	0.083
VQMT 13.1 default	singlethreaded	Y	FullHD 1080p	5.55	0.186
VQMT 13.1 default	singlethreaded	Y	4K 2160p	1.37	0.762



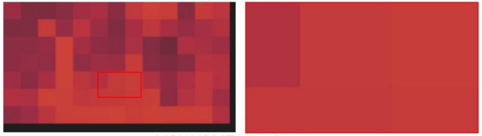
Original, NIQE-y=3.62



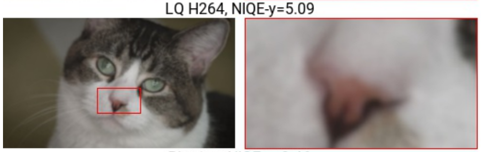
MSU VQMT visualization



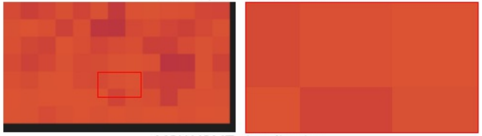
LQ H264, NIQE-y=5.09



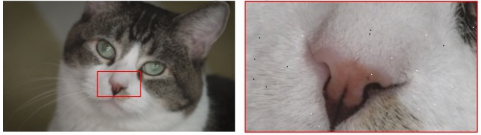
MSU VQMT visualization



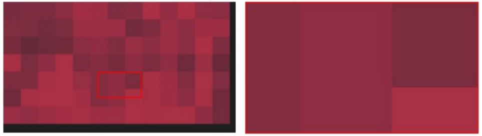
Blurring, NIQE-y=9.46



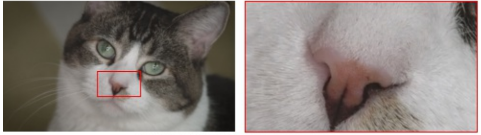
MSU VQMT visualization



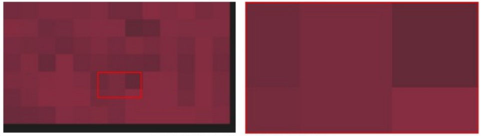
Random points, NIQE-y=3.86



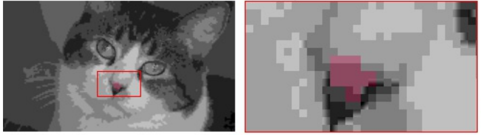
MSU VQMT visualization



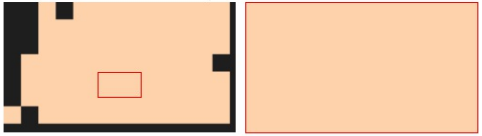
Luminance shift, NIQE-y=3.62



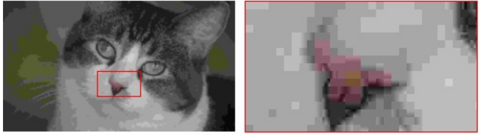
MSU VQMT visualization



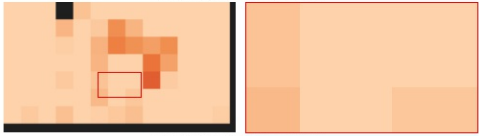
JPEG Q=2, NIQE-y=20.39



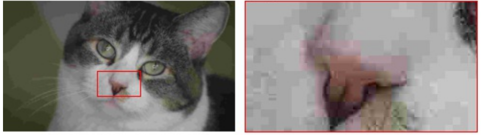
MSU VQMT visualization



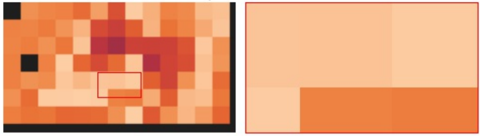
JPEG Q=5, NIQE-y=11.91



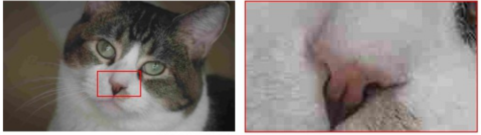
MSU VQMT visualization



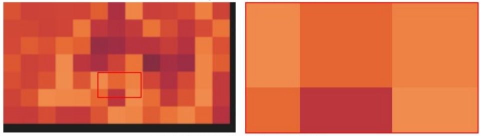
JPEG Q=10, NIQE-y=8.29



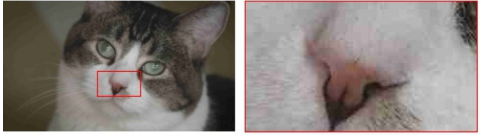
MSU VQMT visualization



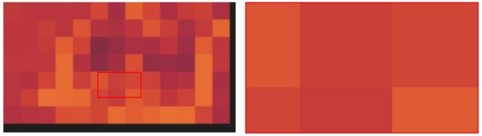
JPEG Q=15, NIQE-y=6.89



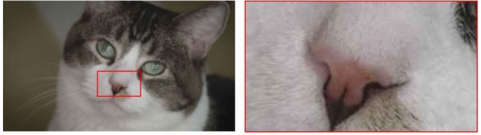
MSU VQMT visualization



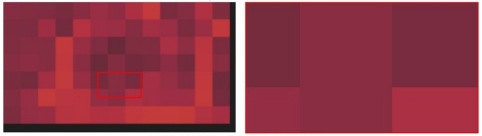
JPEG Q=20, NIQE-y=6.34



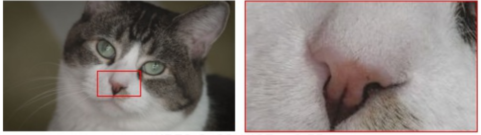
MSU VQMT visualization



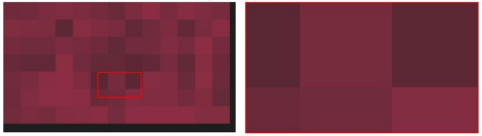
JPEG Q=40, NIQE-y=4.57



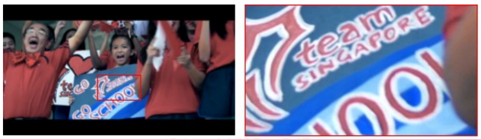
MSU VQMT visualization



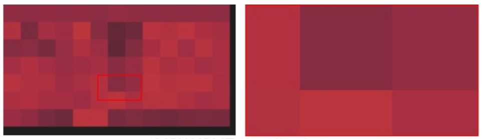
JPEG Q=80, NIQE-y=3.55



MSU VQMT visualization

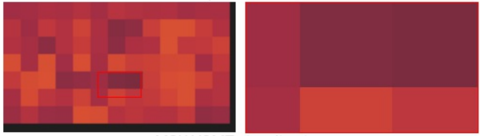


Original, NIQE-y=5.17

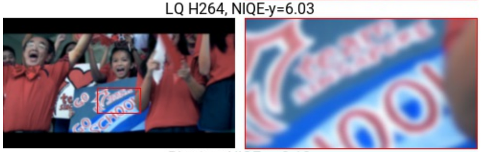


MSU VQMT visualization

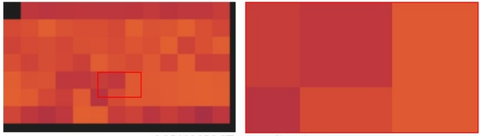
LQ H264, NIQE-y=6.03



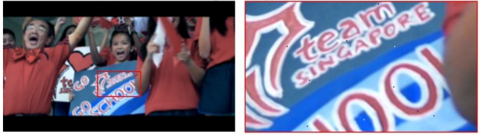
MSU VQMT visualization



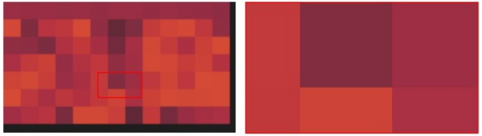
Blurring, NIQE-y=8.18



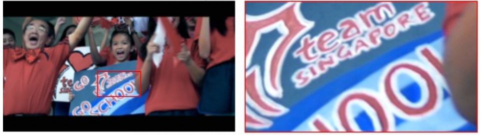
MSU VQMT visualization



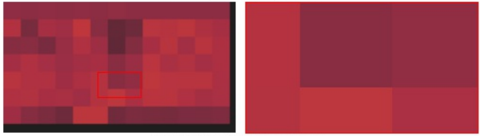
Random points, NIQE-y=5.46



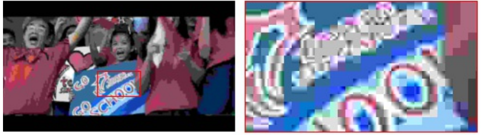
MSU VQMT visualization



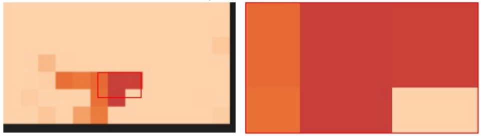
Luminance shift, NIQE-y=5.23



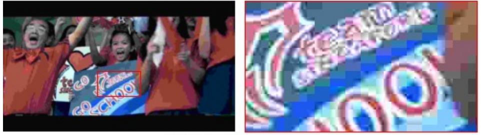
MSU VQMT visualization



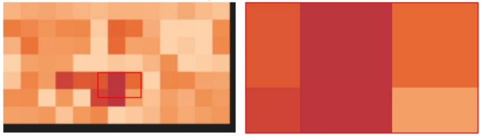
JPEG Q=2, NIQE-y=12.05



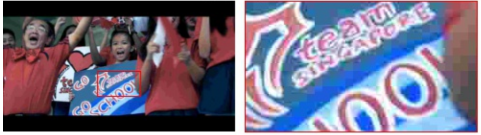
MSU VQMT visualization



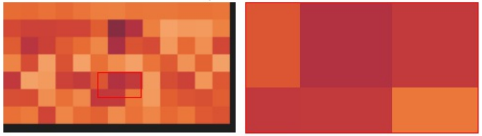
JPEG Q=5, NIQE-y=9.81



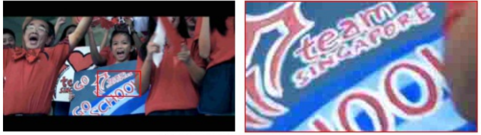
MSU VQMT visualization



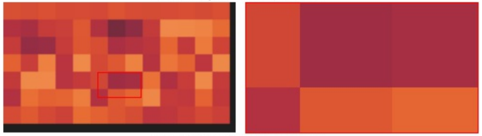
JPEG Q=10, NIQE-y=7.66



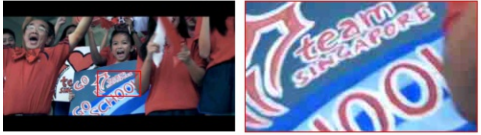
MSU VQMT visualization



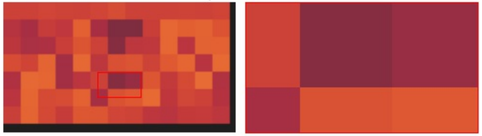
JPEG Q=15, NIQE-y=6.76



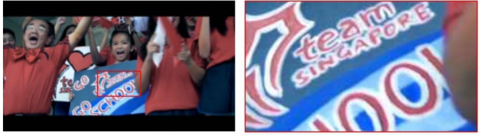
MSU VQMT visualization



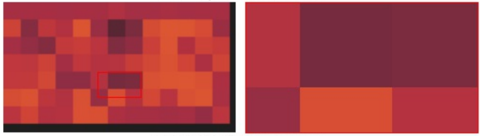
JPEG Q=20, NIQE-y=6.57



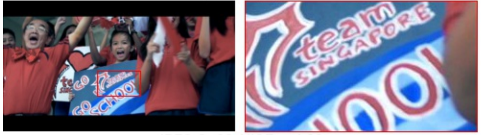
MSU VQMT visualization



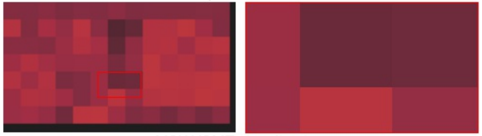
JPEG Q=40, NIQE-y=5.74



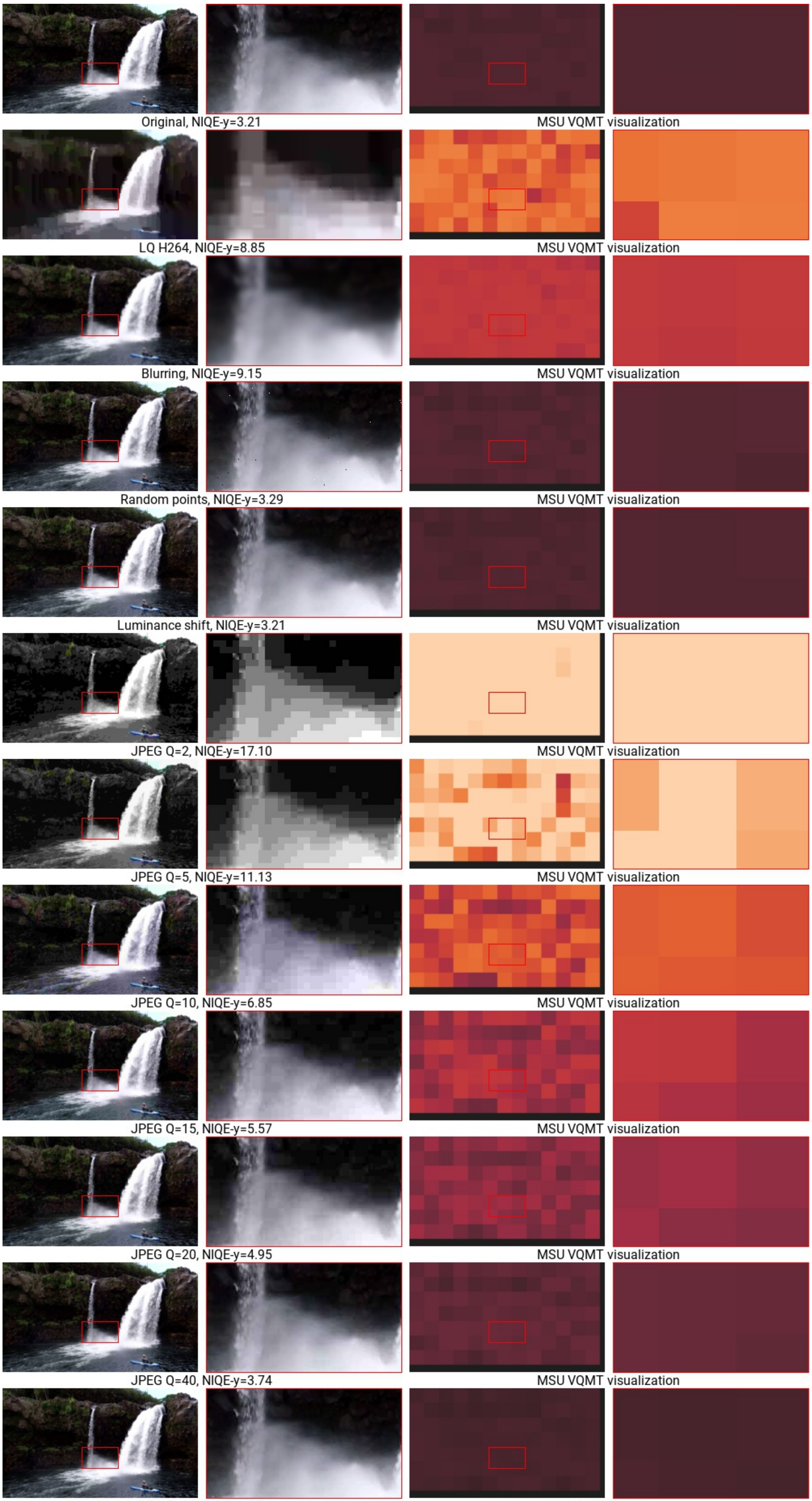
MSU VQMT visualization



JPEG Q=80, NIQE-y=4.88



MSU VQMT visualization



Original, NIQE-y=3.21 MSU VQMT visualization

LQ H264, NIQE-y=8.85 MSU VQMT visualization

Blurring, NIQE-y=9.15 MSU VQMT visualization

Random points, NIQE-y=3.29 MSU VQMT visualization

Luminance shift, NIQE-y=3.21 MSU VQMT visualization

JPEG Q=2, NIQE-y=17.10 MSU VQMT visualization

JPEG Q=5, NIQE-y=11.13 MSU VQMT visualization

JPEG Q=10, NIQE-y=6.85 MSU VQMT visualization

JPEG Q=15, NIQE-y=5.57 MSU VQMT visualization

JPEG Q=20, NIQE-y=4.95 MSU VQMT visualization

JPEG Q=40, NIQE-y=3.74 MSU VQMT visualization

JPEG Q=80, NIQE-y=2.76 MSU VQMT visualization

SSIM-Family

SSIM (Structural Similarity)

General info

Metric type	full-reference image metric
Value range	(images are different) -1..1 (images are same)
Value interpretation	bigger is better quality
MSU VQMT implementations	CPU multithreaded fast (default), cpu multithreaded precice, cpu multithreaded GPU identical, OpenCL (recommended), CUDA
MSU VQMT visualization	pixel-wise
Available colorspaces	Y, U, V, YUV, R, G, B, RGB
Output values	metric value
Aggregated values	standard set
MSU VQMT usages	-metr ssim [over <color components>] -metr ssim_precise [over <color components>] -metr ssim_gpu_id [over <color components>] -metr ssim_cuda [over <color components>] -metr ssim [over <color components>] -dev <OpenCL device>
External links	original paper (Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli)

Algorithm description

Main idea of the structure similarity index (SSIM) is to compare distortion of three image components:

- Luminance comparison
- Contrast comparison
- Structure comparison

This algorithm uses some window function $W_{i,j},\ i,j = 0..N$ and performs convolution with this window, defined as follows:

$$\langle U, W\rangle = \sum_{i=-R,j=-R}^{R,R} U(x+i, y+j)W_{i+R,j+R}, \text{ where } R = \frac{N}{2}$$

In fast implementation, it uses box window: $W_{i,j} = \frac{1}{(N+1)^2}$, in other implementations (precise, CUDA, OpenCL, GPU identical) it uses Gaussian window with σ= 1.5, N=10. You can note, that formula above uses negative U indexes, and indexes out of image area. We should define image values outside of it's edge. In VQMT we spread closest edge pixel to the desired position.

SSIM uses the following convolutions:

$$\mu_x = \langle X$$
$$\mu_y = \langle y$$
$$\sigma_x = \langle (\,$$
$$\sigma_y = \langle (\,$$
$$\sigma_{xy} = \langle (\,$$

And the computed SSIM in each pixel by the following formula:

$$\text{SSIM} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x + \sigma_y + C_2)},$$

where $C_1 = 0.01^2, C_2 = 0.03^2$. Destination metric value is arithmetic mean of SSIM values for each pixel. You also can see SSIM for each individual pixel on visualization.

GPU identical, *CUDA*, *OpenCL* implementations should produce very similar result. They use Gaussian window as presice implementation. Value of Presice implementation can differs from these ones.

Benchmark

Please note: values can vary depending on system configuration, input format and other factors.

Benchmark on VQMT 14.0 BETA r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 OpenCL	OpenCL	Y	HD 720p	328.18	0.004
VQMT 14.0 OpenCL	OpenCL	Y	FullHD 1080p	168.97	0.007
VQMT 14.0 OpenCL	OpenCL	Y	4K 2160p	47.15	0.024
VQMT 14.0 superfast	multithreaded	Y	HD 720p	547.15	0.003
VQMT 14.0 superfast	multithreaded	Y	FullHD 1080p	268.06	0.005
VQMT 14.0 superfast	multithreaded	Y	4K 2160p	57.31	0.02
VQMT 14.0 fast	multithreaded	Y	HD 720p	207.4	0.006
VQMT 14.0 fast	multithreaded	Y	FullHD 1080p	93.27	0.012
VQMT 14.0 fast	multithreaded	Y	4K 2160p	22.23	0.052
VQMT 14.0 precise	multithreaded	Y	HD 720p	114.37	0.01
VQMT 14.0 precise	multithreaded	Y	FullHD 1080p	51.73	0.021
VQMT 14.0 precise	multithreaded	Y	4K 2160p	12.52	0.097
VQMT 14.0 superfast	singlethreaded	Y	HD 720p	167.23	0.007

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 superfast	singlethreaded	Y	FullHD 1080p	82.57	0.013
VQMT 14.0 superfast	singlethreaded	Y	4K 2160p	22.67	0.046
VQMT 14.0 fast	singlethreaded	Y	HD 720p	49.93	0.021
VQMT 14.0 fast	singlethreaded	Y	FullHD 1080p	24.23	0.043
VQMT 14.0 fast	singlethreaded	Y	4K 2160p	6.28	0.171
VQMT 14.0 precise	singlethreaded	Y	HD 720p	44.45	0.024
VQMT 14.0 precise	singlethreaded	Y	FullHD 1080p	21.53	0.049
VQMT 14.0 precise	singlethreaded	Y	4K 2160p	5.53	0.195
VQMT 14.0 OpenCL	OpenCL	YUV	HD 720p	139.09	0.008
VQMT 14.0 OpenCL	OpenCL	YUV	FullHD 1080p	72.87	0.015
VQMT 14.0 OpenCL	OpenCL	YUV	4K 2160p	20.67	0.057
VQMT 14.0 superfast	multithreaded	YUV	HD 720p	394.5	0.004
VQMT 14.0 superfast	multithreaded	YUV	FullHD 1080p	187.15	0.007
VQMT 14.0 superfast	multithreaded	YUV	4K 2160p	42.5	0.027
VQMT 14.0 fast	multithreaded	YUV	HD 720p	83.48	0.013
VQMT 14.0 fast	multithreaded	YUV	FullHD 1080p	37.32	0.03
VQMT 14.0 fast	multithreaded	YUV	4K 2160p	8.29	0.153
VQMT 14.0 precise	multithreaded	YUV	HD 720p	42.91	0.025
VQMT 14.0 precise	multithreaded	YUV	FullHD 1080p	19.23	0.058
VQMT 14.0 precise	multithreaded	YUV	4K 2160p	4.68	0.275
VQMT 14.0 superfast	singlethreaded	YUV	HD 720p	117.32	0.009
VQMT 14.0 superfast	singlethreaded	YUV	FullHD 1080p	56.61	0.019
VQMT 14.0 superfast	singlethreaded	YUV	4K 2160p	18.8	0.057
VQMT 14.0 fast	singlethreaded	YUV	HD 720p	20.39	0.051
VQMT 14.0 fast	singlethreaded	YUV	FullHD 1080p	9.15	0.115
VQMT 14.0 fast	singlethreaded	YUV	4K 2160p	2.29	0.473
VQMT 14.0 precise	singlethreaded	YUV	HD 720p	17.81	0.059
VQMT 14.0 precise	singlethreaded	YUV	FullHD 1080p	7.9	0.134
VQMT 14.0 precise	singlethreaded	YUV	4K 2160p	1.99	0.546

Benchmark on VQMT 14.0 BETA r12792 PRO for Linux.
CPU: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64 cores
GPU: NVIDIA CUDA/TITAN RTX

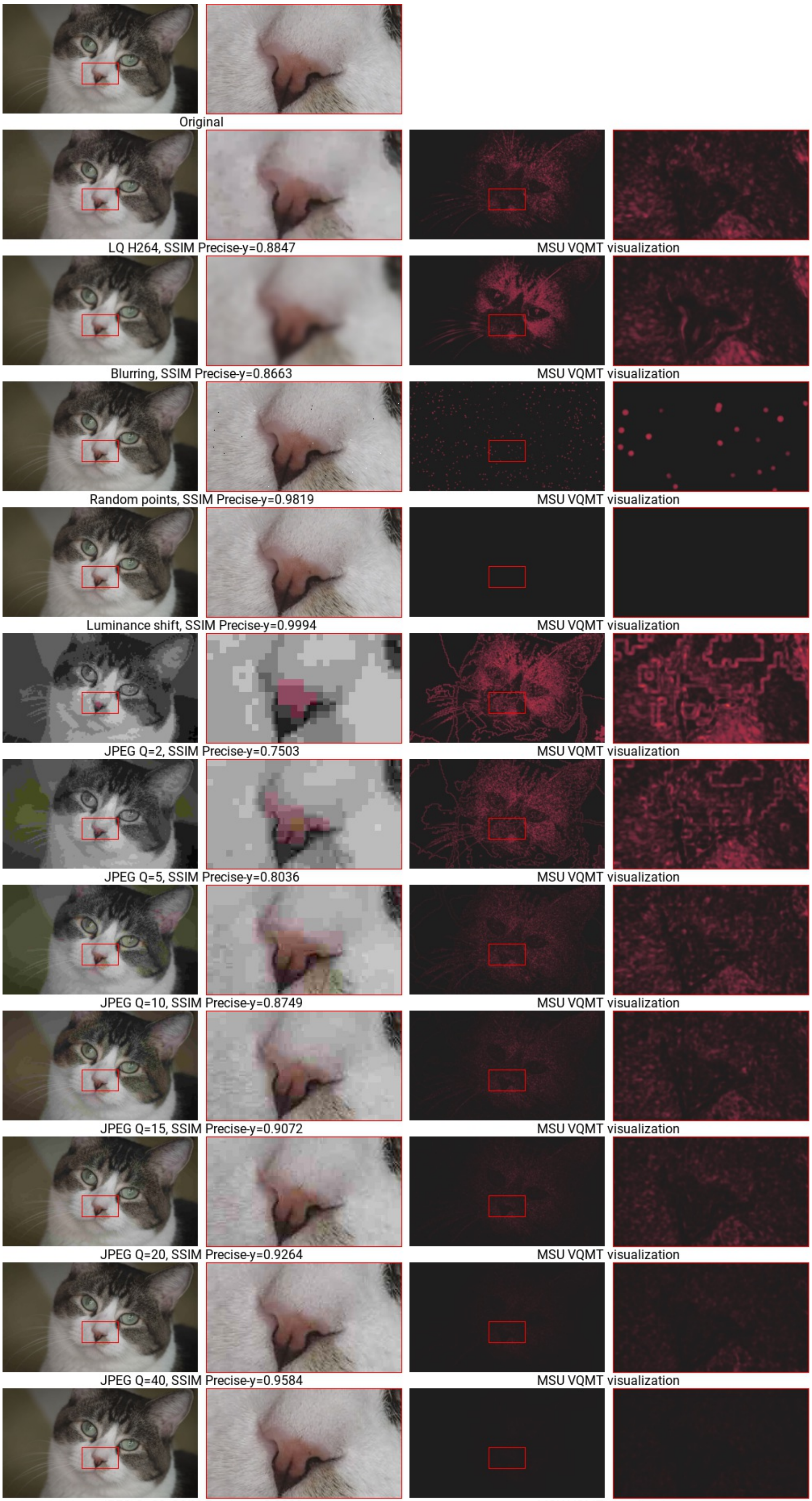
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 OpenCL	OpenCL	Y	HD 720p	986.1	0.003
VQMT 14.0 OpenCL	OpenCL	Y	FullHD 1080p	526.14	0.005
VQMT 14.0 OpenCL	OpenCL	Y	4K 2160p	72.32	0.022
VQMT 14.0 superfast	multithreaded	Y	HD 720p	1283.81	0.002
VQMT 14.0 superfast	multithreaded	Y	FullHD 1080p	648.72	0.004
VQMT 14.0 superfast	multithreaded	Y	4K 2160p	65.73	0.023
VQMT 14.0 fast	multithreaded	Y	HD 720p	854.27	0.003
VQMT 14.0 fast	multithreaded	Y	FullHD 1080p	204.12	0.009
VQMT 14.0 fast	multithreaded	Y	4K 2160p	54.93	0.048
VQMT 14.0 precise	multithreaded	Y	HD 720p	582.73	0.004
VQMT 14.0 precise	multithreaded	Y	FullHD 1080p	190.36	0.009
VQMT 14.0 precise	multithreaded	Y	4K 2160p	55.06	0.045
VQMT 14.0 superfast	singlethreaded	Y	HD 720p	142.32	0.007
VQMT 14.0 superfast	singlethreaded	Y	FullHD 1080p	54.24	0.019
VQMT 14.0 superfast	singlethreaded	Y	4K 2160p	19.54	0.052
VQMT 14.0 fast	singlethreaded	Y	HD 720p	26.99	0.037
VQMT 14.0 fast	singlethreaded	Y	FullHD 1080p	12.65	0.08
VQMT 14.0 fast	singlethreaded	Y	4K 2160p	3.5	0.296
VQMT 14.0 precise	singlethreaded	Y	HD 720p	35.35	0.028
VQMT 14.0 precise	singlethreaded	Y	FullHD 1080p	18.98	0.053
VQMT 14.0 precise	singlethreaded	Y	4K 2160p	5.35	0.193
VQMT 14.0 OpenCL	OpenCL	YUV	HD 720p	368.49	0.005
VQMT 14.0 OpenCL	OpenCL	YUV	FullHD 1080p	201.04	0.009
VQMT 14.0 OpenCL	OpenCL	YUV	4K 2160p	52.51	0.035
VQMT 14.0 superfast	multithreaded	YUV	HD 720p	1111.23	0.003
VQMT 14.0 superfast	multithreaded	YUV	FullHD 1080p	402.11	0.006
VQMT 14.0 superfast	multithreaded	YUV	4K 2160p	109.49	0.021
VQMT 14.0 fast	multithreaded	YUV	HD 720p	286.64	0.007

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 fast	multithreaded	YUV	FullHD 1080p	161.71	0.014
VQMT 14.0 fast	multithreaded	YUV	4K 2160p	33.61	0.136
VQMT 14.0 precise	multithreaded	YUV	HD 720p	216.48	0.008
VQMT 14.0 precise	multithreaded	YUV	FullHD 1080p	97.35	0.018
VQMT 14.0 precise	multithreaded	YUV	4K 2160p	21.35	0.161
VQMT 14.0 superfast	singlethreaded	YUV	HD 720p	79.12	0.013
VQMT 14.0 superfast	singlethreaded	YUV	FullHD 1080p	38.22	0.027
VQMT 14.0 superfast	singlethreaded	YUV	4K 2160p	13.3	0.077
VQMT 14.0 fast	singlethreaded	YUV	HD 720p	11.32	0.089
VQMT 14.0 fast	singlethreaded	YUV	FullHD 1080p	5.28	0.193
VQMT 14.0 fast	singlethreaded	YUV	4K 2160p	1.48	0.705
VQMT 14.0 precise	singlethreaded	YUV	HD 720p	17.71	0.057
VQMT 14.0 precise	singlethreaded	YUV	FullHD 1080p	8.34	0.122
VQMT 14.0 precise	singlethreaded	YUV	4K 2160p	2.28	0.458

Benchmark on VQMT 13.1 r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 OpenCL	OpenCL	Y	HD 720p	339.83	0.004
VQMT 13.1 OpenCL	OpenCL	Y	FullHD 1080p	167.01	0.007
VQMT 13.1 OpenCL	OpenCL	Y	4K 2160p	44.57	0.025
VQMT 13.1 fast	multithreaded	Y	HD 720p	233.06	0.005
VQMT 13.1 fast	multithreaded	Y	FullHD 1080p	102.66	0.011
VQMT 13.1 fast	multithreaded	Y	4K 2160p	23.67	0.049
VQMT 13.1 precise	multithreaded	Y	HD 720p	114.89	0.01
VQMT 13.1 precise	multithreaded	Y	FullHD 1080p	51.22	0.021
VQMT 13.1 precise	multithreaded	Y	4K 2160p	12.35	0.096
VQMT 13.1 fast	singlethreaded	Y	HD 720p	55.71	0.019
VQMT 13.1 fast	singlethreaded	Y	FullHD 1080p	26.34	0.039
VQMT 13.1 fast	singlethreaded	Y	4K 2160p	7.06	0.15

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 precise	singlethreaded	Y	HD 720p	44.79	0.023
VQMT 13.1 precise	singlethreaded	Y	FullHD 1080p	21.42	0.049
VQMT 13.1 precise	singlethreaded	Y	4K 2160p	5.54	0.192



Original

LQ H264, SSIM Precise-y=0.8847

MSU VQMT visualization

Blurring, SSIM Precise-y=0.8663

MSU VQMT visualization

Random points, SSIM Precise-y=0.9819

MSU VQMT visualization

Luminance shift, SSIM Precise-y=0.9994

MSU VQMT visualization

JPEG Q=2, SSIM Precise-y=0.7503

MSU VQMT visualization

JPEG Q=5, SSIM Precise-y=0.8036

MSU VQMT visualization

JPEG Q=10, SSIM Precise-y=0.8749

MSU VQMT visualization

JPEG Q=15, SSIM Precise-y=0.9072

MSU VQMT visualization

JPEG Q=20, SSIM Precise-y=0.9264

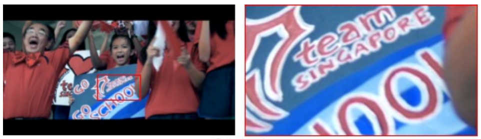
MSU VQMT visualization

JPEG Q=40, SSIM Precise-y=0.9584

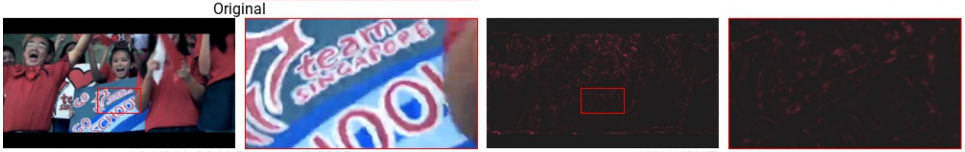
MSU VQMT visualization

JPEG Q=80, SSIM Precise-y=0.9829

MSU VQMT visualization

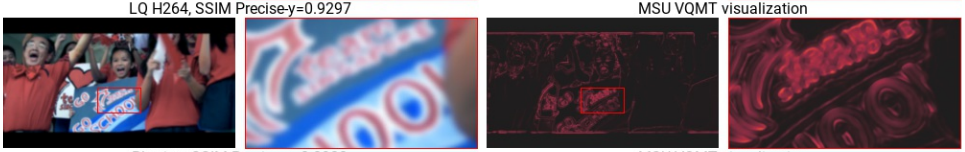


Original



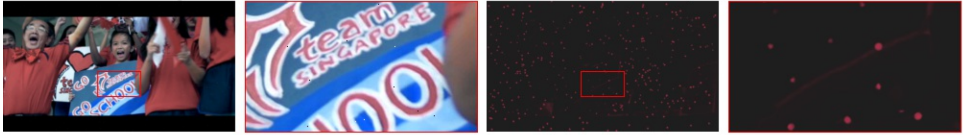
LQ H264, SSIM Precise-y=0.9297

MSU VQMT visualization



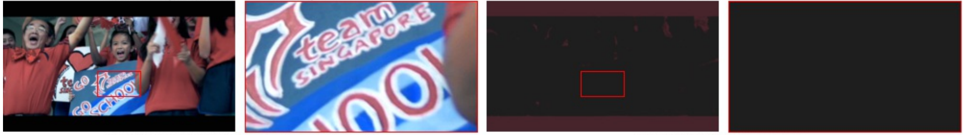
Blurring, SSIM Precise-y=0.8920

MSU VQMT visualization



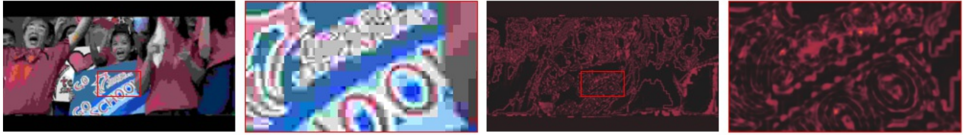
Random points, SSIM Precise-y=0.9698

MSU VQMT visualization



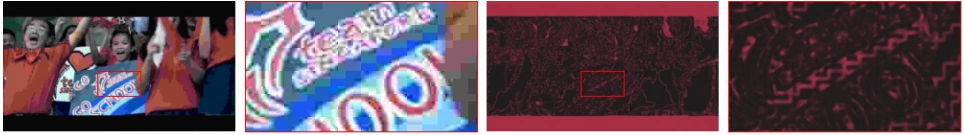
Luminance shift, SSIM Precise-y=0.9031

MSU VQMT visualization



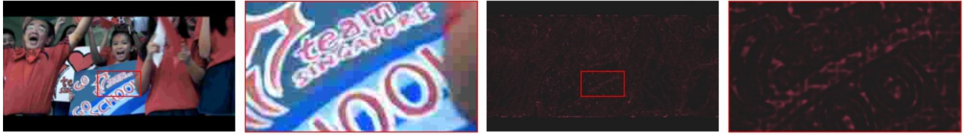
JPEG Q=2, SSIM Precise-y=0.7616

MSU VQMT visualization



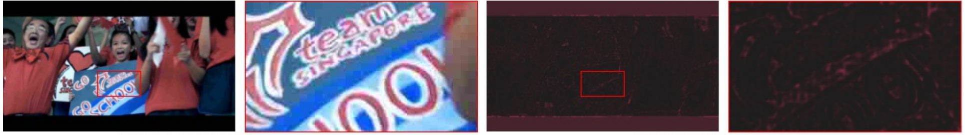
JPEG Q=5, SSIM Precise-y=0.6345

MSU VQMT visualization



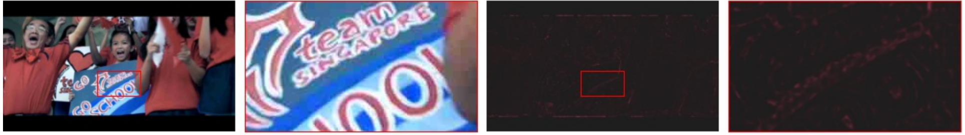
JPEG Q=10, SSIM Precise-y=0.9107

MSU VQMT visualization



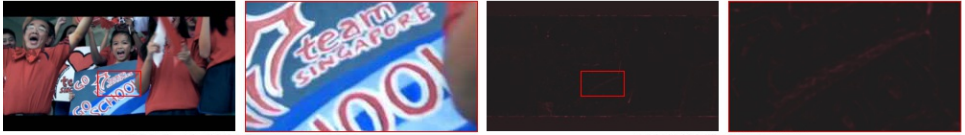
JPEG Q=15, SSIM Precise-y=0.8431

MSU VQMT visualization



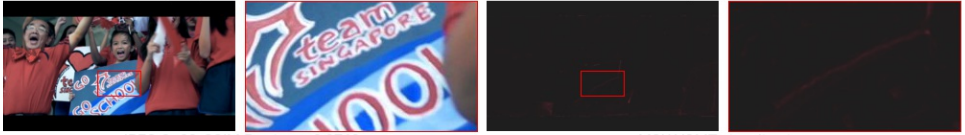
JPEG Q=20, SSIM Precise-y=0.9447

MSU VQMT visualization



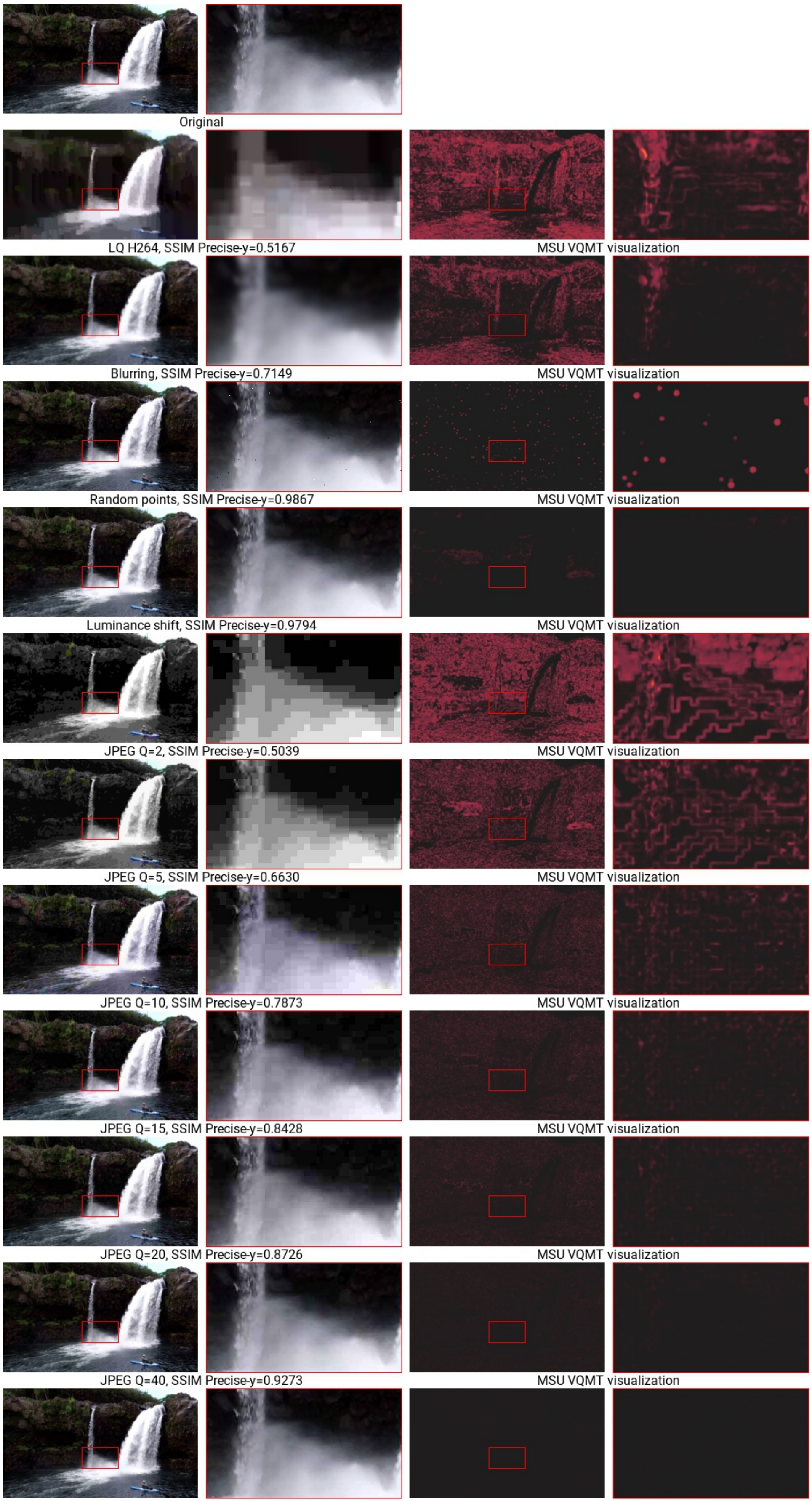
JPEG Q=40, SSIM Precise-y=0.9329

MSU VQMT visualization



JPEG Q=80, SSIM Precise-y=0.9769

MSU VQMT visualization



Original

LQ H264, SSIM Precise-y=0.5167 MSU VQMT visualization

Blurring, SSIM Precise-y=0.7149 MSU VQMT visualization

Random points, SSIM Precise-y=0.9867 MSU VQMT visualization

Luminance shift, SSIM Precise-y=0.9794 MSU VQMT visualization

JPEG Q=2, SSIM Precise-y=0.5039 MSU VQMT visualization

JPEG Q=5, SSIM Precise-y=0.6630 MSU VQMT visualization

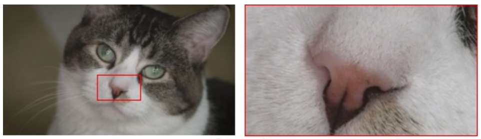
JPEG Q=10, SSIM Precise-y=0.7873 MSU VQMT visualization

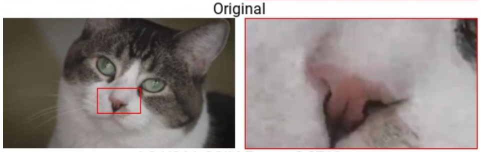
JPEG Q=15, SSIM Precise-y=0.8428 MSU VQMT visualization

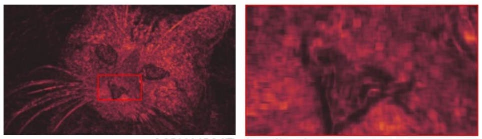
JPEG Q=20, SSIM Precise-y=0.8726 MSU VQMT visualization

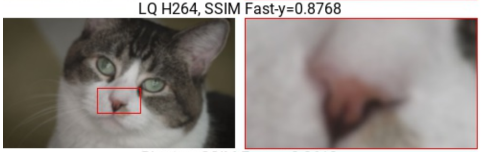
JPEG Q=40, SSIM Precise-y=0.9273 MSU VQMT visualization

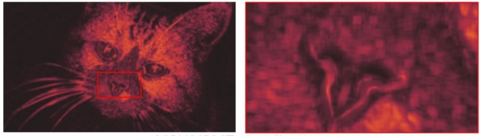
JPEG Q=80, SSIM Precise-y=0.9711 MSU VQMT visualization

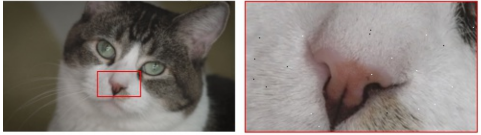

Original

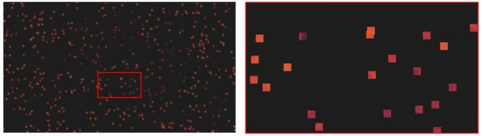

LQ H264, SSIM Fast-y=0.8768

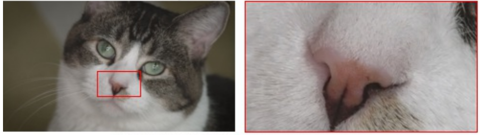

MSU VQMT visualization

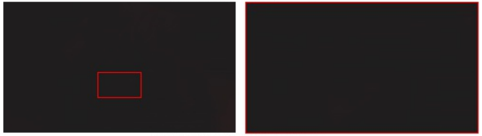

Blurring, SSIM Fast-y=0.8613

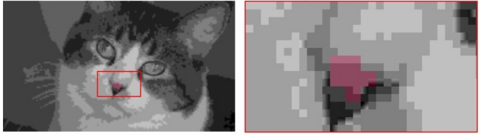

MSU VQMT visualization

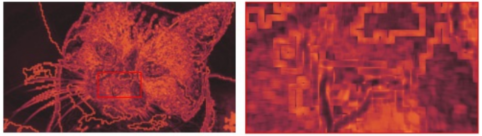

Random points, SSIM Fast-y=0.9790

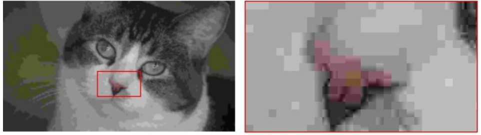

MSU VQMT visualization

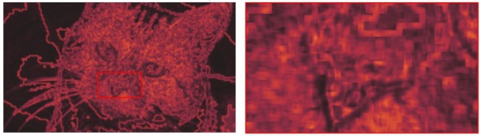

Luminance shift, SSIM Fast-y=0.9994

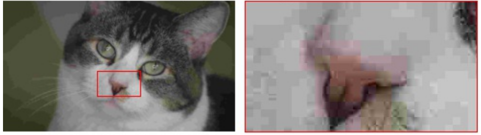

MSU VQMT visualization

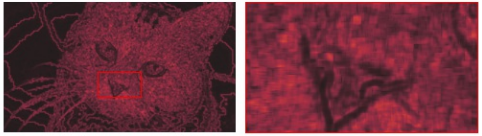

JPEG Q=2, SSIM Fast-y=0.7188

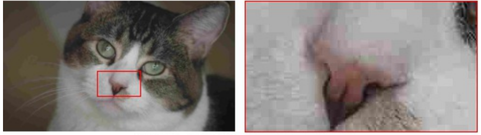

MSU VQMT visualization

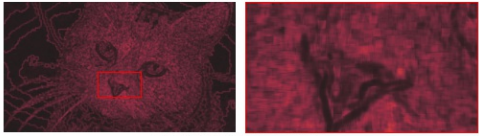

JPEG Q=5, SSIM Fast-y=0.7840

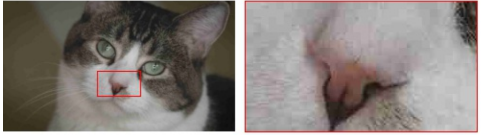

MSU VQMT visualization

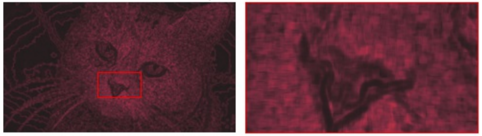

JPEG Q=10, SSIM Fast-y=0.8700

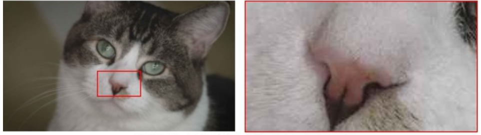

MSU VQMT visualization

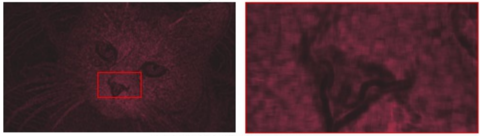

JPEG Q=15, SSIM Fast-y=0.9069

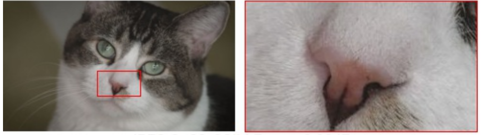

MSU VQMT visualization

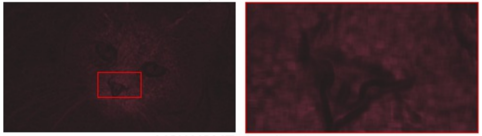

JPEG Q=20, SSIM Fast-y=0.9281

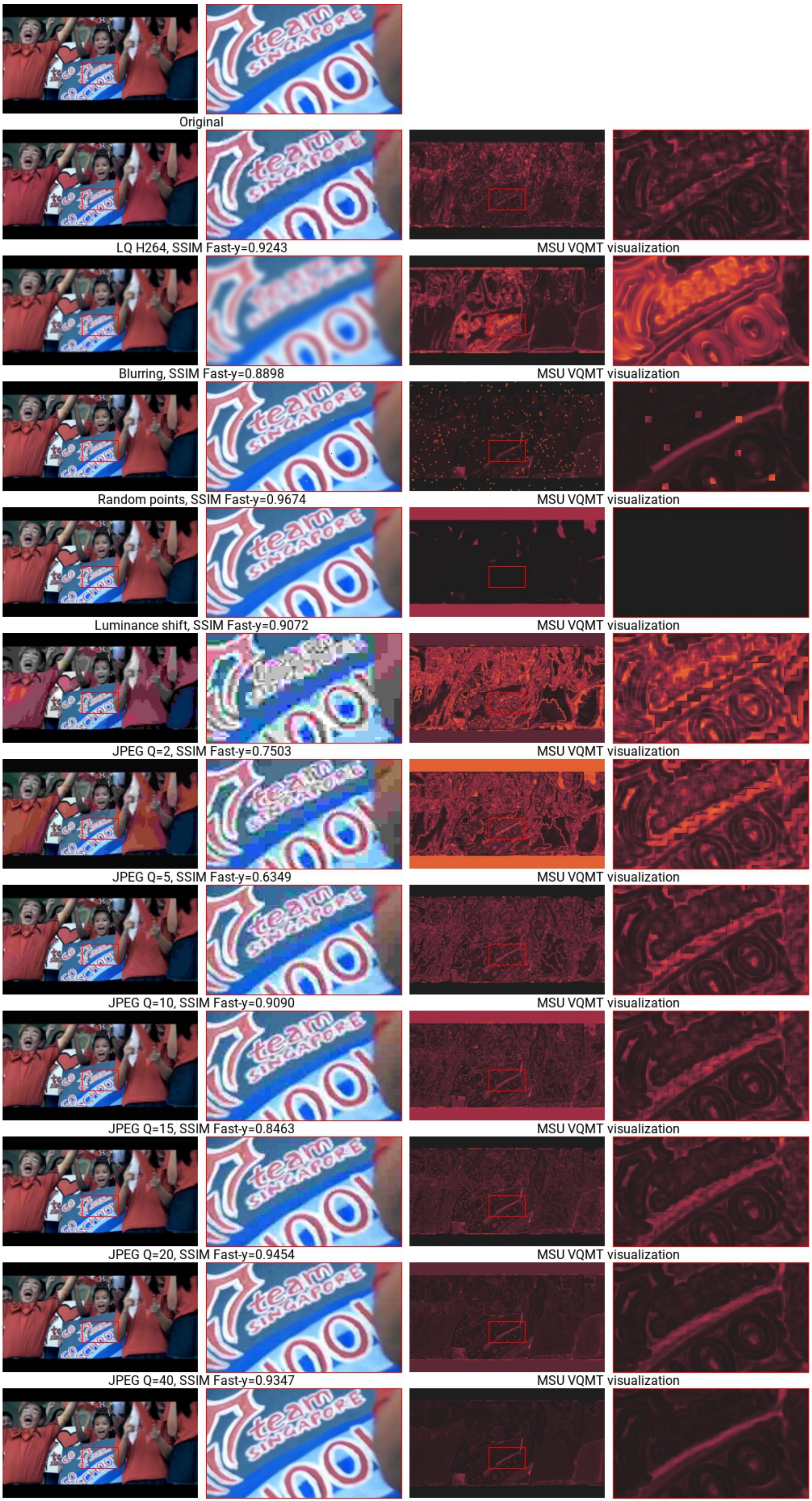

MSU VQMT visualization


JPEG Q=40, SSIM Fast-y=0.9613


MSU VQMT visualization


JPEG Q=80, SSIM Fast-y=0.9847


MSU VQMT visualization



Original

LQ H264, SSIM Fast-y=0.9243 MSU VQMT visualization

Blurring, SSIM Fast-y=0.8898 MSU VQMT visualization

Random points, SSIM Fast-y=0.9674 MSU VQMT visualization

Luminance shift, SSIM Fast-y=0.9072 MSU VQMT visualization

JPEG Q=2, SSIM Fast-y=0.7503 MSU VQMT visualization

JPEG Q=5, SSIM Fast-y=0.6349 MSU VQMT visualization

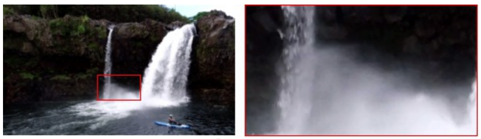
JPEG Q=10, SSIM Fast-y=0.9090 MSU VQMT visualization

JPEG Q=15, SSIM Fast-y=0.8463 MSU VQMT visualization

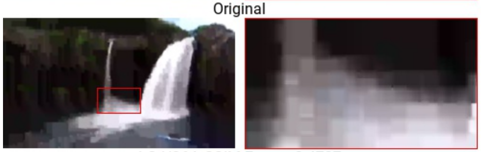
JPEG Q=20, SSIM Fast-y=0.9454 MSU VQMT visualization

JPEG Q=40, SSIM Fast-y=0.9347 MSU VQMT visualization

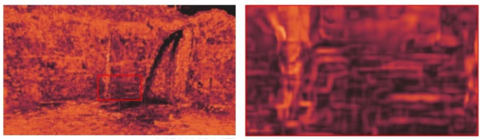
JPEG Q=80, SSIM Fast-y=0.9771 MSU VQMT visualization



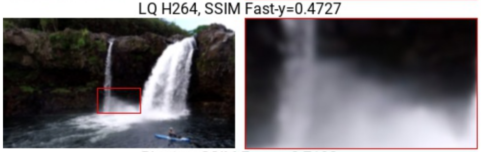
Original



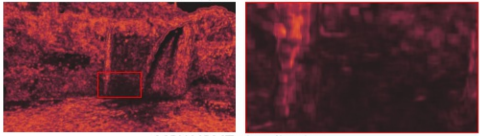
LQ H264, SSIM Fast-y=0.4727



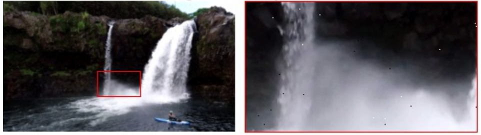
MSU VQMT visualization



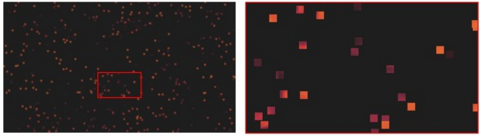
Blurring, SSIM Fast-y=0.7193



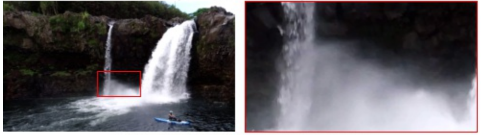
MSU VQMT visualization



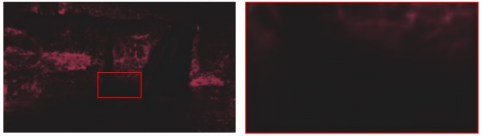
Random points, SSIM Fast-y=0.9859



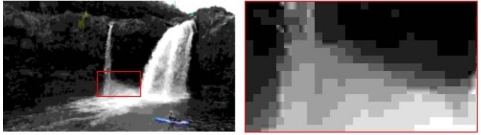
MSU VQMT visualization



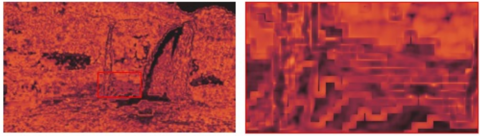
Luminance shift, SSIM Fast-y=0.9815



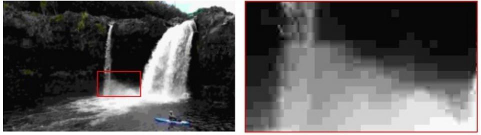
MSU VQMT visualization



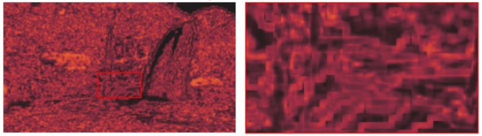
JPEG Q=2, SSIM Fast-y=0.4962



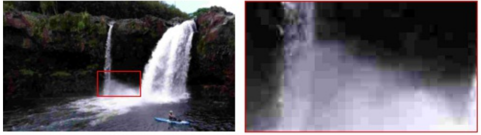
MSU VQMT visualization



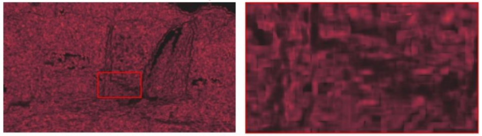
JPEG Q=5, SSIM Fast-y=0.6672



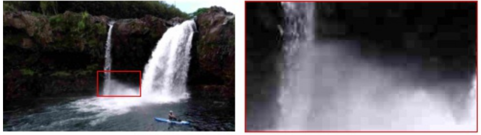
MSU VQMT visualization



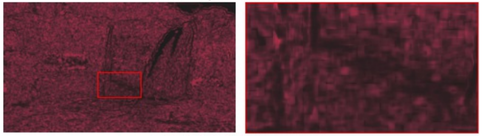
JPEG Q=10, SSIM Fast-y=0.8031



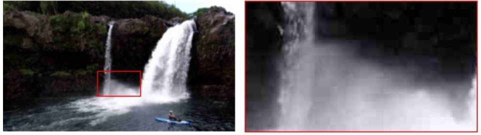
MSU VQMT visualization



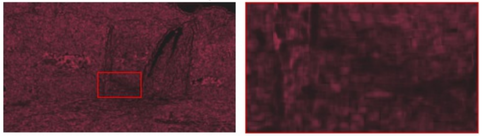
JPEG Q=15, SSIM Fast-y=0.8592



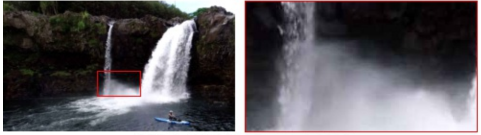
MSU VQMT visualization



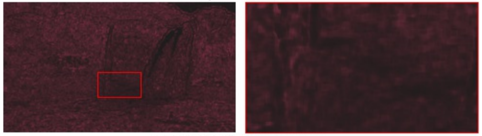
JPEG Q=20, SSIM Fast-y=0.8883



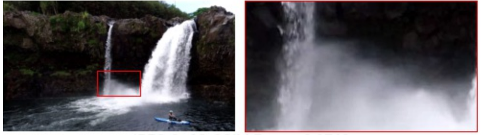
MSU VQMT visualization



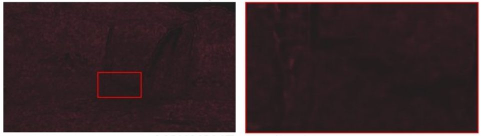
JPEG Q=40, SSIM Fast-y=0.9385



MSU VQMT visualization



JPEG Q=80, SSIM Fast-y=0.9763



MSU VQMT visualization

MS-SSIM (Multi-Scale Structural Similarity)

General info

Metric type	full-reference image metric
Value range	(images are different) -1..1 (images are same)
Value interpretation	bigger is better quality
MSU VQMT implementations	CPU multithreaded superfast (default), CPU multithreaded fast, cpu multithreaded precice, cpu multithreaded GPU identical, OpenCL (recommended), CUDA
MSU VQMT visualization	pixel-wise
Available colorspaces	Y, U, V, YUV, R, G, B, RGB
Output values	metric value
Aggregated values	standard set
MSU VQMT usages	-metr msssim [over <color components>] -metr msssim_precise [over <color components>] -metr msssim_gpu_id [over <color components>] -metr msssim_cuda [over <color components>] -metr msssim [over <color components>] -dev <OpenCL device>
Other names	MSSSIM
External links	original paper (Z. Wang, A. C. Bovik and E. P. Simoncelli)

Algorithm description

This metric performs SSIM calculation as described in SSIM paragraph for 5 scales of input images. Each next scale divides width and height by 2. The result SSIM values are producted with the following powers: 0.0448, 0.2856, 0.3001, 0.2363, 0.1333.

Benchmark

Please note: values can vary depending on system configuration, input format and other factors.

**Benchmark on VQMT 14.0 BETA r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti**

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 OpenCL	OpenCL	Y	HD 720p	260.39	0.005
VQMT 14.0 OpenCL	OpenCL	Y	FullHD 1080p	142.8	0.008
VQMT 14.0 OpenCL	OpenCL	Y	4K 2160p	40.52	0.028
VQMT 14.0 superfast	multithreaded	Y	HD 720p	507.07	0.003
VQMT 14.0 superfast	multithreaded	Y	FullHD 1080p	245.32	0.005
VQMT 14.0 superfast	multithreaded	Y	4K 2160p	53.64	0.021
VQMT 14.0 fast	multithreaded	Y	HD 720p	145.36	0.008

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 fast	multithreaded	Y	FullHD 1080p	64.56	0.017
VQMT 14.0 fast	multithreaded	Y	4K 2160p	13.5	0.092
VQMT 14.0 precise	multithreaded	Y	HD 720p	64.27	0.017
VQMT 14.0 precise	multithreaded	Y	FullHD 1080p	28.81	0.038
VQMT 14.0 precise	multithreaded	Y	4K 2160p	6.86	0.179
VQMT 14.0 superfast	singlethreaded	Y	HD 720p	147.65	0.008
VQMT 14.0 superfast	singlethreaded	Y	FullHD 1080p	71.87	0.015
VQMT 14.0 superfast	singlethreaded	Y	4K 2160p	23.86	0.045
VQMT 14.0 fast	singlethreaded	Y	HD 720p	44.63	0.023
VQMT 14.0 fast	singlethreaded	Y	FullHD 1080p	21.31	0.05
VQMT 14.0 fast	singlethreaded	Y	4K 2160p	5.46	0.198
VQMT 14.0 precise	singlethreaded	Y	HD 720p	27.4	0.038
VQMT 14.0 precise	singlethreaded	Y	FullHD 1080p	12.27	0.086
VQMT 14.0 precise	singlethreaded	Y	4K 2160p	3.09	0.348
VQMT 14.0 OpenCL	OpenCL	YUV	HD 720p	108.27	0.011
VQMT 14.0 OpenCL	OpenCL	YUV	FullHD 1080p	56.89	0.019
VQMT 14.0 OpenCL	OpenCL	YUV	4K 2160p	16.33	0.072
VQMT 14.0 superfast	multithreaded	YUV	HD 720p	333.33	0.004
VQMT 14.0 superfast	multithreaded	YUV	FullHD 1080p	157.74	0.008
VQMT 14.0 superfast	multithreaded	YUV	4K 2160p	37.6	0.031
VQMT 14.0 fast	multithreaded	YUV	HD 720p	57.37	0.019
VQMT 14.0 fast	multithreaded	YUV	FullHD 1080p	24.84	0.046
VQMT 14.0 fast	multithreaded	YUV	4K 2160p	5.07	0.257
VQMT 14.0 precise	multithreaded	YUV	HD 720p	22.82	0.047
VQMT 14.0 precise	multithreaded	YUV	FullHD 1080p	10.21	0.109
VQMT 14.0 precise	multithreaded	YUV	4K 2160p	2.51	0.521
VQMT 14.0 superfast	singlethreaded	YUV	HD 720p	91.76	0.012
VQMT 14.0 superfast	singlethreaded	YUV	FullHD 1080p	43.68	0.024
VQMT 14.0 superfast	singlethreaded	YUV	4K 2160p	13.38	0.08
VQMT 14.0 fast	singlethreaded	YUV	HD 720p	17.84	0.059

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 fast	singlethreaded	YUV	FullHD 1080p	7.94	0.133
VQMT 14.0 fast	singlethreaded	YUV	4K 2160p	1.96	0.555
VQMT 14.0 precise	singlethreaded	YUV	HD 720p	9.86	0.107
VQMT 14.0 precise	singlethreaded	YUV	FullHD 1080p	4.31	0.245
VQMT 14.0 precise	singlethreaded	YUV	4K 2160p	1.09	0.975

**Benchmark on VQMT 14.0 BETA r12792 PRO for Linux.
CPU: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64 cores
GPU: NVIDIA CUDA/TITAN RTX**

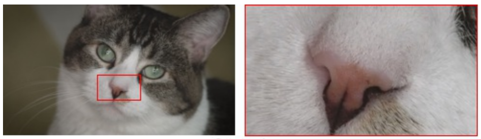
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 OpenCL	OpenCL	Y	HD 720p	788.75	0.003
VQMT 14.0 OpenCL	OpenCL	Y	FullHD 1080p	530.35	0.005
VQMT 14.0 OpenCL	OpenCL	Y	4K 2160p	67.99	0.023
VQMT 14.0 superfast	multithreaded	Y	HD 720p	1157.03	0.002
VQMT 14.0 superfast	multithreaded	Y	FullHD 1080p	626.04	0.004
VQMT 14.0 superfast	multithreaded	Y	4K 2160p	76.98	0.021
VQMT 14.0 fast	multithreaded	Y	HD 720p	331.02	0.005
VQMT 14.0 fast	multithreaded	Y	FullHD 1080p	79.82	0.02
VQMT 14.0 fast	multithreaded	Y	4K 2160p	11.97	0.178
VQMT 14.0 precise	multithreaded	Y	HD 720p	373.32	0.005
VQMT 14.0 precise	multithreaded	Y	FullHD 1080p	135.07	0.012
VQMT 14.0 precise	multithreaded	Y	4K 2160p	33.4	0.084
VQMT 14.0 superfast	singlethreaded	Y	HD 720p	90.46	0.011
VQMT 14.0 superfast	singlethreaded	Y	FullHD 1080p	41.53	0.024
VQMT 14.0 superfast	singlethreaded	Y	4K 2160p	15.84	0.065
VQMT 14.0 fast	singlethreaded	Y	HD 720p	23.67	0.043
VQMT 14.0 fast	singlethreaded	Y	FullHD 1080p	11.75	0.086
VQMT 14.0 fast	singlethreaded	Y	4K 2160p	3.73	0.277
VQMT 14.0 precise	singlethreaded	Y	HD 720p	22.92	0.044
VQMT 14.0 precise	singlethreaded	Y	FullHD 1080p	11.47	0.088
VQMT 14.0 precise	singlethreaded	Y	4K 2160p	3.0	0.345

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 OpenCL	OpenCL	YUV	HD 720p	302.88	0.005
VQMT 14.0 OpenCL	OpenCL	YUV	FullHD 1080p	144.55	0.011
VQMT 14.0 OpenCL	OpenCL	YUV	4K 2160p	42.86	0.037
VQMT 14.0 superfast	multithreaded	YUV	HD 720p	1080.48	0.003
VQMT 14.0 superfast	multithreaded	YUV	FullHD 1080p	380.3	0.006
VQMT 14.0 superfast	multithreaded	YUV	4K 2160p	90.99	0.022
VQMT 14.0 fast	multithreaded	YUV	HD 720p	141.68	0.012
VQMT 14.0 fast	multithreaded	YUV	FullHD 1080p	56.37	0.036
VQMT 14.0 fast	multithreaded	YUV	4K 2160p	8.04	0.354
VQMT 14.0 precise	multithreaded	YUV	HD 720p	133.28	0.011
VQMT 14.0 precise	multithreaded	YUV	FullHD 1080p	61.39	0.032
VQMT 14.0 precise	multithreaded	YUV	4K 2160p	11.03	0.305
VQMT 14.0 superfast	singlethreaded	YUV	HD 720p	55.19	0.018
VQMT 14.0 superfast	singlethreaded	YUV	FullHD 1080p	28.86	0.035
VQMT 14.0 superfast	singlethreaded	YUV	4K 2160p	9.99	0.103
VQMT 14.0 fast	singlethreaded	YUV	HD 720p	10.52	0.096
VQMT 14.0 fast	singlethreaded	YUV	FullHD 1080p	5.18	0.196
VQMT 14.0 fast	singlethreaded	YUV	4K 2160p	1.29	0.801
VQMT 14.0 precise	singlethreaded	YUV	HD 720p	10.23	0.099
VQMT 14.0 precise	singlethreaded	YUV	FullHD 1080p	4.72	0.215
VQMT 14.0 precise	singlethreaded	YUV	4K 2160p	1.17	0.883

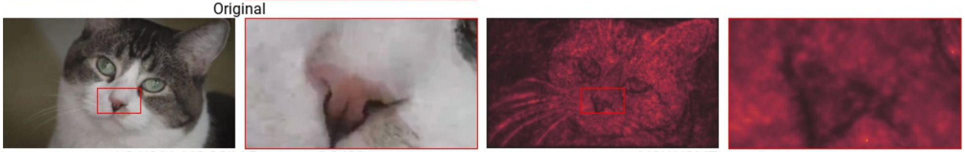
**Benchmark on VQMT 13.1 r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti**

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 OpenCL	OpenCL	Y	HD 720p	270.25	0.005
VQMT 13.1 OpenCL	OpenCL	Y	FullHD 1080p	146.66	0.008
VQMT 13.1 OpenCL	OpenCL	Y	4K 2160p	38.95	0.028
VQMT 13.1 fast	multithreaded	Y	HD 720p	152.03	0.007
VQMT 13.1 fast	multithreaded	Y	FullHD 1080p	67.06	0.017

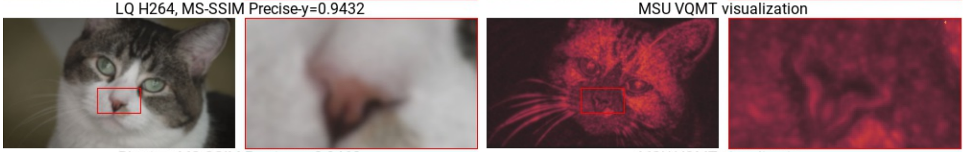
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 fast	multithreaded	Y	4K 2160p	13.72	0.089
VQMT 13.1 precise	multithreaded	Y	HD 720p	64.31	0.016
VQMT 13.1 precise	multithreaded	Y	FullHD 1080p	28.71	0.038
VQMT 13.1 precise	multithreaded	Y	4K 2160p	6.79	0.178
VQMT 13.1 fast	singlethreaded	Y	HD 720p	51.11	0.02
VQMT 13.1 fast	singlethreaded	Y	FullHD 1080p	24.44	0.043
VQMT 13.1 fast	singlethreaded	Y	4K 2160p	6.3	0.169
VQMT 13.1 precise	singlethreaded	Y	HD 720p	27.35	0.038
VQMT 13.1 precise	singlethreaded	Y	FullHD 1080p	12.17	0.085
VQMT 13.1 precise	singlethreaded	Y	4K 2160p	3.09	0.345



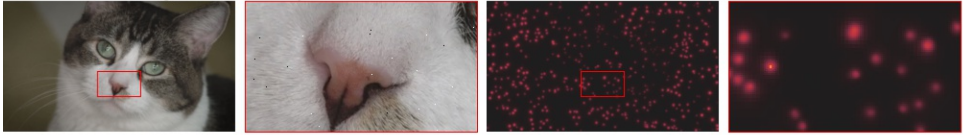
Original



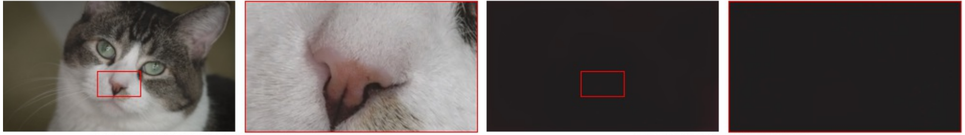
LQ H264, MS-SSIM Precise-y=0.9432 MSU VQMT visualization



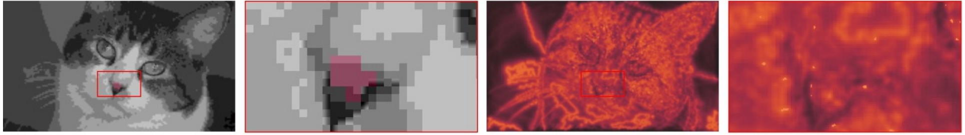
Blurring, MS-SSIM Precise-y=0.9460 MSU VQMT visualization



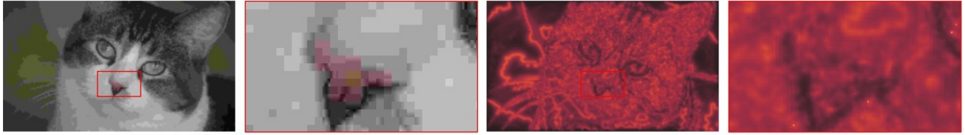
Random points, MS-SSIM Precise-y=0.9927 MSU VQMT visualization



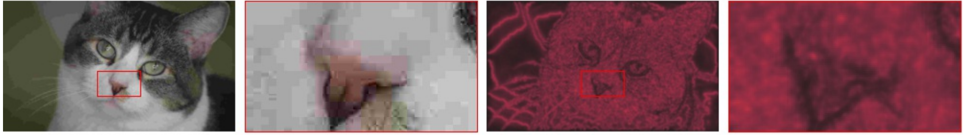
Luminance shift, MS-SSIM Precise-y=0.9999 MSU VQMT visualization



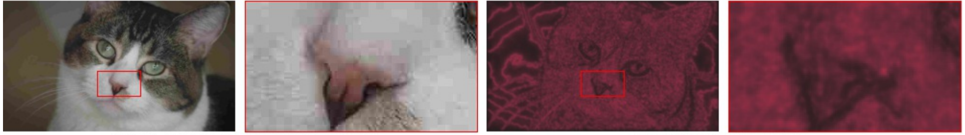
JPEG Q=2, MS-SSIM Precise-y=0.7857 MSU VQMT visualization



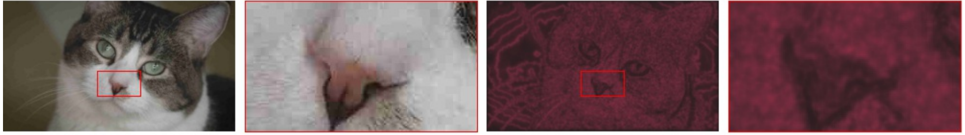
JPEG Q=5, MS-SSIM Precise-y=0.8568 MSU VQMT visualization



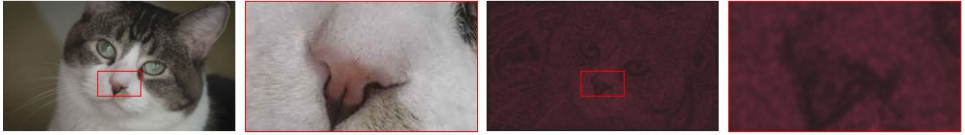
JPEG Q=10, MS-SSIM Precise-y=0.9351 MSU VQMT visualization



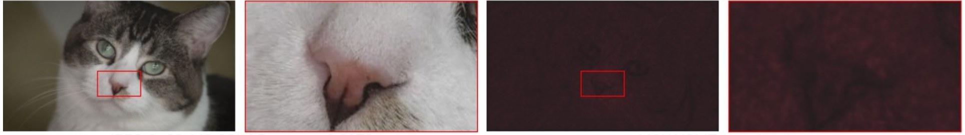
JPEG Q=15, MS-SSIM Precise-y=0.9593 MSU VQMT visualization



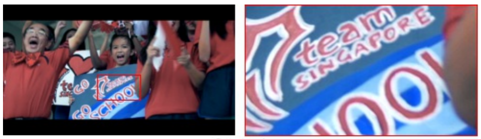
JPEG Q=20, MS-SSIM Precise-y=0.9722 MSU VQMT visualization



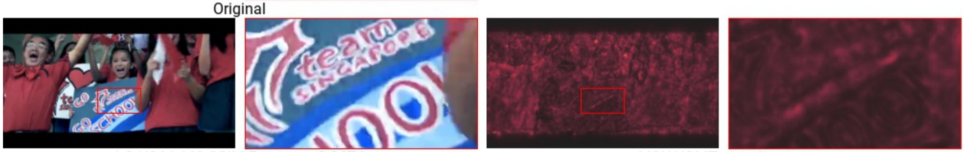
JPEG Q=40, MS-SSIM Precise-y=0.9890 MSU VQMT visualization



JPEG Q=80, MS-SSIM Precise-y=0.9972 MSU VQMT visualization

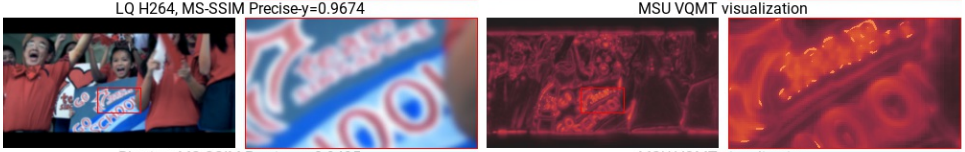


Original



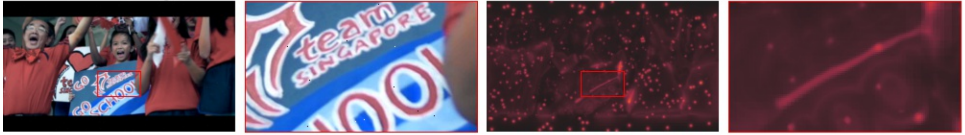
LQ H264, MS-SSIM Precise-y=0.9674

MSU VQMT visualization



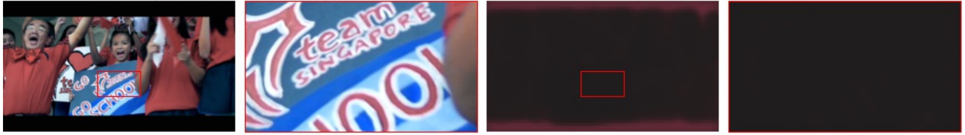
Blurring, MS-SSIM Precise-y=0.9495

MSU VQMT visualization



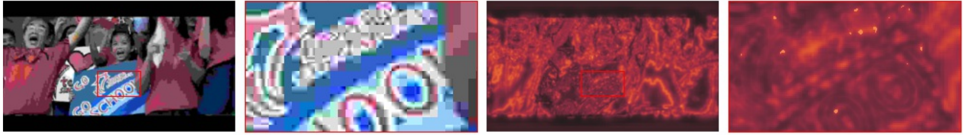
Random points, MS-SSIM Precise-y=0.9830

MSU VQMT visualization



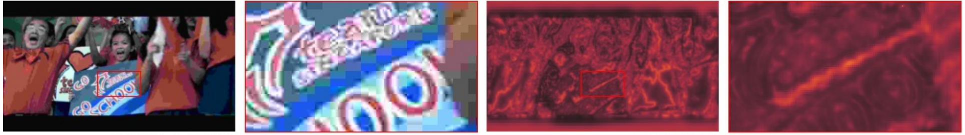
Luminance shift, MS-SSIM Precise-y=0.9953

MSU VQMT visualization



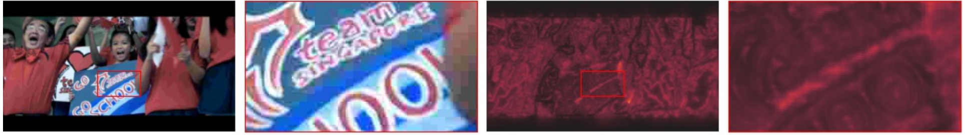
JPEG Q=2, MS-SSIM Precise-y=0.8658

MSU VQMT visualization



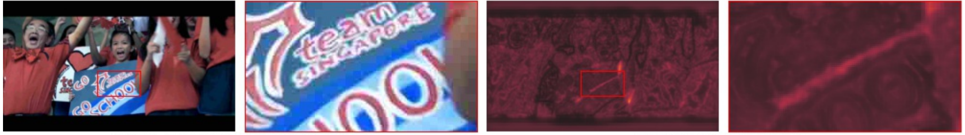
JPEG Q=5, MS-SSIM Precise-y=0.8934

MSU VQMT visualization



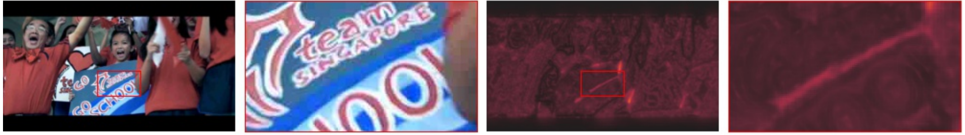
JPEG Q=10, MS-SSIM Precise-y=0.9560

MSU VQMT visualization



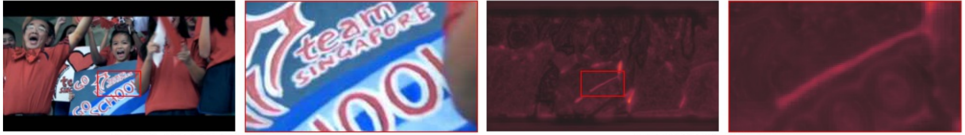
JPEG Q=15, MS-SSIM Precise-y=0.9654

MSU VQMT visualization



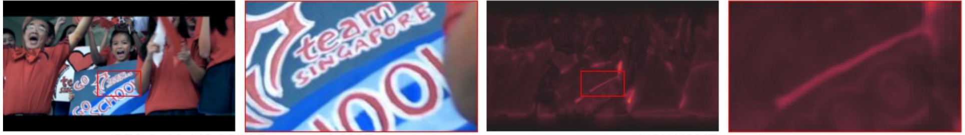
JPEG Q=20, MS-SSIM Precise-y=0.9761

MSU VQMT visualization



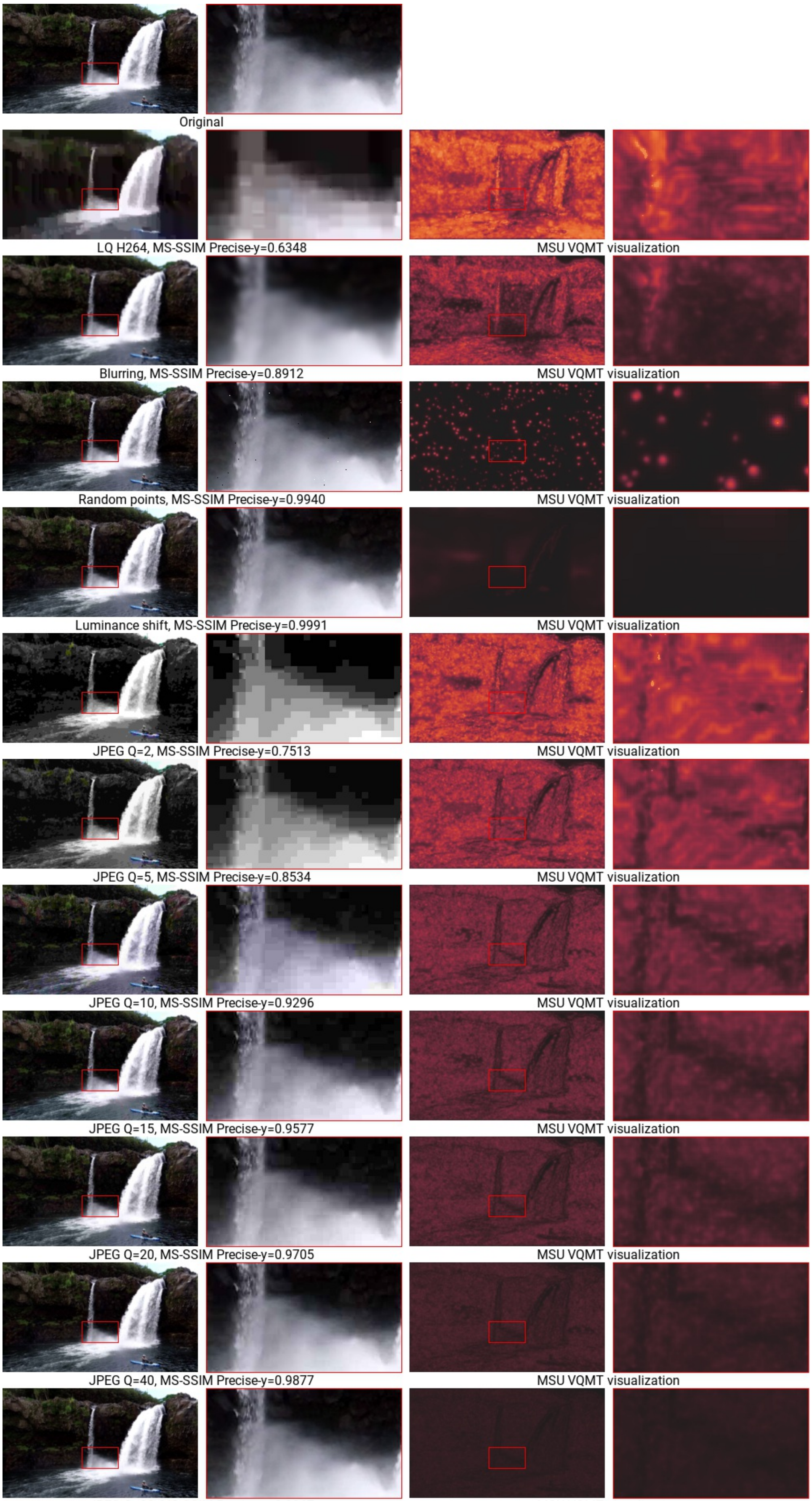
JPEG Q=40, MS-SSIM Precise-y=0.9832

MSU VQMT visualization



JPEG Q=80, MS-SSIM Precise-y=0.9887

MSU VQMT visualization



Original

LQ H264, MS-SSIM Precise-y=0.6348 MSU VQMT visualization

Blurring, MS-SSIM Precise-y=0.8912 MSU VQMT visualization

Random points, MS-SSIM Precise-y=0.9940 MSU VQMT visualization

Luminance shift, MS-SSIM Precise-y=0.9991 MSU VQMT visualization

JPEG Q=2, MS-SSIM Precise-y=0.7513 MSU VQMT visualization

JPEG Q=5, MS-SSIM Precise-y=0.8534 MSU VQMT visualization

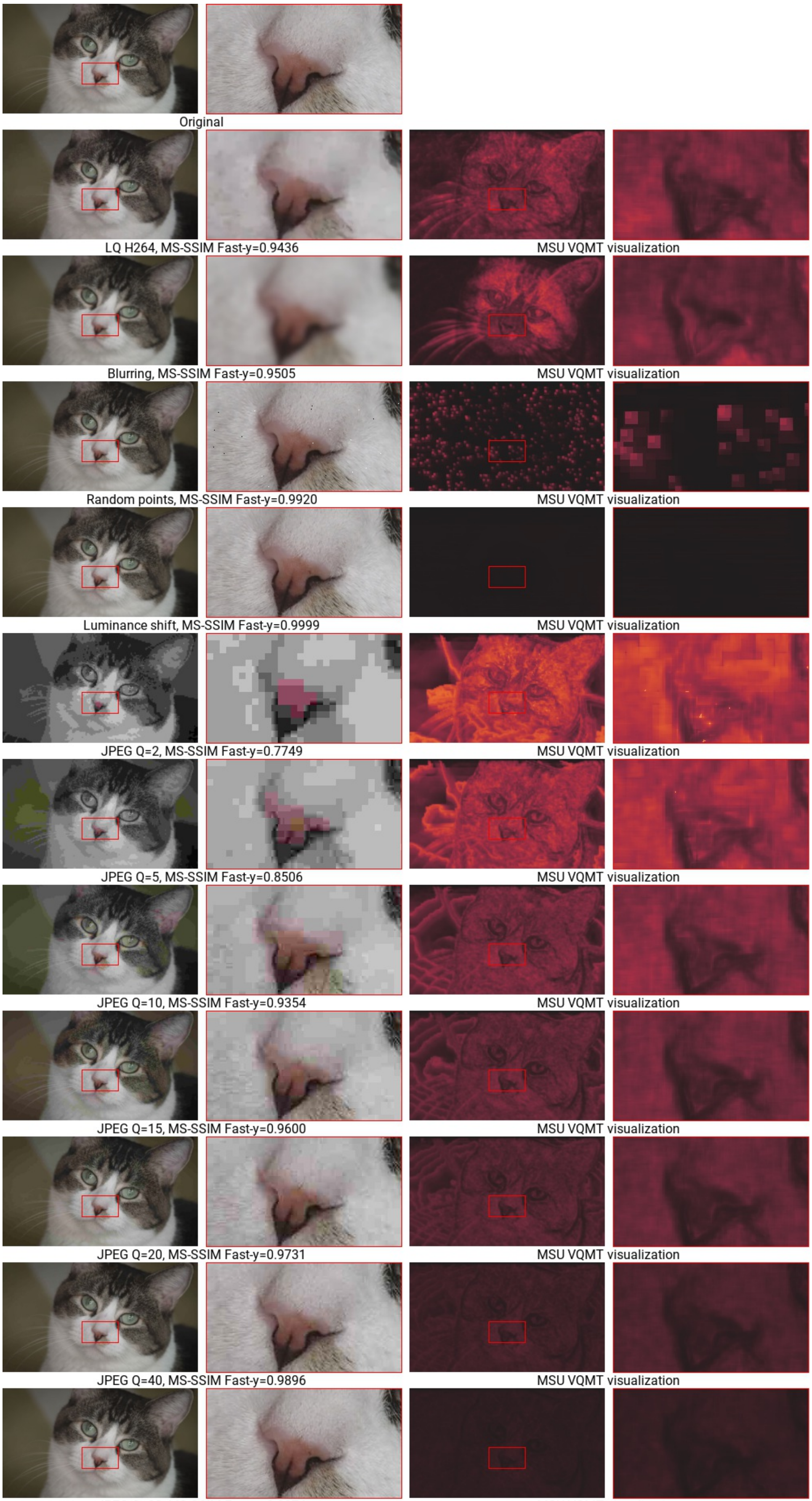
JPEG Q=10, MS-SSIM Precise-y=0.9296 MSU VQMT visualization

JPEG Q=15, MS-SSIM Precise-y=0.9577 MSU VQMT visualization

JPEG Q=20, MS-SSIM Precise-y=0.9705 MSU VQMT visualization

JPEG Q=40, MS-SSIM Precise-y=0.9877 MSU VQMT visualization

JPEG Q=80, MS-SSIM Precise-y=0.9967 MSU VQMT visualization



Original

LQ H264, MS-SSIM Fast-y=0.9436

MSU VQMT visualization

Blurring, MS-SSIM Fast-y=0.9505

MSU VQMT visualization

Random points, MS-SSIM Fast-y=0.9920

MSU VQMT visualization

Luminance shift, MS-SSIM Fast-y=0.9999

MSU VQMT visualization

JPEG Q=2, MS-SSIM Fast-y=0.7749

MSU VQMT visualization

JPEG Q=5, MS-SSIM Fast-y=0.8506

MSU VQMT visualization

JPEG Q=10, MS-SSIM Fast-y=0.9354

MSU VQMT visualization

JPEG Q=15, MS-SSIM Fast-y=0.9600

MSU VQMT visualization

JPEG Q=20, MS-SSIM Fast-y=0.9731

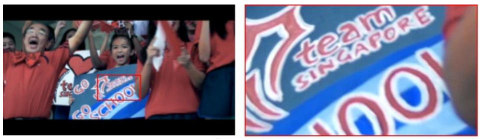
MSU VQMT visualization

JPEG Q=40, MS-SSIM Fast-y=0.9896

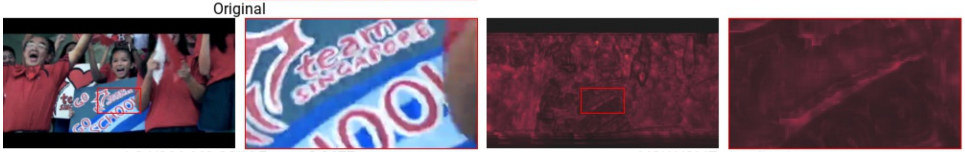
MSU VQMT visualization

JPEG Q=80, MS-SSIM Fast-y=0.9974

MSU VQMT visualization

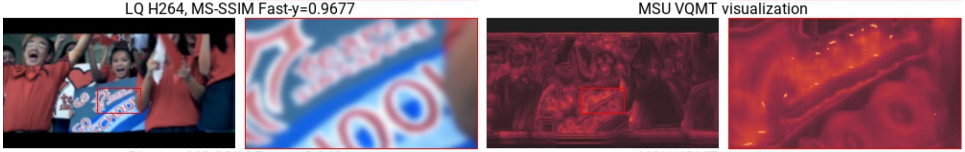


Original



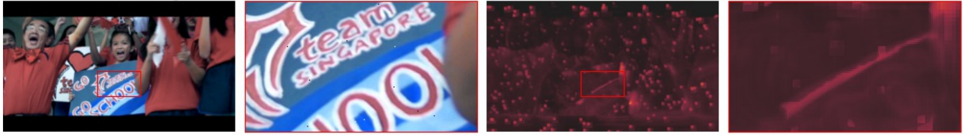
LQ H264, MS-SSIM Fast-y=0.9677

MSU VQMT visualization



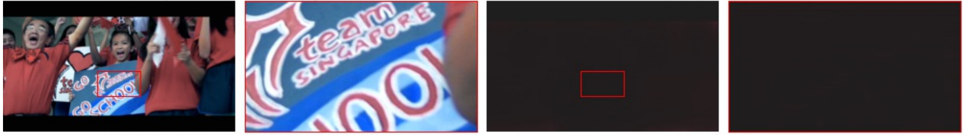
Blurring, MS-SSIM Fast-y=0.9491

MSU VQMT visualization



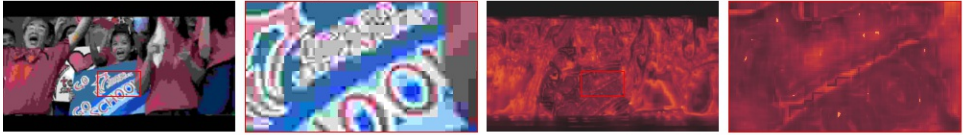
Random points, MS-SSIM Fast-y=0.9807

MSU VQMT visualization



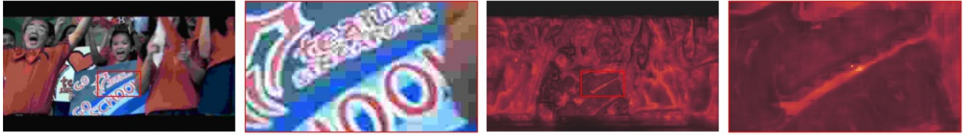
Luminance shift, MS-SSIM Fast-y=0.9998

MSU VQMT visualization



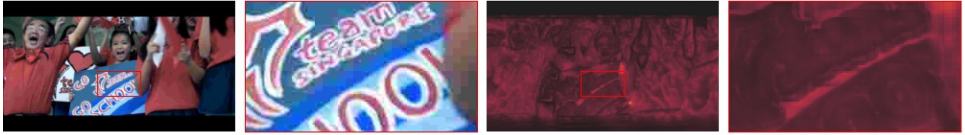
JPEG Q=2, MS-SSIM Fast-y=0.8670

MSU VQMT visualization



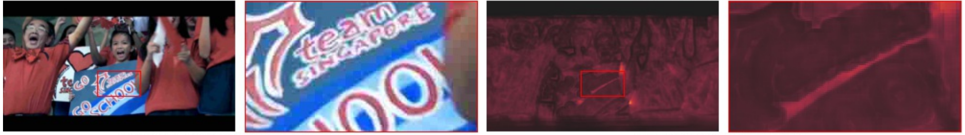
JPEG Q=5, MS-SSIM Fast-y=0.9124

MSU VQMT visualization



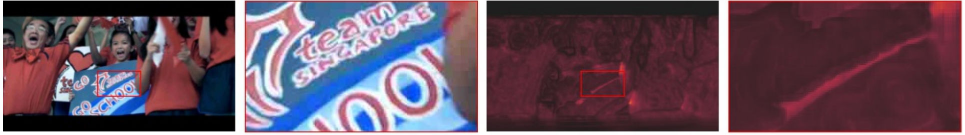
JPEG Q=10, MS-SSIM Fast-y=0.9564

MSU VQMT visualization



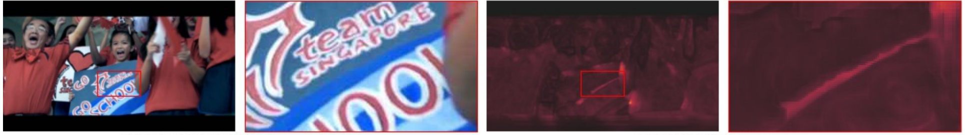
JPEG Q=15, MS-SSIM Fast-y=0.9694

MSU VQMT visualization



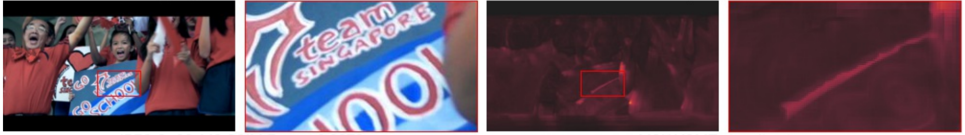
JPEG Q=20, MS-SSIM Fast-y=0.9755

MSU VQMT visualization



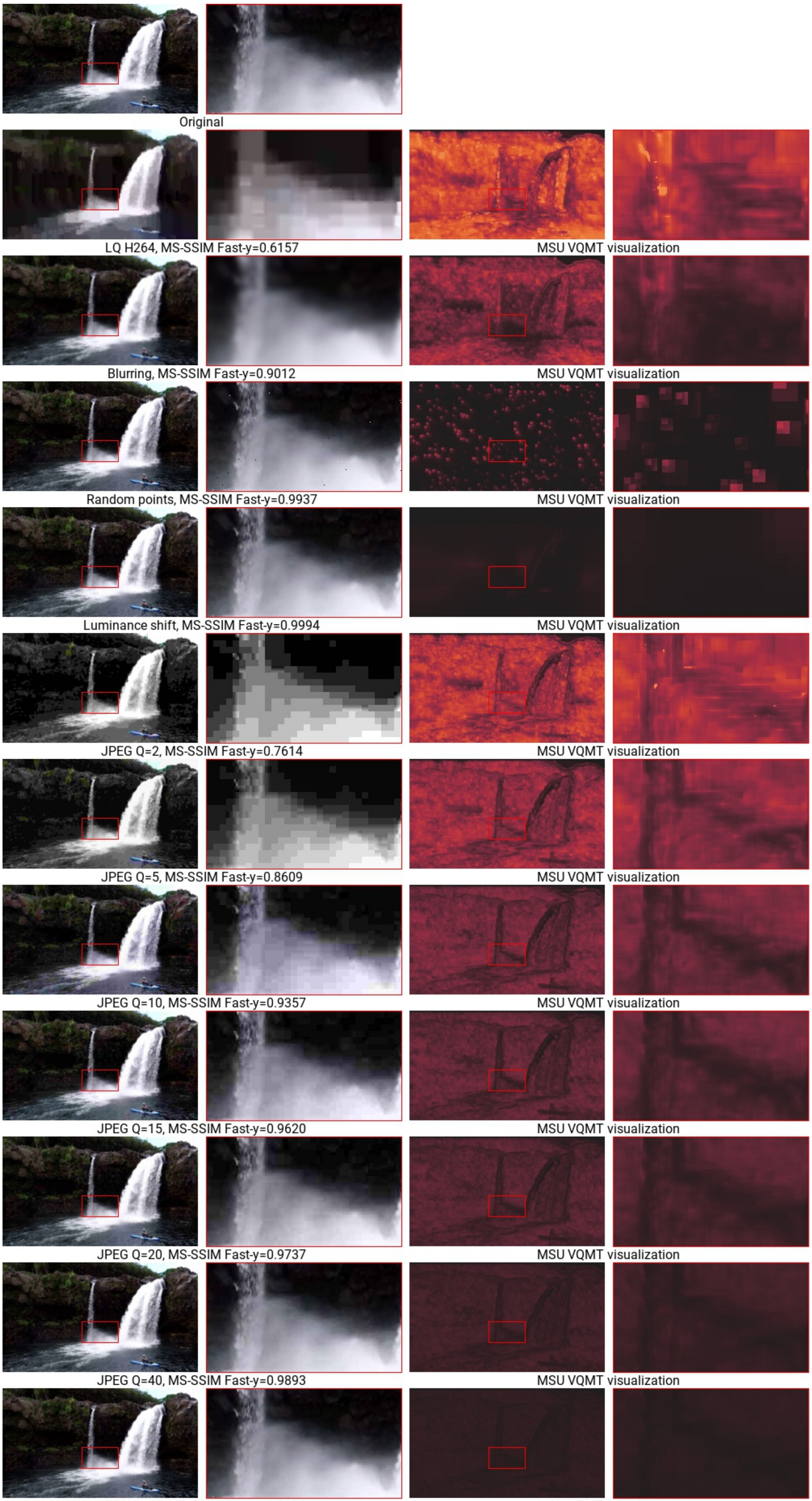
JPEG Q=40, MS-SSIM Fast-y=0.9833

MSU VQMT visualization



JPEG Q=80, MS-SSIM Fast-y=0.9872

MSU VQMT visualization



Original

LQ H264, MS-SSIM Fast-y=0.6157

MSU VQMT visualization

Blurring, MS-SSIM Fast-y=0.9012

MSU VQMT visualization

Random points, MS-SSIM Fast-y=0.9937

MSU VQMT visualization

Luminance shift, MS-SSIM Fast-y=0.9994

MSU VQMT visualization

JPEG Q=2, MS-SSIM Fast-y=0.7614

MSU VQMT visualization

JPEG Q=5, MS-SSIM Fast-y=0.8609

MSU VQMT visualization

JPEG Q=10, MS-SSIM Fast-y=0.9357

MSU VQMT visualization

JPEG Q=15, MS-SSIM Fast-y=0.9620

MSU VQMT visualization

JPEG Q=20, MS-SSIM Fast-y=0.9737

MSU VQMT visualization

JPEG Q=40, MS-SSIM Fast-y=0.9893

MSU VQMT visualization

JPEG Q=80, MS-SSIM Fast-y=0.9972

MSU VQMT visualization

3-SSIM (Three-commponent Structural Similarity)

General info

Metric type	full-reference image metric
Value range	(images are different) -1..1 (images are same)
Value interpretation	bigger is better quality
MSU VQMT implementations	CPU multithreaded (default), OpenCL (recommended), CUDA
Available colorspaces	Y, U, V
Output values	metric value
Aggregated values	standard set
MSU VQMT usages	-metr 3ssim [over <color components>] -metr 3ssim_cuda [over <color components>] -metr 3ssim [over <color components>] -dev <OpenCL device>
Other names	3-SSIM
External links	original paper (C. Li and A. C. Bovik)

Algorithm description

3-Component SSIM Index based on region division of source frames. There are 3 types of regions – edges, textures and smooth regions. Result metric calculated as weighted average of SSIM metric for those regions. In fact, human eye can see difference more precisely on textured or edge regions than on smooth regions. Division based on gradient magnitude is presented in every pixel of images.

Benchmark

Please note: values can vary depending on system configuration, input format and other factors.

**Benchmark on VQMT 14.0 BETA r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti**

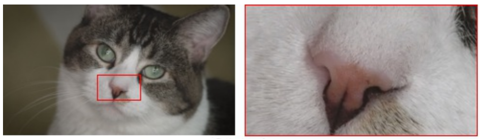
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	multithreaded	Y	HD 720p	54.74	0.02
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	24.39	0.045
VQMT 14.0 default	multithreaded	Y	4K 2160p	5.83	0.209
VQMT 14.0 default	singlethreaded	Y	HD 720p	29.26	0.036
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	12.99	0.081
VQMT 14.0 default	singlethreaded	Y	4K 2160p	3.29	0.328

**Benchmark on VQMT 14.0 BETA r12792 PRO for Linux.
CPU: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64 cores
GPU: NVIDIA CUDA/TITAN RTX**

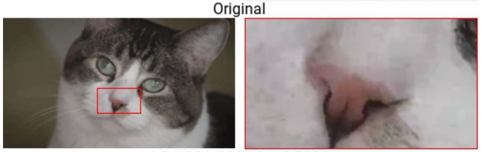
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	multithreaded	Y	HD 720p	309.37	0.005
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	114.47	0.013
VQMT 14.0 default	multithreaded	Y	4K 2160p	28.56	0.086
VQMT 14.0 default	singlethreaded	Y	HD 720p	24.26	0.042
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	11.52	0.088
VQMT 14.0 default	singlethreaded	Y	4K 2160p	3.32	0.31

Benchmark on VQMT 13.1 r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

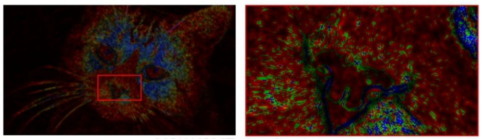
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 default	multithreaded	Y	HD 720p	54.9	0.019
VQMT 13.1 default	multithreaded	Y	FullHD 1080p	24.34	0.044
VQMT 13.1 default	multithreaded	Y	4K 2160p	5.84	0.206
VQMT 13.1 default	singlethreaded	Y	HD 720p	29.5	0.035
VQMT 13.1 default	singlethreaded	Y	FullHD 1080p	12.94	0.08
VQMT 13.1 default	singlethreaded	Y	4K 2160p	3.28	0.325



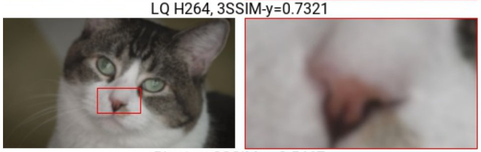
Original



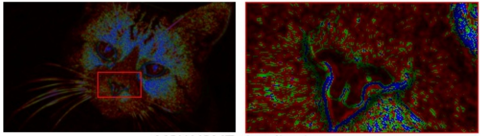
LQ H264, 3SSIM-y=0.7321



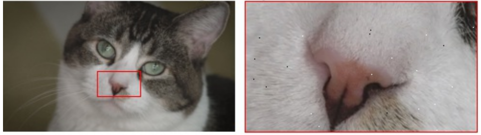
MSU VQMT visualization



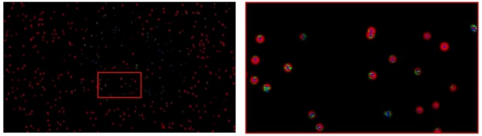
Blurring, 3SSIM-y=0.5467



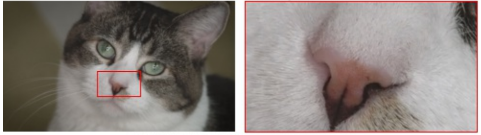
MSU VQMT visualization



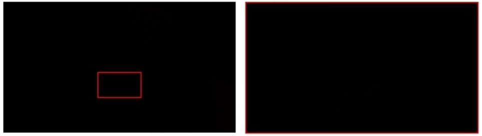
Random points, 3SSIM-y=0.9679



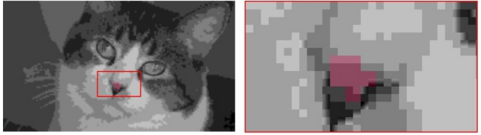
MSU VQMT visualization



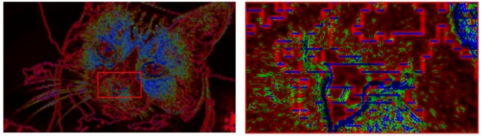
Luminance shift, 3SSIM-y=0.9999



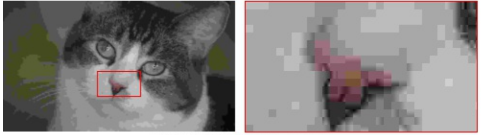
MSU VQMT visualization



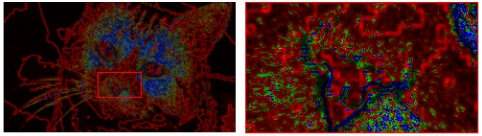
JPEG Q=2, 3SSIM-y=0.4614



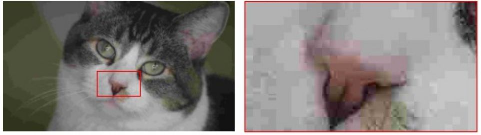
MSU VQMT visualization



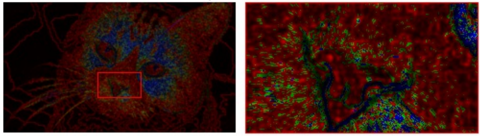
JPEG Q=5, 3SSIM-y=0.6304



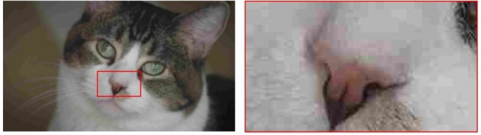
MSU VQMT visualization



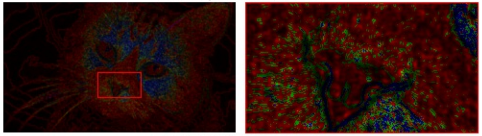
JPEG Q=10, 3SSIM-y=0.7925



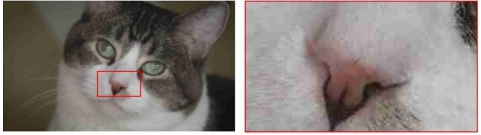
MSU VQMT visualization



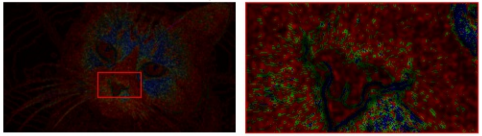
JPEG Q=15, 3SSIM-y=0.8524



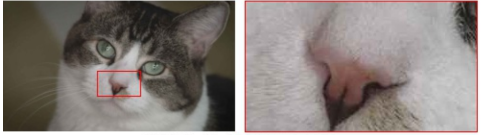
MSU VQMT visualization



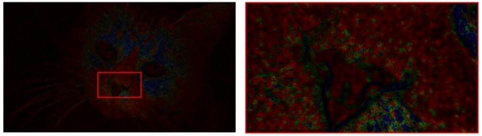
JPEG Q=20, 3SSIM-y=0.8856



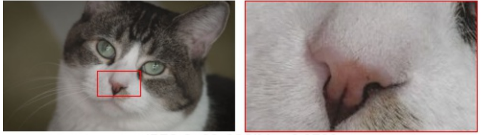
MSU VQMT visualization



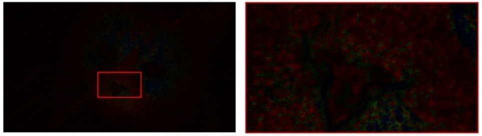
JPEG Q=40, 3SSIM-y=0.9378



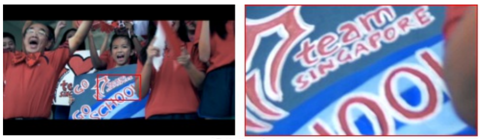
MSU VQMT visualization



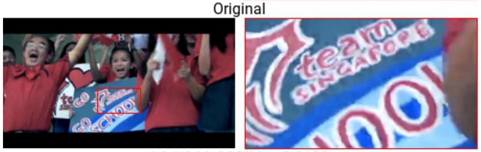
JPEG Q=80, 3SSIM-y=0.9788



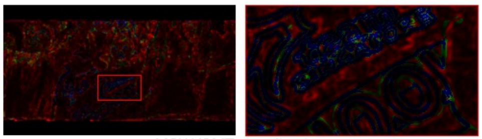
MSU VQMT visualization



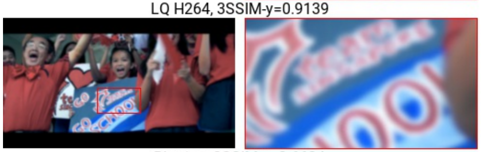
Original



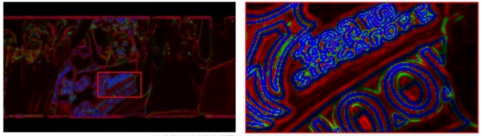
LQ H264, 3SSIM-y=0.9139



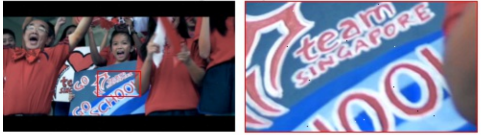
MSU VQMT visualization



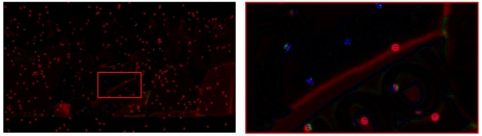
Blurring, 3SSIM-y=0.6624



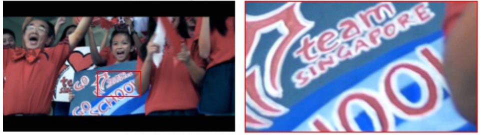
MSU VQMT visualization



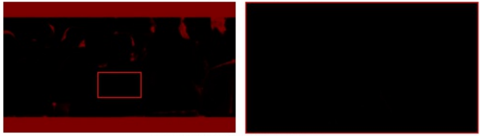
Random points, 3SSIM-y=0.9662



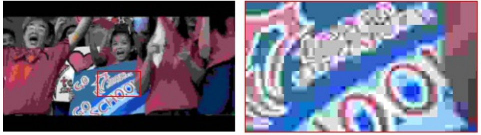
MSU VQMT visualization



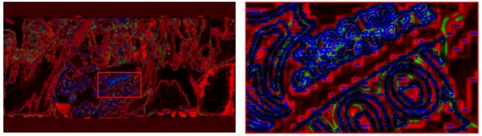
Luminance shift, 3SSIM-y=0.9834



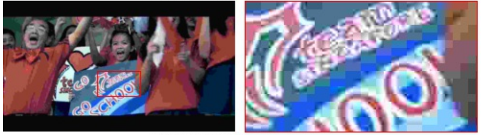
MSU VQMT visualization



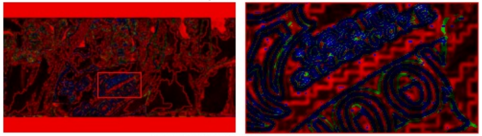
JPEG Q=2, 3SSIM-y=0.6776



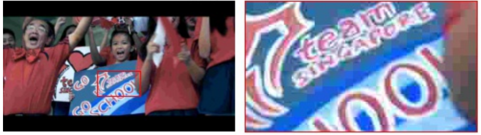
MSU VQMT visualization



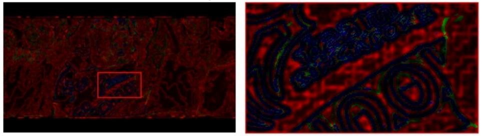
JPEG Q=5, 3SSIM-y=0.8164



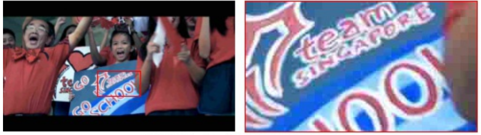
MSU VQMT visualization



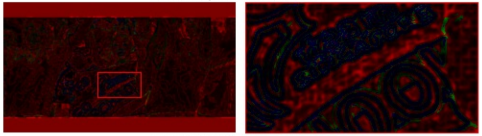
JPEG Q=10, 3SSIM-y=0.9310



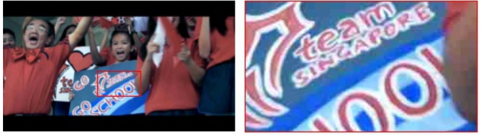
MSU VQMT visualization



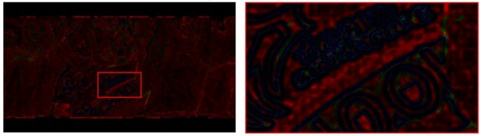
JPEG Q=15, 3SSIM-y=0.9376



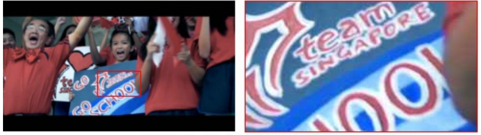
MSU VQMT visualization



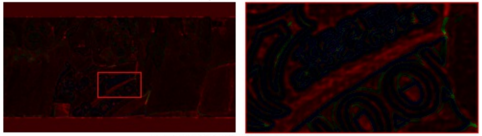
JPEG Q=20, 3SSIM-y=0.9626



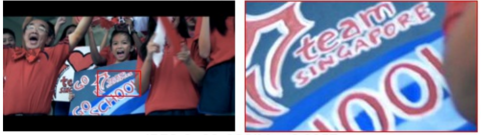
MSU VQMT visualization



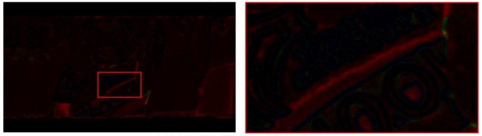
JPEG Q=40, 3SSIM-y=0.9717



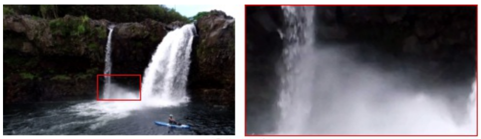
MSU VQMT visualization



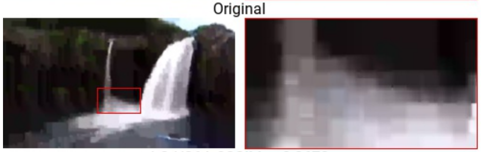
JPEG Q=80, 3SSIM-y=0.9857



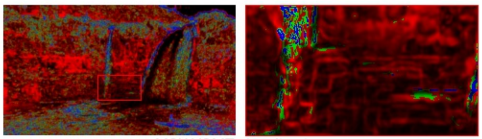
MSU VQMT visualization



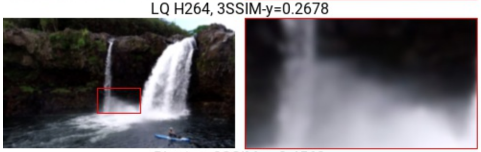
Original



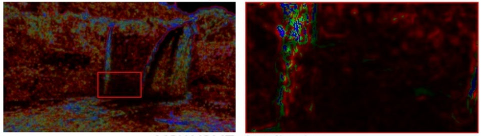
LQ H264, 3SSIM-y=0.2678



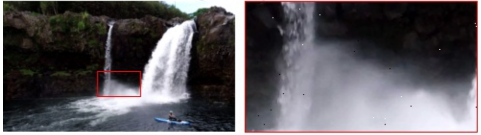
MSU VQMT visualization



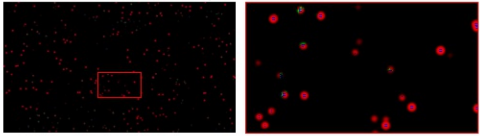
Blurring, 3SSIM-y=0.4563



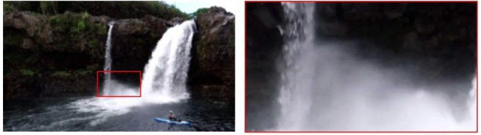
MSU VQMT visualization



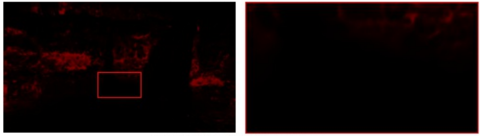
Random points, 3SSIM-y=0.9822



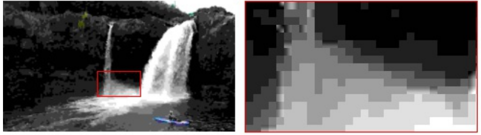
MSU VQMT visualization



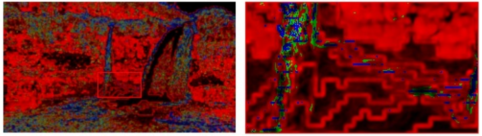
Luminance shift, 3SSIM-y=0.9958



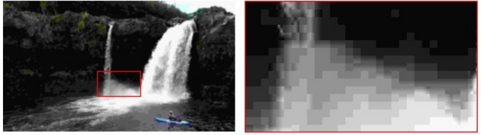
MSU VQMT visualization



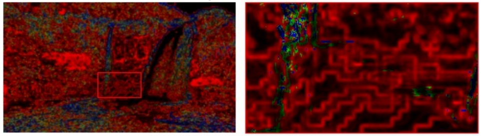
JPEG Q=2, 3SSIM-y=0.4640



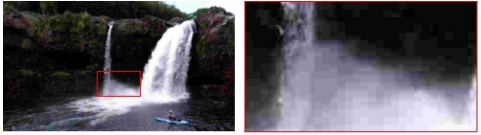
MSU VQMT visualization



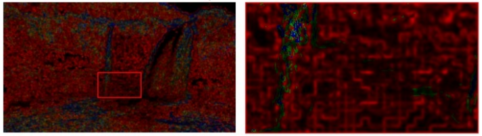
JPEG Q=5, 3SSIM-y=0.6538



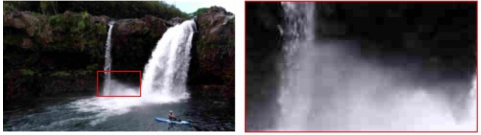
MSU VQMT visualization



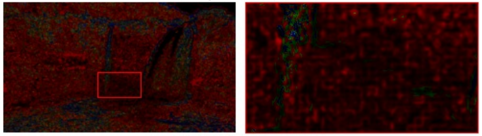
JPEG Q=10, 3SSIM-y=0.8012



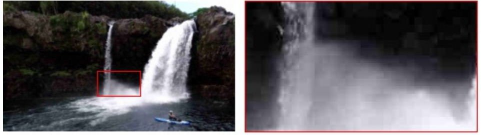
MSU VQMT visualization



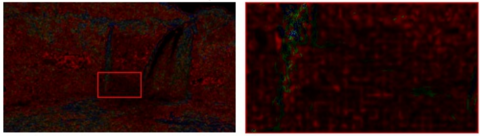
JPEG Q=15, 3SSIM-y=0.8591



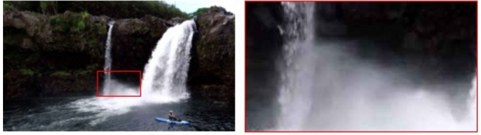
MSU VQMT visualization



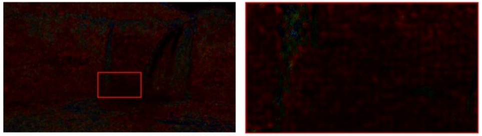
JPEG Q=20, 3SSIM-y=0.8876



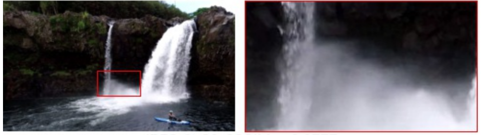
MSU VQMT visualization



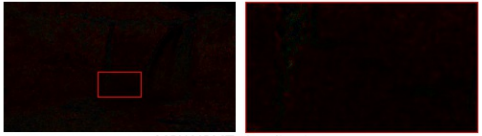
JPEG Q=40, 3SSIM-y=0.9366



MSU VQMT visualization



JPEG Q=80, 3SSIM-y=0.9777



MSU VQMT visualization

ST-SSIM (Spatio-Temporal SSIM Index)

General info

Metric type	full-reference temporal metric
MSU VQMT implementations	this metric was temporary excluded from VQMT due to unstability of results
External links	original paper (A. K. Moorthy and A. C. Bovik)

Algorithm description

The idea of this algorithm is to use motion-oriented weighted windows for SSIM Index. MSU Motion Estimation algorithm is used to retrieve this information. Based on the ME results, weighting window is constructed for every pixel. This window can use up to 33 consecutive frames (16 + current frame + 16). Then SSIM Index is calculated for every window to take into account temporal distortions as well. In addition, another spooling technique is used in this implementation. We use only lower 6% of metric values for the frame to calculate frame metric value. This causes larger metric values difference for difference files.

Benchmark

Please note: values can vary depending on system configuration, input format and other factors.

**Benchmark on VQMT 14.0 BETA r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti**

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame

**Benchmark on VQMT 14.0 BETA r12792 PRO for Linux.
CPU: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64 cores
GPU: NVIDIA CUDA/TITAN RTX**

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame

**Benchmark on VQMT 13.1 r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti**

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame

Norm calculation metrics

Delta

General info

Metric type	full-reference image metric
Value range	(original is much brigther) -1..1 (distorted is much brigther)
Value interpretation	bigger is darker original, 0 - same brightness
MSU VQMT implementations	CPU multithreaded
MSU VQMT visualization	pixel-wise
Available colorspaces	R, G, B, Y, U, V, L
Output values	metric value
Aggregated values	standard set
MSU VQMT usages	-metr delta [over <color components>]
Other names	mean difference

Algorithm description

The value of this metric is the mean difference of the color value in the corresponding points of image.

$$d(X,Y) = \frac{\sum\limits_{i=1,j=1}^{m,n} Y_{i,j} - X_{i,j}}{mn}$$

where m – video width, n – video height, image data are in range 0..1. This formula can be rewriten to the following way: $\text{mean distorted brightness} - \text{mean original brightness}$.

This metric doesn't show a quality loss, because it can be 0 for completely different images, but you can detect general brightness shifts using this metric if you are sure images have same structure.

Benchmark

Please note: values can vary depending on system configuration, input format and other factors.

**Benchmark on VQMT 14.0 BETA r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti**

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	multithreaded	Y	HD 720p	557.44	0.003
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	272.49	0.005
VQMT 14.0 default	multithreaded	Y	4K 2160p	58.9	0.019
VQMT 14.0 default	singlethreaded	Y	HD 720p	179.48	0.007

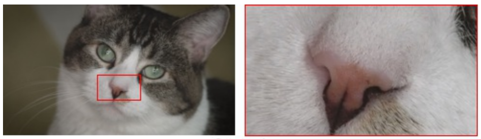
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	88.64	0.012
VQMT 14.0 default	singlethreaded	Y	4K 2160p	22.82	0.046

Benchmark on VQMT 14.0 BETA r12792 PRO for Linux.
CPU: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64 cores
GPU: NVIDIA CUDA/TITAN RTX

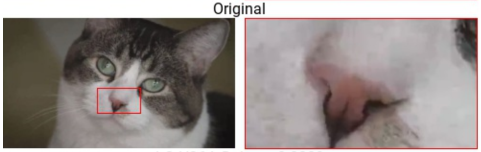
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	multithreaded	Y	HD 720p	1271.1	0.002
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	646.32	0.004
VQMT 14.0 default	multithreaded	Y	4K 2160p	119.24	0.016
VQMT 14.0 default	singlethreaded	Y	HD 720p	152.69	0.007
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	66.5	0.015
VQMT 14.0 default	singlethreaded	Y	4K 2160p	20.66	0.05

Benchmark on VQMT 13.1 r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

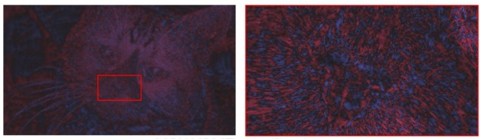
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 default	multithreaded	Y	HD 720p	568.83	0.003
VQMT 13.1 default	multithreaded	Y	FullHD 1080p	260.59	0.005
VQMT 13.1 default	multithreaded	Y	4K 2160p	54.22	0.02
VQMT 13.1 default	singlethreaded	Y	HD 720p	181.75	0.006
VQMT 13.1 default	singlethreaded	Y	FullHD 1080p	88.77	0.012
VQMT 13.1 default	singlethreaded	Y	4K 2160p	23.25	0.045



Original

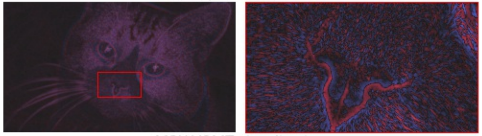


LQ H264, Delta-y=0.0009



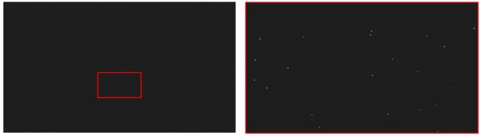
MSU VQMT visualization

Blurring, Delta-y=0.0003

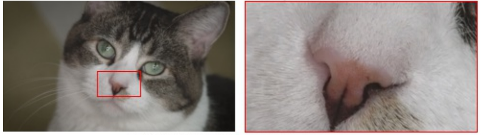


MSU VQMT visualization

Random points, Delta-y=0.0001

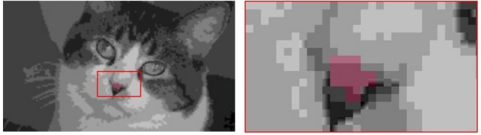


MSU VQMT visualization

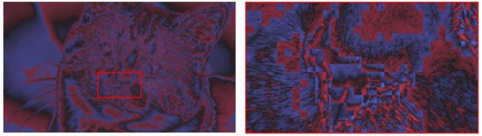


Luminance shift, Delta-y=0.0078

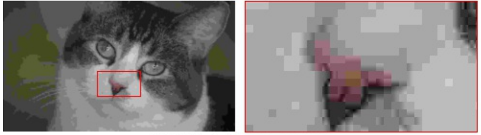
MSU VQMT visualization



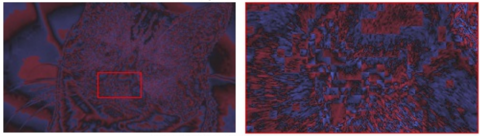
JPEG Q=2, Delta-y=-0.0019



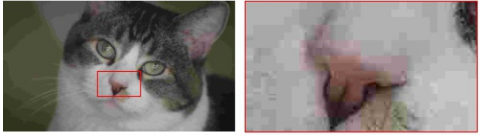
MSU VQMT visualization



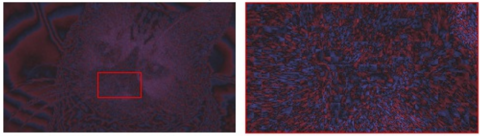
JPEG Q=5, Delta-y=-0.0003



MSU VQMT visualization

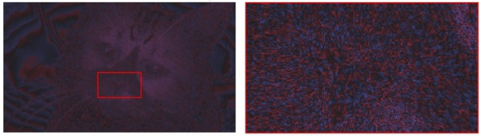


JPEG Q=10, Delta-y=0.0015



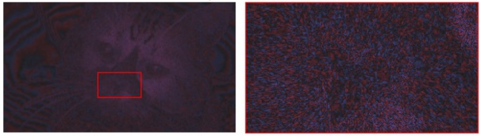
MSU VQMT visualization

JPEG Q=15, Delta-y=-0.0009



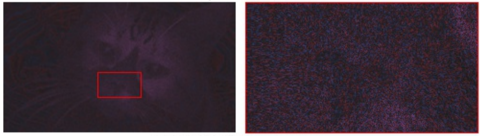
MSU VQMT visualization

JPEG Q=20, Delta-y=-0.0002

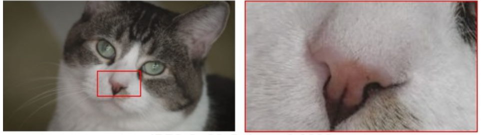


MSU VQMT visualization

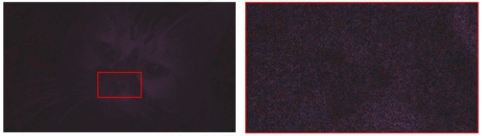
JPEG Q=40, Delta-y=0.0002



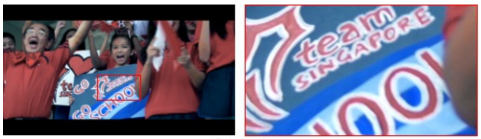
MSU VQMT visualization



JPEG Q=80, Delta-y=0.0003

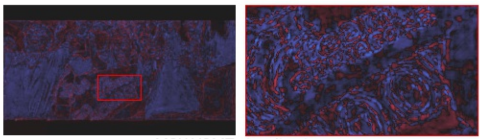


MSU VQMT visualization

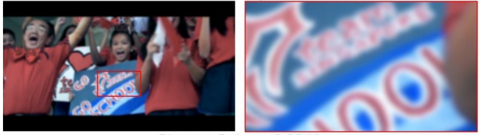


Original

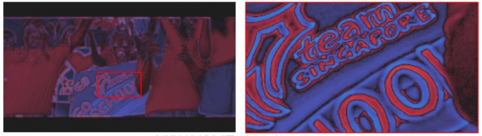
LQ H264, Delta-y=-0.0093



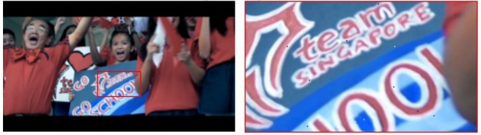
MSU VQMT visualization



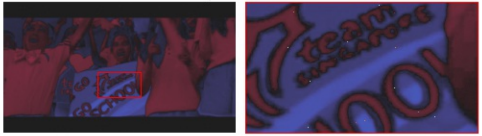
Blurring, Delta-y=0.0046



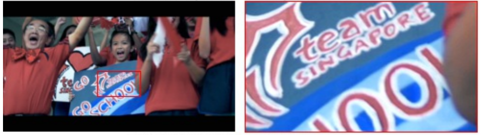
MSU VQMT visualization



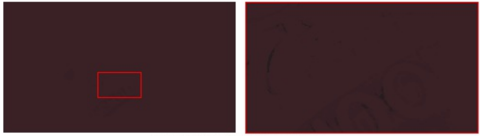
Random points, Delta-y=0.0042



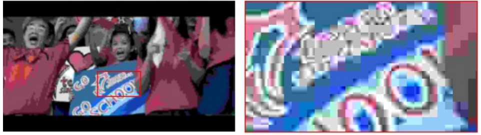
MSU VQMT visualization



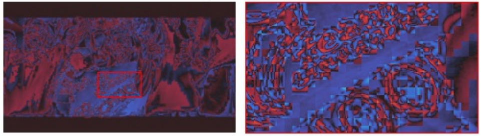
Luminance shift, Delta-y=0.0078



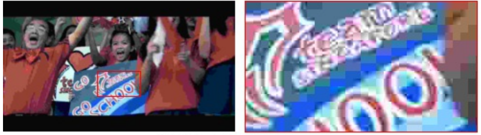
MSU VQMT visualization



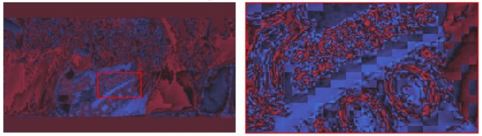
JPEG Q=2, Delta-y=0.0047



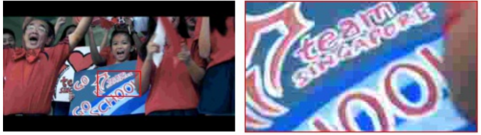
MSU VQMT visualization



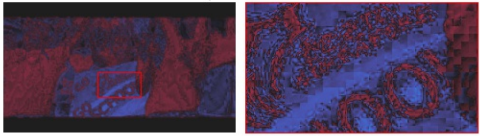
JPEG Q=5, Delta-y=0.0133



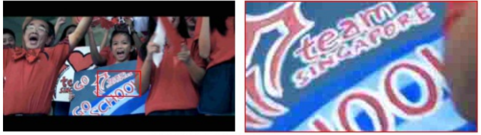
MSU VQMT visualization



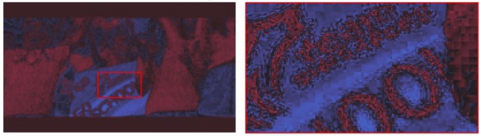
JPEG Q=10, Delta-y=0.0044



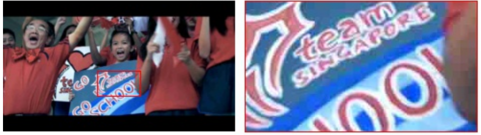
MSU VQMT visualization



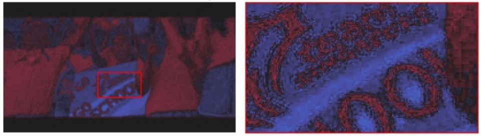
JPEG Q=15, Delta-y=0.0063



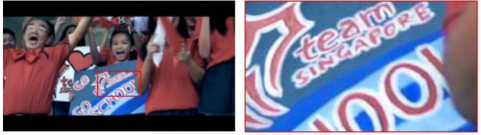
MSU VQMT visualization



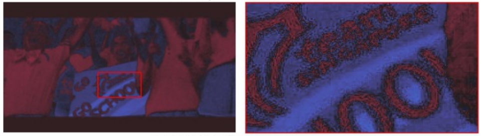
JPEG Q=20, Delta-y=0.0044



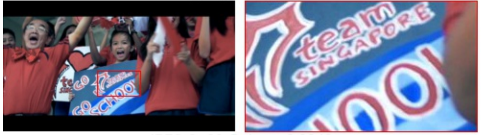
MSU VQMT visualization



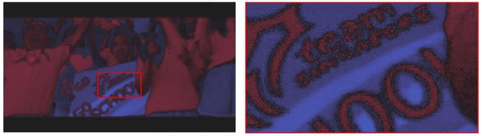
JPEG Q=40, Delta-y=0.0053



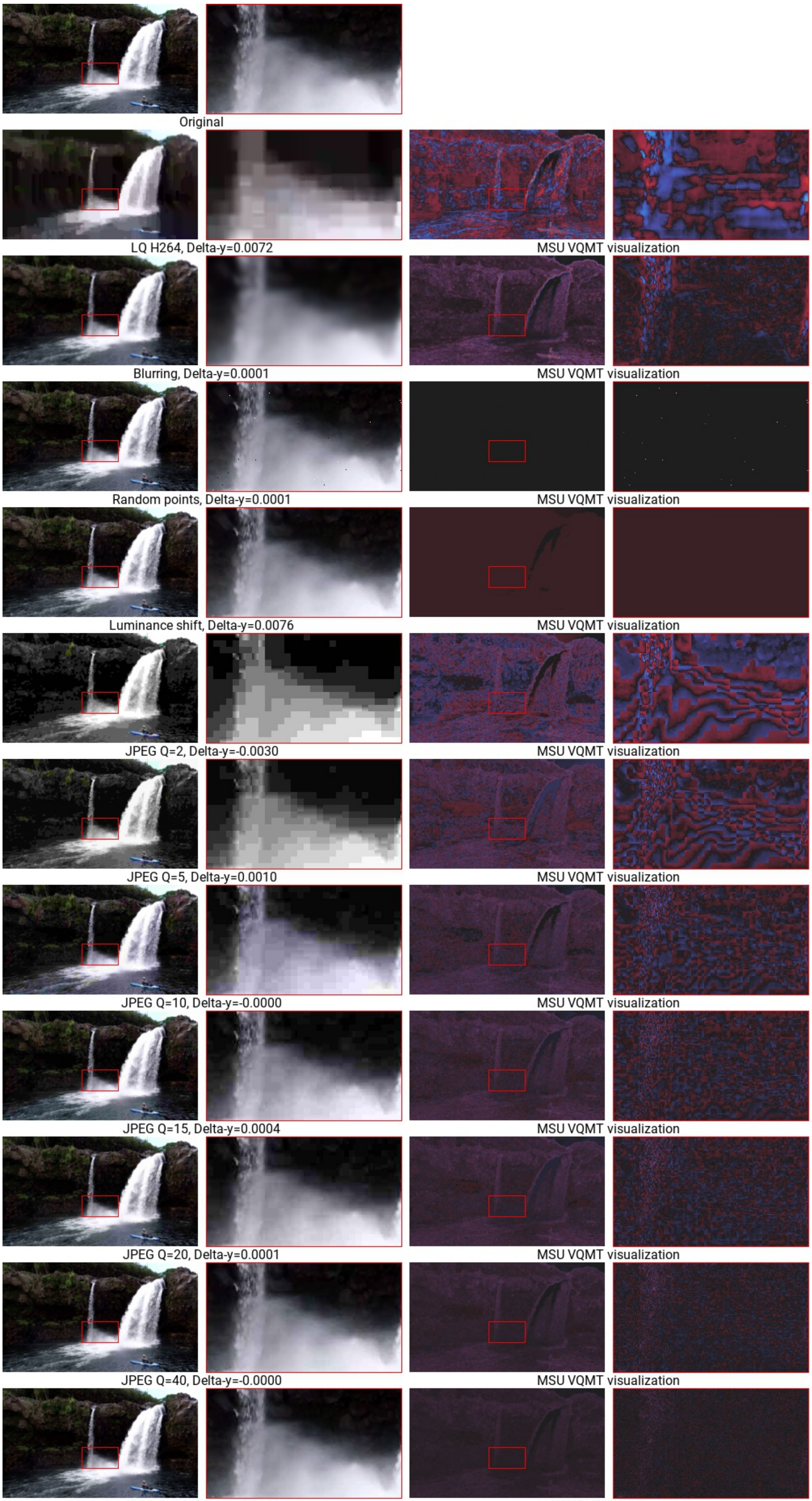
MSU VQMT visualization



JPEG Q=80, Delta-y=0.0042



MSU VQMT visualization



Original

LQ H264, Delta-y=0.0072

MSU VQMT visualization

Blurring, Delta-y=0.0001

MSU VQMT visualization

Random points, Delta-y=0.0001

MSU VQMT visualization

Luminance shift, Delta-y=0.0076

MSU VQMT visualization

JPEG Q=2, Delta-y=-0.0030

MSU VQMT visualization

JPEG Q=5, Delta-y=0.0010

MSU VQMT visualization

JPEG Q=10, Delta-y=-0.0000

MSU VQMT visualization

JPEG Q=15, Delta-y=0.0004

MSU VQMT visualization

JPEG Q=20, Delta-y=0.0001

MSU VQMT visualization

JPEG Q=40, Delta-y=-0.0000

MSU VQMT visualization

JPEG Q=80, Delta-y=0.0000

MSU VQMT visualization

Identity

General info

Metric type	full-reference image metric
Value range	(images are different) 0..1 (images are same)
Value interpretation	binary mode: 1 - images are same, 0 - images are not same; pixel mode: proportion of same pixels
MSU VQMT implementations	CPU multithreaded
MSU VQMT visualization	pixel-wise
Available colorspaces	R, G, B, Y, U, V, L, RGB, YUV
Output values	metric value
Aggregated values	standard set
MSU VQMT usages	-metr identity [over <color components>]
Other names	inverse L∞-norm

Algorithm description

Metric cas to modes: binary and pixels. In binary mode only two values are possible: 1 if images are pixel-wise similar, 0 if images have at least 1 different value pixel.

In pixel mode the value is proporsion of similar pixels. 1 means all pixels are similar, 0 means all pixels are dirrefent.

Benchmark

Please note: values can vary depending on system configuration, input format and other factors.

**Benchmark on VQMT 14.0 BETA r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti**

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	multithreaded	Y	HD 720p	587.08	0.003
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	286.13	0.005
VQMT 14.0 default	multithreaded	Y	4K 2160p	60.78	0.019
VQMT 14.0 default	multithreaded	YUV	HD 720p	456.22	0.003
VQMT 14.0 default	multithreaded	YUV	FullHD 1080p	217.33	0.006
VQMT 14.0 default	multithreaded	YUV	4K 2160p	48.15	0.024
VQMT 14.0 default	singlethreaded	Y	HD 720p	183.14	0.006
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	90.95	0.012
VQMT 14.0 default	singlethreaded	Y	4K 2160p	22.98	0.046
VQMT 14.0 default	singlethreaded	YUV	HD 720p	143.53	0.008

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	singlethreaded	YUV	FullHD 1080p	70.54	0.015
VQMT 14.0 default	singlethreaded	YUV	4K 2160p	24.26	0.044

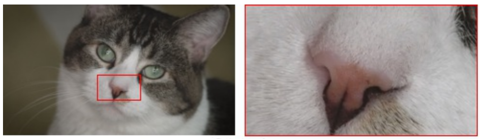
Benchmark on VQMT 14.0 BETA r12792 PRO for Linux.
CPU: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64 cores
GPU: NVIDIA CUDA/TITAN RTX

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	multithreaded	Y	HD 720p	1242.42	0.002
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	766.85	0.004
VQMT 14.0 default	multithreaded	Y	4K 2160p	85.69	0.02
VQMT 14.0 default	multithreaded	YUV	HD 720p	1203.94	0.002
VQMT 14.0 default	multithreaded	YUV	FullHD 1080p	614.29	0.005
VQMT 14.0 default	multithreaded	YUV	4K 2160p	53.1	0.031
VQMT 14.0 default	singlethreaded	Y	HD 720p	177.72	0.006
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	81.29	0.012
VQMT 14.0 default	singlethreaded	Y	4K 2160p	23.69	0.043
VQMT 14.0 default	singlethreaded	YUV	HD 720p	148.95	0.007
VQMT 14.0 default	singlethreaded	YUV	FullHD 1080p	71.48	0.014
VQMT 14.0 default	singlethreaded	YUV	4K 2160p	21.52	0.048

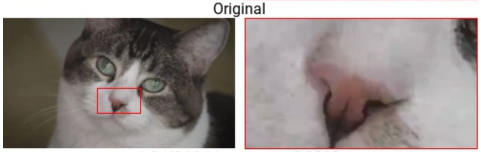
Benchmark on VQMT 13.1 r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 default	multithreaded	Y	HD 720p	600.62	0.002
VQMT 13.1 default	multithreaded	Y	FullHD 1080p	272.64	0.005
VQMT 13.1 default	multithreaded	Y	4K 2160p	56.1	0.02
VQMT 13.1 default	multithreaded	YUV	HD 720p	462.14	0.003
VQMT 13.1 default	multithreaded	YUV	FullHD 1080p	212.43	0.006
VQMT 13.1 default	multithreaded	YUV	4K 2160p	45.26	0.025
VQMT 13.1 default	singlethreaded	Y	HD 720p	185.49	0.006
VQMT 13.1 default	singlethreaded	Y	FullHD 1080p	91.03	0.012

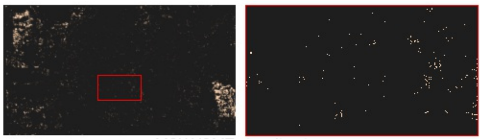
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 default	singlethreaded	Y	4K 2160p	23.56	0.044
VQMT 13.1 default	singlethreaded	YUV	HD 720p	142.98	0.008
VQMT 13.1 default	singlethreaded	YUV	FullHD 1080p	69.82	0.015
VQMT 13.1 default	singlethreaded	YUV	4K 2160p	23.44	0.045



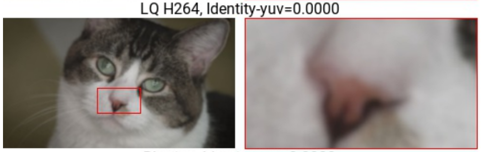
Original



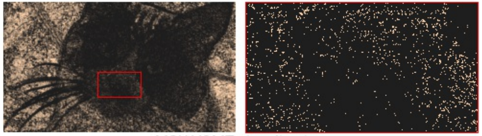
LQ H264, Identity-yuv=0.0000



MSU VQMT visualization

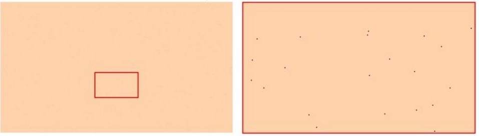


Blurring, Identity-yuv=0.0000

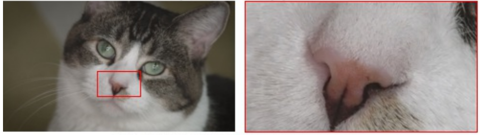


MSU VQMT visualization

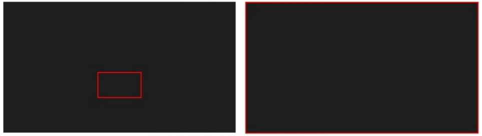
Random points, Identity-yuv=0.0000



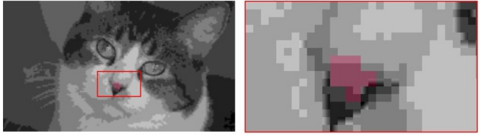
MSU VQMT visualization



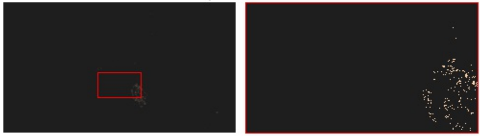
Luminance shift, Identity-yuv=0.0000



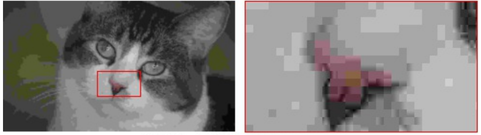
MSU VQMT visualization



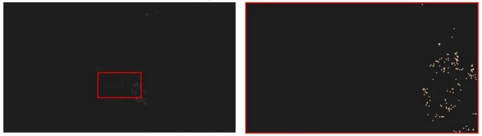
JPEG Q=2, Identity-yuv=0.0000



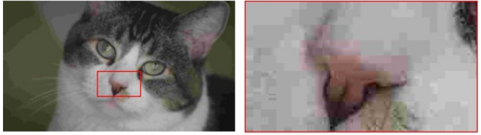
MSU VQMT visualization



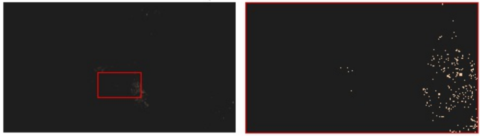
JPEG Q=5, Identity-yuv=0.0000



MSU VQMT visualization

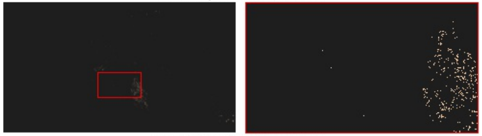


JPEG Q=10, Identity-yuv=0.0000

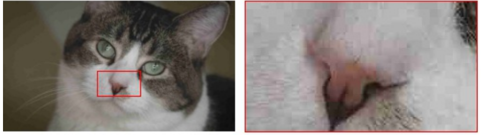


MSU VQMT visualization

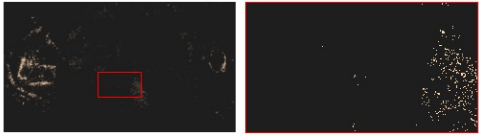
JPEG Q=15, Identity-yuv=0.0000



MSU VQMT visualization

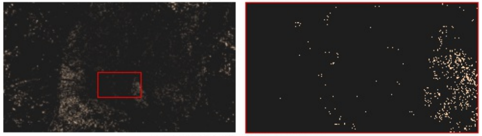


JPEG Q=20, Identity-yuv=0.0000

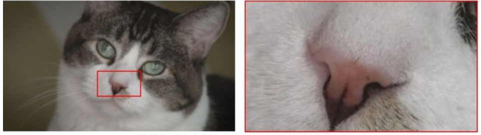


MSU VQMT visualization

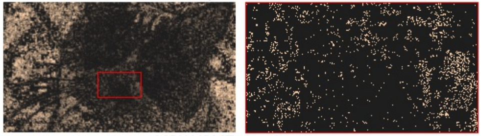
JPEG Q=40, Identity-yuv=0.0000



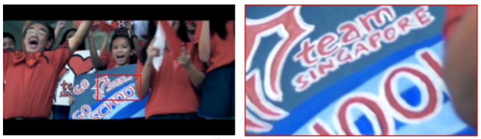
MSU VQMT visualization



JPEG Q=80, Identity-yuv=0.0000



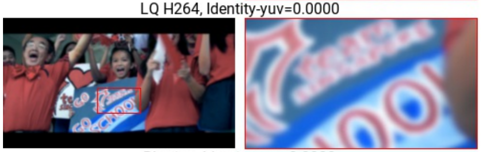
MSU VQMT visualization



Original

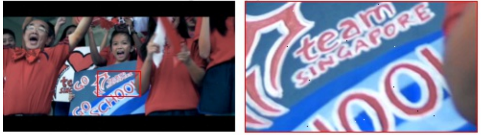
LQ H264, Identity-yuv=0.0000

MSU VQMT visualization



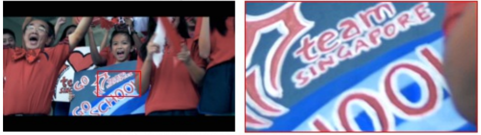
Blurring, Identity-yuv=0.0000

MSU VQMT visualization



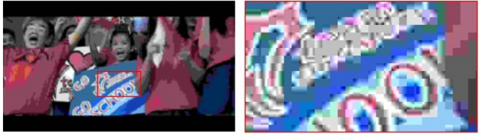
Random points, Identity-yuv=0.0000

MSU VQMT visualization



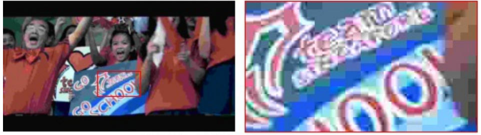
Luminance shift, Identity-yuv=0.0000

MSU VQMT visualization



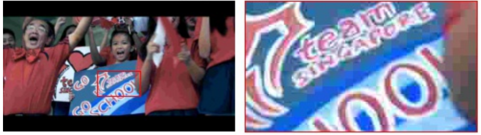
JPEG Q=2, Identity-yuv=0.0000

MSU VQMT visualization



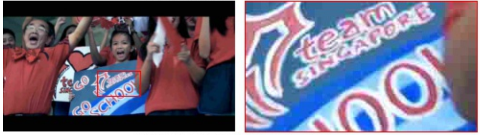
JPEG Q=5, Identity-yuv=0.0000

MSU VQMT visualization



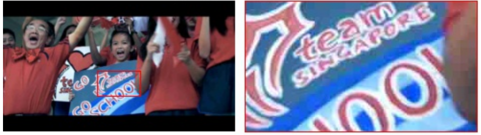
JPEG Q=10, Identity-yuv=0.0000

MSU VQMT visualization



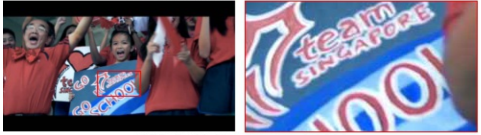
JPEG Q=15, Identity-yuv=0.0000

MSU VQMT visualization



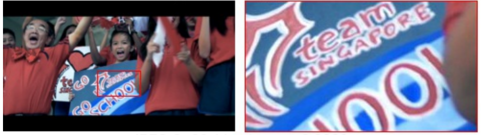
JPEG Q=20, Identity-yuv=0.0000

MSU VQMT visualization



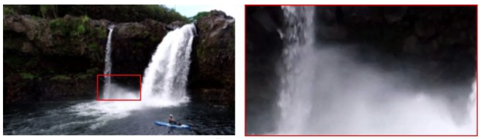
JPEG Q=40, Identity-yuv=0.0000

MSU VQMT visualization

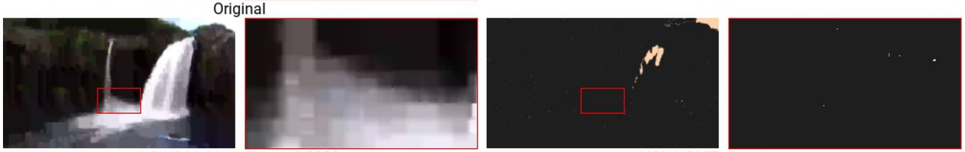


JPEG Q=80, Identity-yuv=0.0000

MSU VQMT visualization

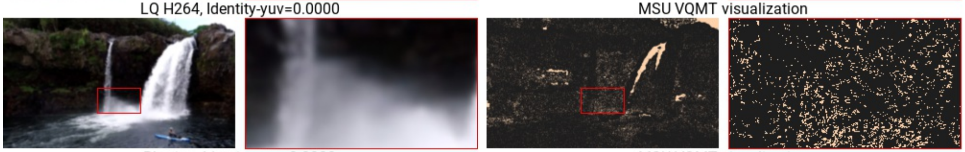


Original



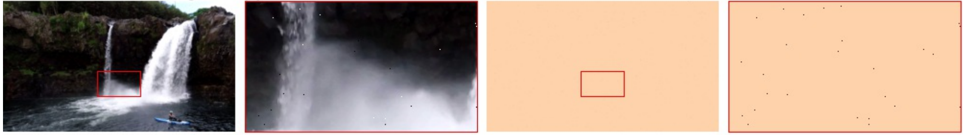
LQ H264, Identity-yuv=0.0000

MSU VQMT visualization



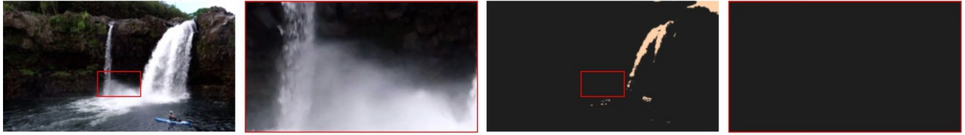
Blurring, Identity-yuv=0.0000

MSU VQMT visualization



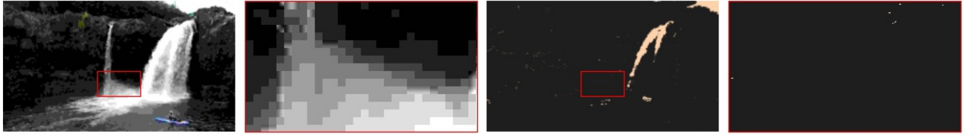
Random points, Identity-yuv=0.0000

MSU VQMT visualization



Luminance shift, Identity-yuv=0.0000

MSU VQMT visualization

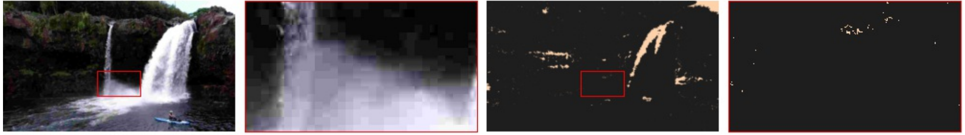


JPEG Q=2, Identity-yuv=0.0000

MSU VQMT visualization

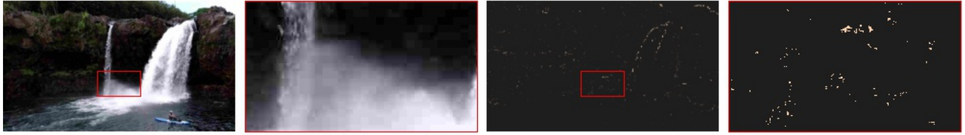
JPEG Q=5, Identity-yuv=0.0000

MSU VQMT visualization



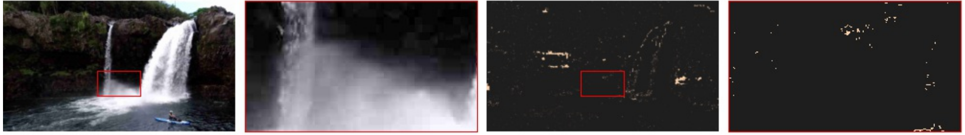
JPEG Q=10, Identity-yuv=0.0000

MSU VQMT visualization



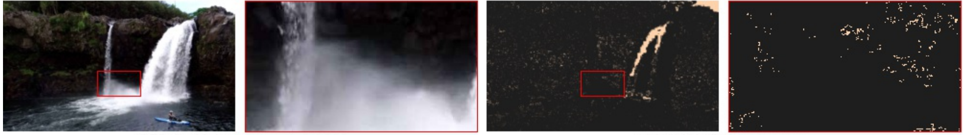
JPEG Q=15, Identity-yuv=0.0000

MSU VQMT visualization



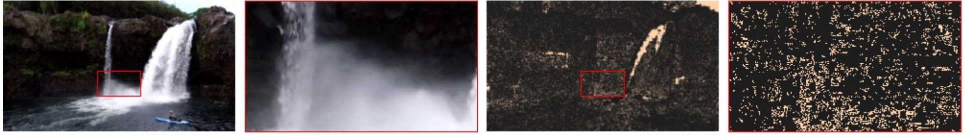
JPEG Q=20, Identity-yuv=0.0000

MSU VQMT visualization



JPEG Q=40, Identity-yuv=0.0000

MSU VQMT visualization



JPEG Q=80, Identity-yuv=0.0000

MSU VQMT visualization

MSAD (Mean Sum of Absolute Differences)

General info

Metric type	full-reference image metric
Value range	(same images) 0..1 (completely different)
Value interpretation	smaller is better quality
MSU VQMT implementations	CPU multithreaded
MSU VQMT visualization	pixel-wise
Available colorspaces	R, G, B, Y, U, V, L
Output values	metric value
Aggregated values	standard set
MSU VQMT usages	-metr msad [over <color components>]
Other names	SAD, L_1-norm
External links	Wikipedia

Algorithm description

This metric has very similar formula to Delta, but has a modulo around the difference:

$$d(X,Y) = \frac{\sum\limits_{i=1,j=1}^{m,n} |Y_{i,j} - X_{i,j}|}{mn}$$

where m – video width, n – video height, image data are in range 0..1. Metric depends only on difference of original and distorted, it is L_1-norm of this difference.

Unlike Delta, this metric will show real difference between images, 0 means completely equivalent images.

Benchmark

Please note: values can vary depending on system configuration, input format and other factors.

**Benchmark on VQMT 14.0 BETA r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti**

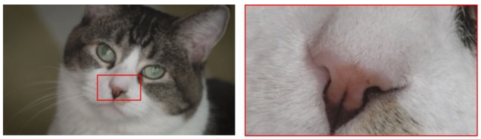
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	multithreaded	Y	HD 720p	554.19	0.003
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	270.36	0.005
VQMT 14.0 default	multithreaded	Y	4K 2160p	58.58	0.019
VQMT 14.0 default	singlethreaded	Y	HD 720p	180.56	0.006
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	89.07	0.012
VQMT 14.0 default	singlethreaded	Y	4K 2160p	22.38	0.047

Benchmark on VQMT 14.0 BETA r12792 PRO for Linux.
CPU: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64 cores
GPU: NVIDIA CUDA/TITAN RTX

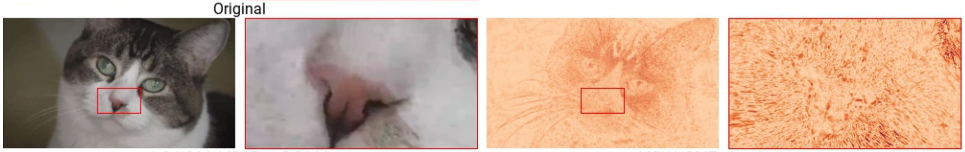
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	multithreaded	Y	HD 720p	1209.23	0.002
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	681.39	0.004
VQMT 14.0 default	multithreaded	Y	4K 2160p	86.73	0.019
VQMT 14.0 default	singlethreaded	Y	HD 720p	144.09	0.007
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	64.47	0.016
VQMT 14.0 default	singlethreaded	Y	4K 2160p	22.49	0.046

Benchmark on VQMT 13.1 r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

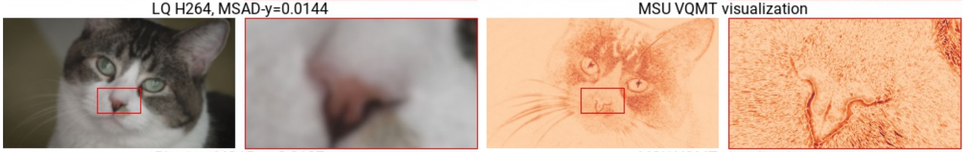
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 default	multithreaded	Y	HD 720p	568.44	0.003
VQMT 13.1 default	multithreaded	Y	FullHD 1080p	258.12	0.005
VQMT 13.1 default	multithreaded	Y	4K 2160p	54.08	0.02
VQMT 13.1 default	singlethreaded	Y	HD 720p	183.09	0.006
VQMT 13.1 default	singlethreaded	Y	FullHD 1080p	89.08	0.012
VQMT 13.1 default	singlethreaded	Y	4K 2160p	22.81	0.046



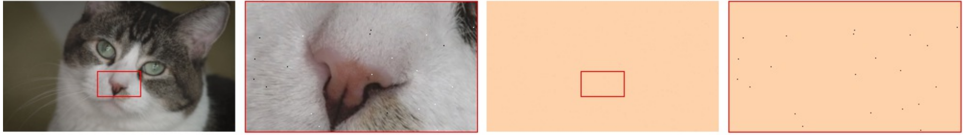
Original



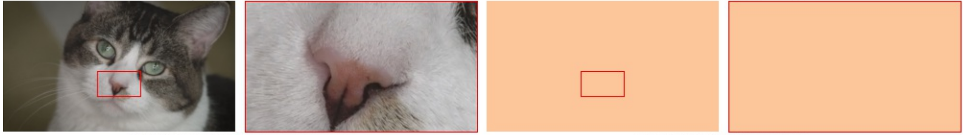
LQ H264, MSAD-y=0.0144 MSU VQMT visualization



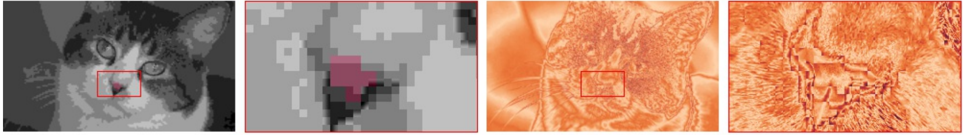
Blurring, MSAD-y=0.0127 MSU VQMT visualization



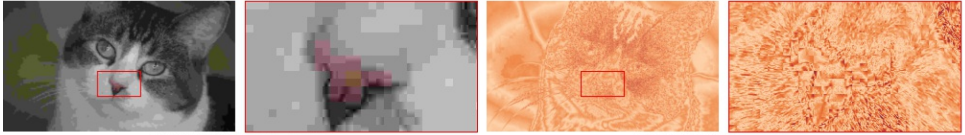
Random points, MSAD-y=0.0003 MSU VQMT visualization



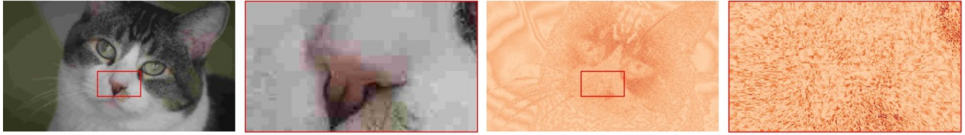
Luminance shift, MSAD-y=0.0078 MSU VQMT visualization



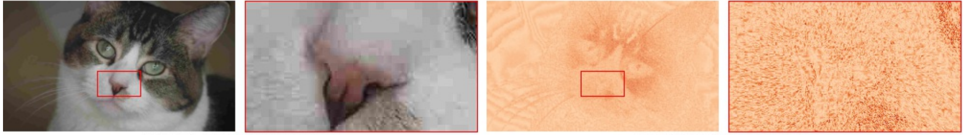
JPEG Q=2, MSAD-y=0.0374 MSU VQMT visualization



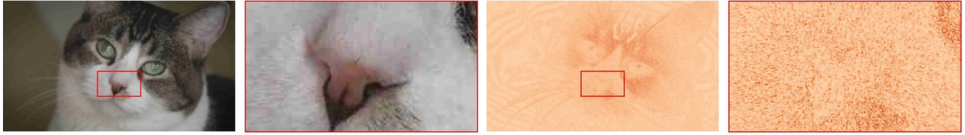
JPEG Q=5, MSAD-y=0.0259 MSU VQMT visualization



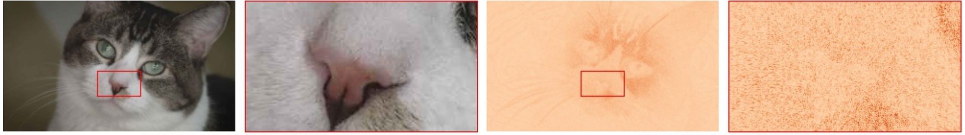
JPEG Q=10, MSAD-y=0.0160 MSU VQMT visualization



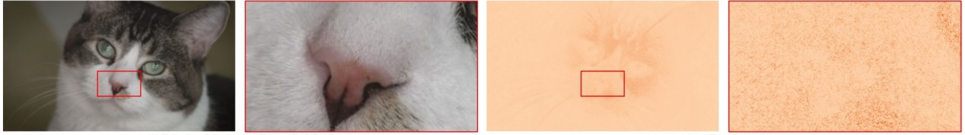
JPEG Q=15, MSAD-y=0.0128 MSU VQMT visualization



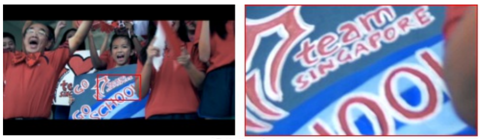
JPEG Q=20, MSAD-y=0.0107 MSU VQMT visualization



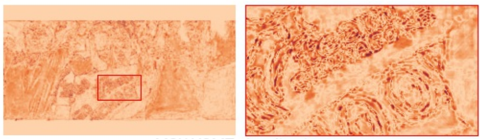
JPEG Q=40, MSAD-y=0.0074 MSU VQMT visualization



JPEG Q=80, MSAD-y=0.0043 MSU VQMT visualization

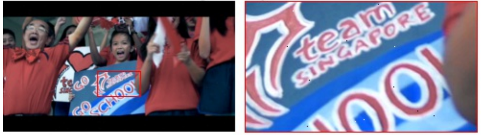

Original

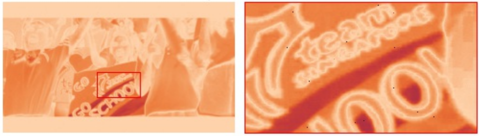
LQ H264, MSAD-y=0.0179

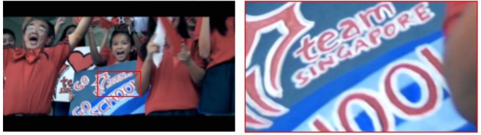

MSU VQMT visualization

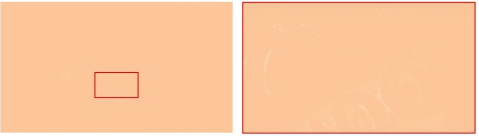
Blurring, MSAD-y=0.0330

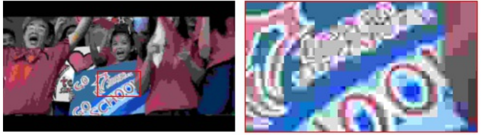
MSU VQMT visualization

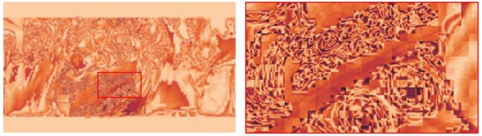

Random points, MSAD-y=0.0224

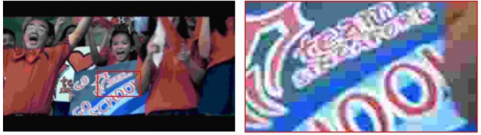

MSU VQMT visualization

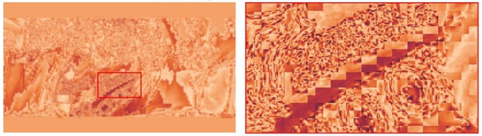

Luminance shift, MSAD-y=0.0078

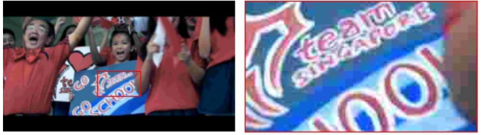

MSU VQMT visualization

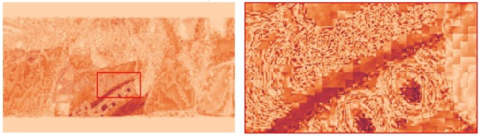

JPEG Q=2, MSAD-y=0.0379

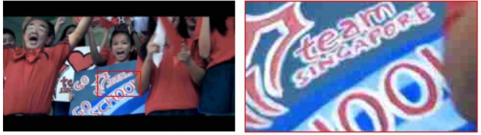

MSU VQMT visualization

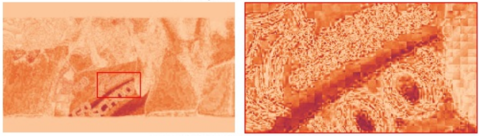

JPEG Q=5, MSAD-y=0.0378

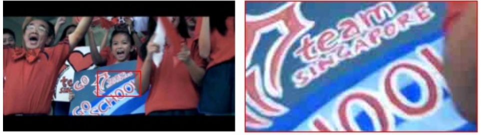

MSU VQMT visualization

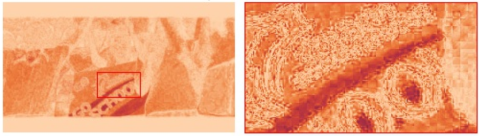

JPEG Q=10, MSAD-y=0.0257

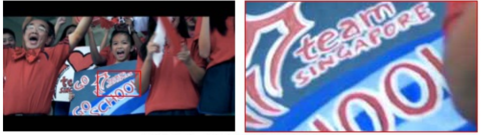

MSU VQMT visualization

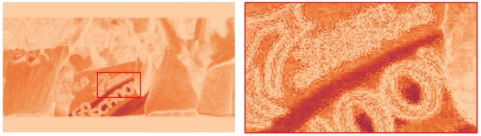

JPEG Q=15, MSAD-y=0.0264

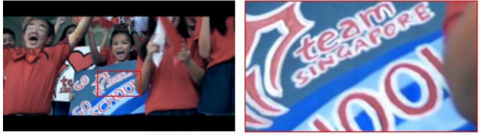

MSU VQMT visualization

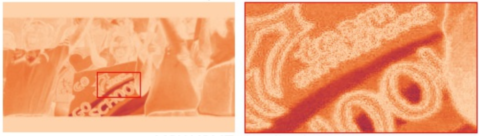

JPEG Q=20, MSAD-y=0.0242

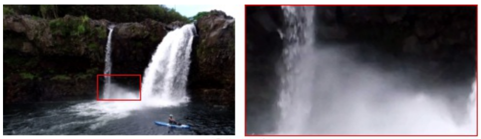

MSU VQMT visualization


JPEG Q=40, MSAD-y=0.0240

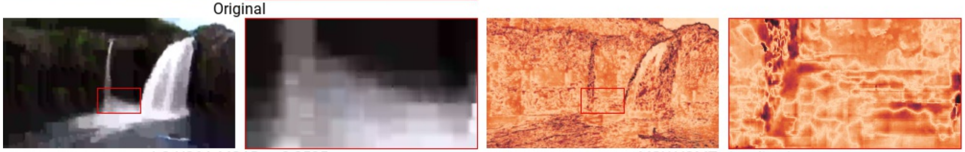

MSU VQMT visualization


JPEG Q=80, MSAD-y=0.0225

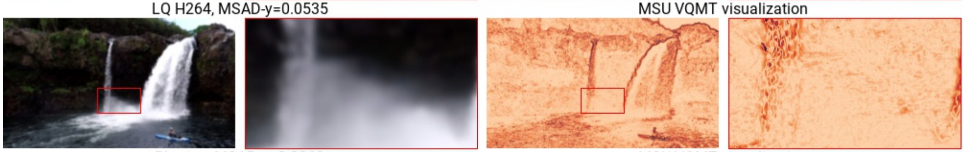

MSU VQMT visualization



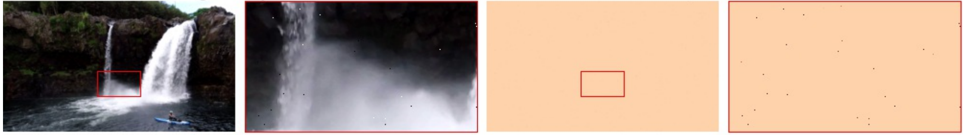
Original



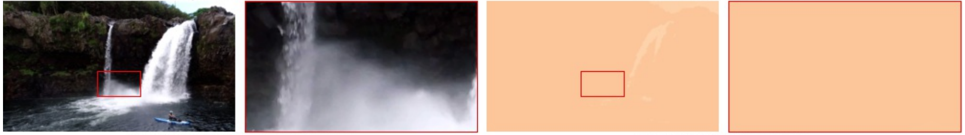
LQ H264, MSAD-y=0.0535 MSU VQMT visualization



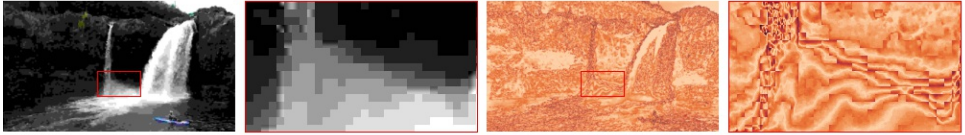
Blurring, MSAD-y=0.0268 MSU VQMT visualization



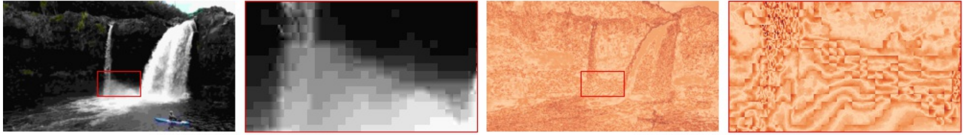
Random points, MSAD-y=0.0003 MSU VQMT visualization



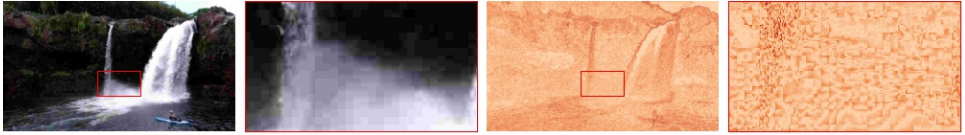
Luminance shift, MSAD-y=0.0076 MSU VQMT visualization



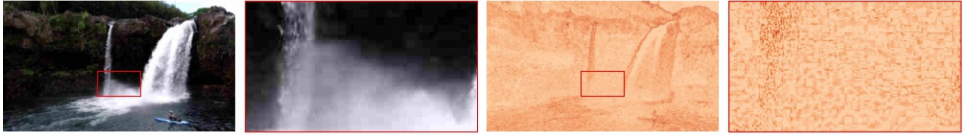
JPEG Q=2, MSAD-y=0.0429 MSU VQMT visualization



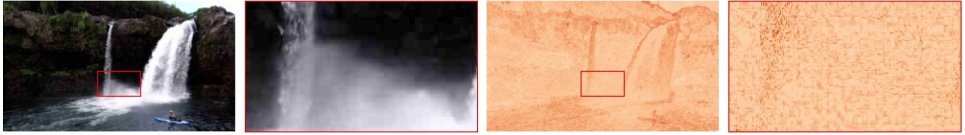
JPEG Q=5, MSAD-y=0.0320 MSU VQMT visualization



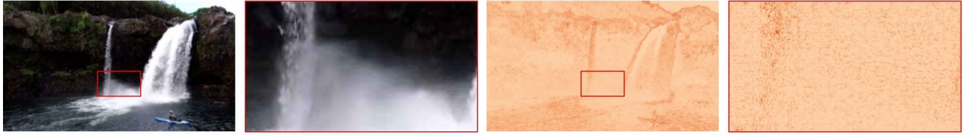
JPEG Q=10, MSAD-y=0.0223 MSU VQMT visualization



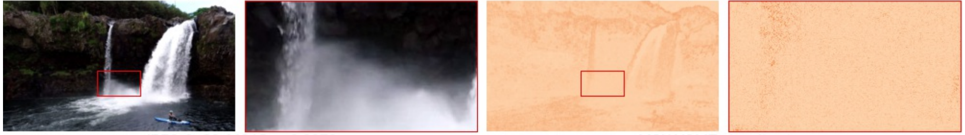
JPEG Q=15, MSAD-y=0.0184 MSU VQMT visualization



JPEG Q=20, MSAD-y=0.0163 MSU VQMT visualization



JPEG Q=40, MSAD-y=0.0120 MSU VQMT visualization



JPEG Q=80, MSAD-y=0.0073 MSU VQMT visualization

MSE (Mean Squared Error)

General info

Metric type	full-reference image metric
Value range	(same images) 0..1 (completely different)
Value interpretation	smaller is better quality
MSU VQMT implementations	CPU multithreaded
MSU VQMT visualization	pixel-wise
Available colorspaces	R, G, B, Y, U, V, L
Output values	metric value
Aggregated values	standard set
MSU VQMT usages	-metr mse [over <color components>]
Other names	L_2-norm
External links	Wikipedia

Algorithm description

Metric depends only on difference of original and distorted, it is L2-norm of this difference. Metric could be computed using the following formula:

$$d(X,Y) = \frac{\sum\limits_{i=1,j=1}^{m,n}(Y_{i,j} - X_{i,j})^2}{mn}$$

where m – video width, n – video height, image data are in range 0..1.

Benchmark

Please note: values can vary depending on system configuration, input format and other factors.

Benchmark on VQMT 14.0 BETA r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

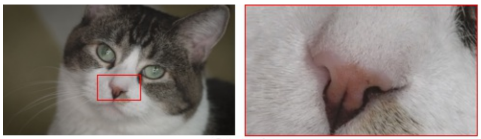
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	multithreaded	Y	HD 720p	558.26	0.003
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	270.53	0.005
VQMT 14.0 default	multithreaded	Y	4K 2160p	58.87	0.019
VQMT 14.0 default	singlethreaded	Y	HD 720p	178.24	0.007
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	88.43	0.012
VQMT 14.0 default	singlethreaded	Y	4K 2160p	22.51	0.047

Benchmark on VQMT 14.0 BETA r12792 PRO for Linux.
CPU: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64 cores
GPU: NVIDIA CUDA/TITAN RTX

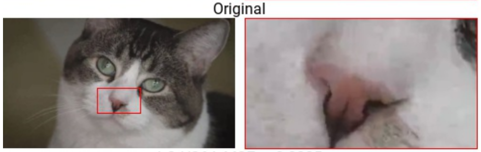
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	multithreaded	Y	HD 720p	1222.28	0.002
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	666.49	0.004
VQMT 14.0 default	multithreaded	Y	4K 2160p	84.68	0.02
VQMT 14.0 default	singlethreaded	Y	HD 720p	150.56	0.007
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	67.56	0.015
VQMT 14.0 default	singlethreaded	Y	4K 2160p	21.72	0.047

Benchmark on VQMT 13.1 r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

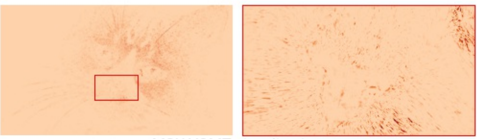
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 default	multithreaded	Y	HD 720p	565.05	0.003
VQMT 13.1 default	multithreaded	Y	FullHD 1080p	257.4	0.005
VQMT 13.1 default	multithreaded	Y	4K 2160p	54.12	0.02
VQMT 13.1 default	singlethreaded	Y	HD 720p	180.58	0.006
VQMT 13.1 default	singlethreaded	Y	FullHD 1080p	88.57	0.012
VQMT 13.1 default	singlethreaded	Y	4K 2160p	22.55	0.046



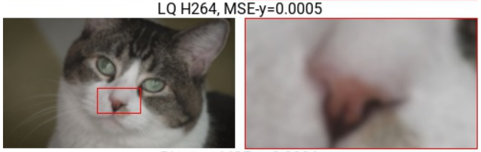
Original



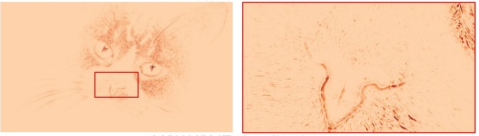
LQ H264, MSE-y=0.0005



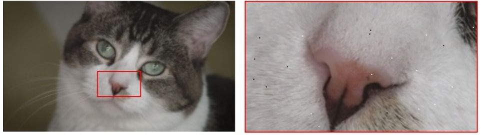
MSU VQMT visualization



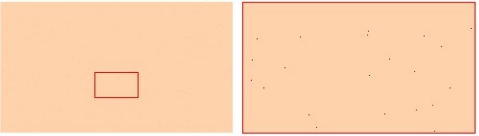
Blurring, MSE-y=0.0006



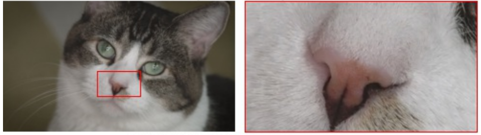
MSU VQMT visualization



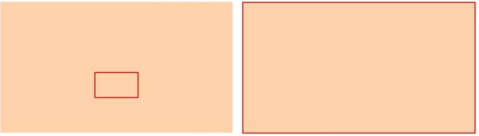
Random points, MSE-y=0.0002



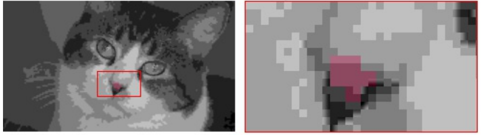
MSU VQMT visualization



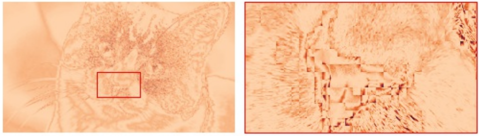
Luminance shift, MSE-y=0.0001



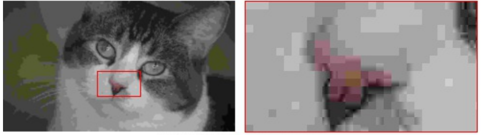
MSU VQMT visualization



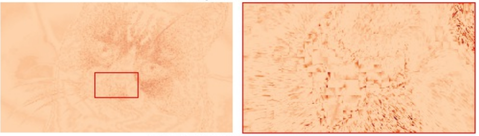
JPEG Q=2, MSE-y=0.0022



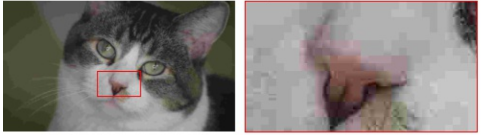
MSU VQMT visualization



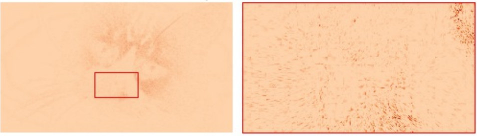
JPEG Q=5, MSE-y=0.0011



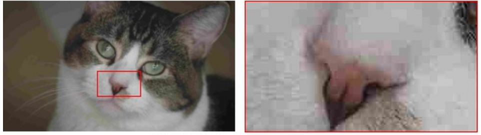
MSU VQMT visualization



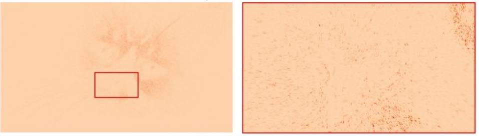
JPEG Q=10, MSE-y=0.0005



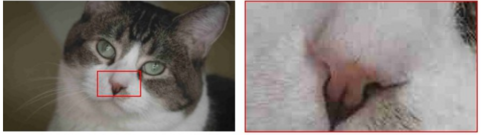
MSU VQMT visualization



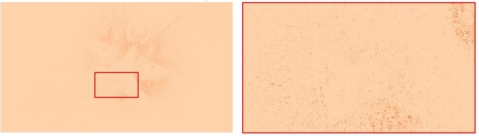
JPEG Q=15, MSE-y=0.0003



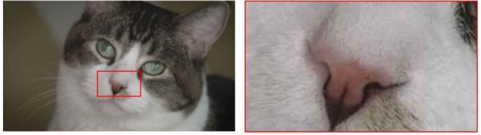
MSU VQMT visualization



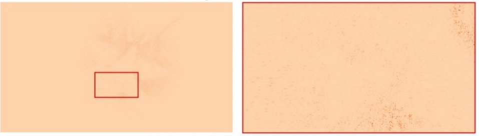
JPEG Q=20, MSE-y=0.0003



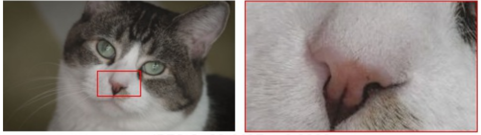
MSU VQMT visualization



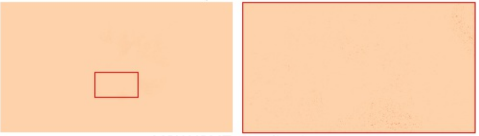
JPEG Q=40, MSE-y=0.0001



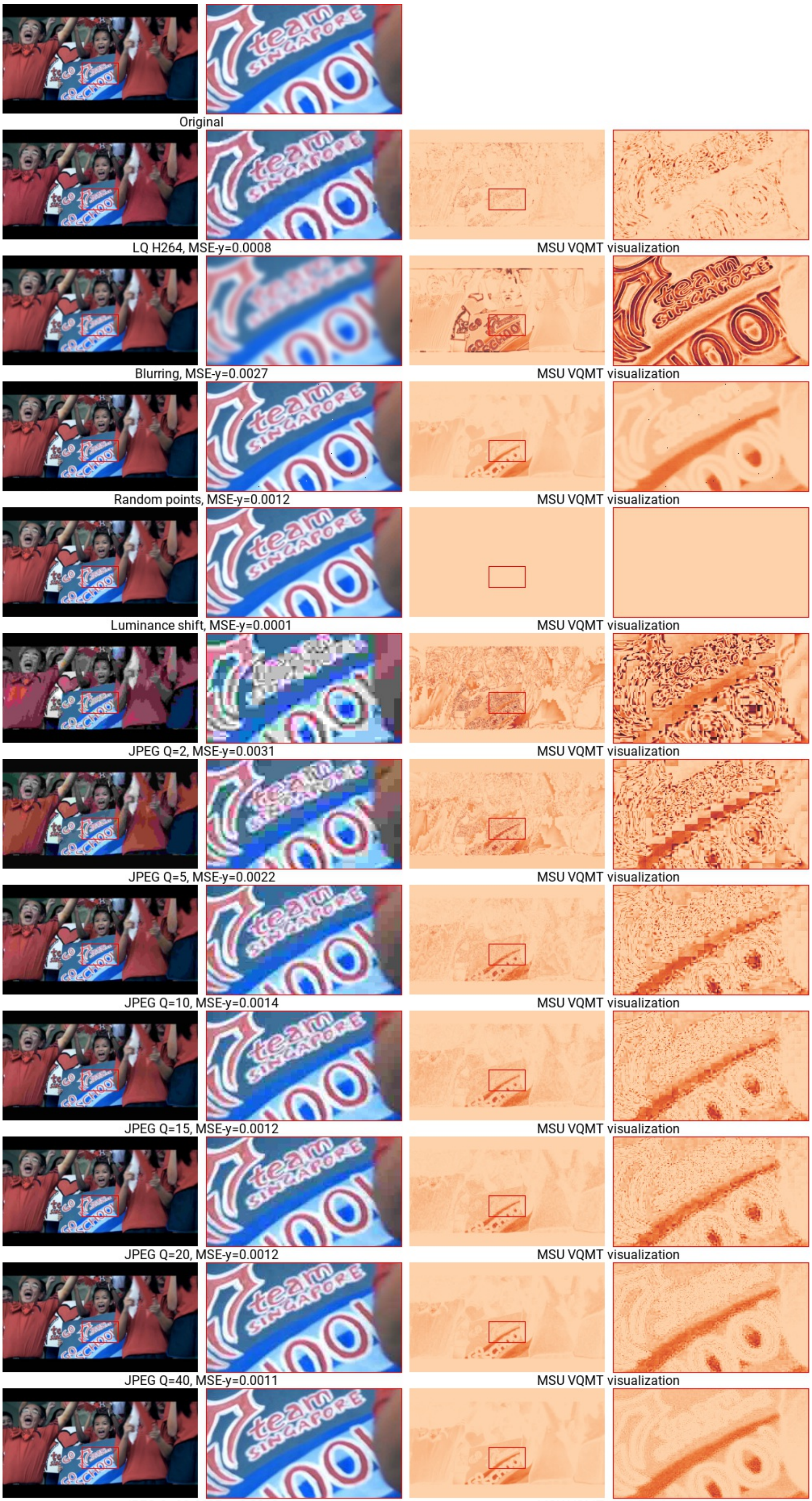
MSU VQMT visualization



JPEG Q=80, MSE-y=0.0000



MSU VQMT visualization



Original

LQ H264, MSE-y=0.0008 MSU VQMT visualization

Blurring, MSE-y=0.0027 MSU VQMT visualization

Random points, MSE-y=0.0012 MSU VQMT visualization

Luminance shift, MSE-y=0.0001 MSU VQMT visualization

JPEG Q=2, MSE-y=0.0031 MSU VQMT visualization

JPEG Q=5, MSE-y=0.0022 MSU VQMT visualization

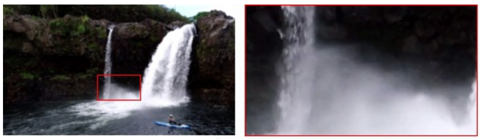
JPEG Q=10, MSE-y=0.0014 MSU VQMT visualization

JPEG Q=15, MSE-y=0.0012 MSU VQMT visualization

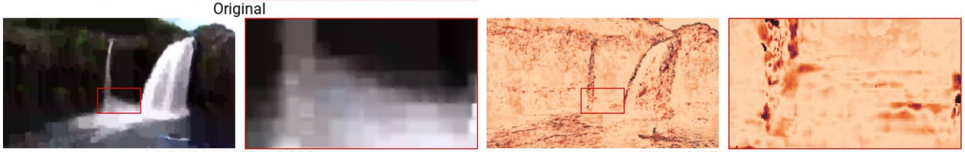
JPEG Q=20, MSE-y=0.0012 MSU VQMT visualization

JPEG Q=40, MSE-y=0.0011 MSU VQMT visualization

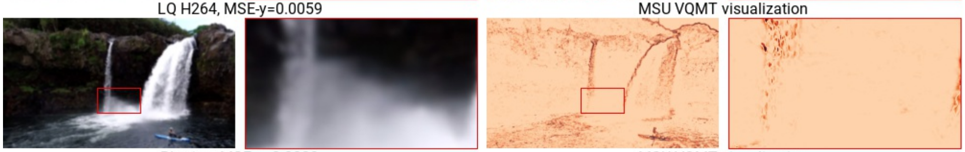
JPEG Q=80, MSE-y=0.0010 MSU VQMT visualization



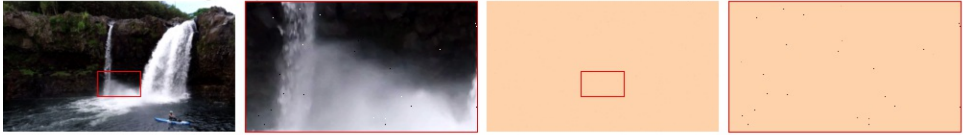
Original



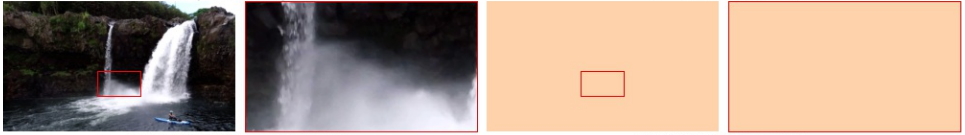
LQ H264, MSE-y=0.0059 MSU VQMT visualization



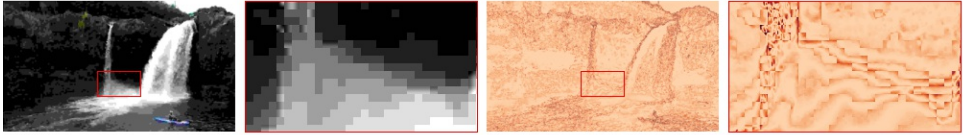
Blurring, MSE-y=0.0022 MSU VQMT visualization



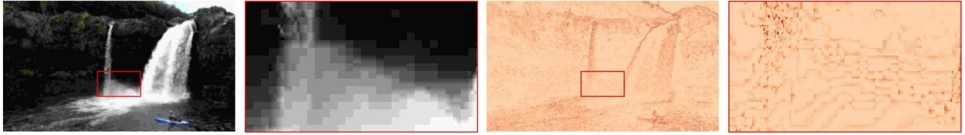
Random points, MSE-y=0.0002 MSU VQMT visualization



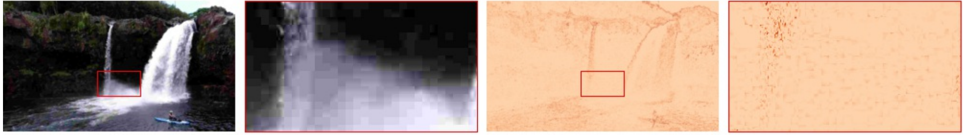
Luminance shift, MSE-y=0.0001 MSU VQMT visualization



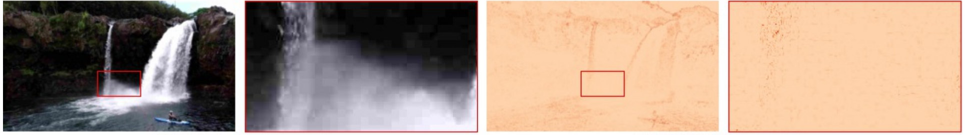
JPEG Q=2, MSE-y=0.0035 MSU VQMT visualization



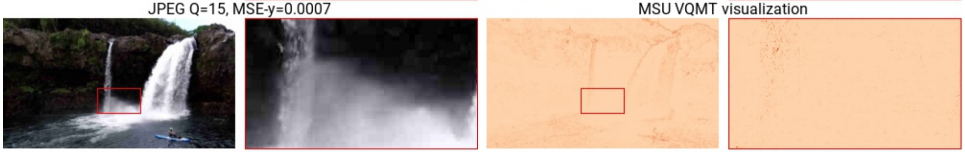
JPEG Q=5, MSE-y=0.0020 MSU VQMT visualization



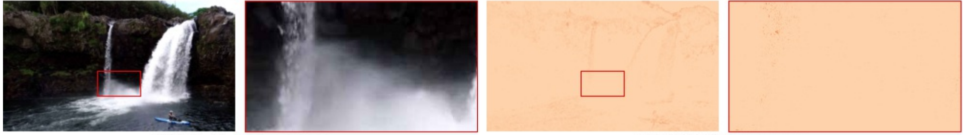
JPEG Q=10, MSE-y=0.0011 MSU VQMT visualization



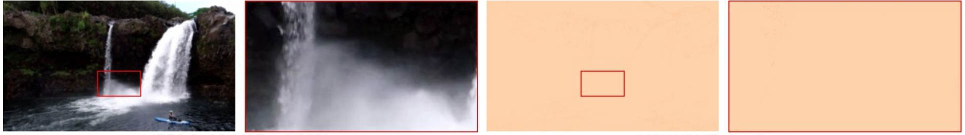
JPEG Q=15, MSE-y=0.0007 MSU VQMT visualization



JPEG Q=20, MSE-y=0.0006 MSU VQMT visualization



JPEG Q=40, MSE-y=0.0003 MSU VQMT visualization



JPEG Q=80, MSE-y=0.0001 MSU VQMT visualization

Other metrics

VQM (DCT-based Video Quality Metric)

General info

Metric type	full-reference image metric
Value range	(images are same) 0..∞ (bad quality)
Value interpretation	smaller is better quality
MSU VQMT implementations	CPU multithreaded
MSU VQMT visualization	pixel-wise
Available colorspaces	Y
Output values	metric value
Aggregated values	standard set
MSU VQMT usages	-metr vqm
External links	original paper (Feng Xiao)

Algorithm description

This metric uses discrete cosine transform (DCT) to predict human rank. It is different from widely spreaded VQM metric by ITU, that currently not implemented in VQMT. Following calculations are processed to get value of metric:

- **Color transform**. YUV color space is used for metric calculation.
- **DCT transform of blocks 8x8**. It is used to separate images into different frequencies.
- **Conversion** from DCT coefficients to local contrast (LC) using following equation:

$$LC_{i,j} = \frac{DCT_{i,j} \cdot DC}{2^{20}}$$

where DC is the DCT coefficient with indexes (0, 0).

- **Conversion** from LC to just-noticeable difference:

$$JND_{i,j} = LC_{i,j} \cdot CSF_{i,j}$$

where CSF is Contrast Sensitivity Function. Inverse MPEG-4 default quantization matrix is used as CSF in original article.

- **Weighted pooling of mean and maximum distortions**. First, absolute difference “D” is calculated for JND coefficients following by VQM value construction:

$$\mathrm{VQM} = \mathrm{mean}(|D|) + 0.005 \cdot \max(|D|)$$

Please, refer the original paper for the details.

Benchmark

Please note: values can vary depending on system configuration, input format and other factors.

Benchmark on VQMT 14.0 BETA r12792 PRO for Windows.

CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

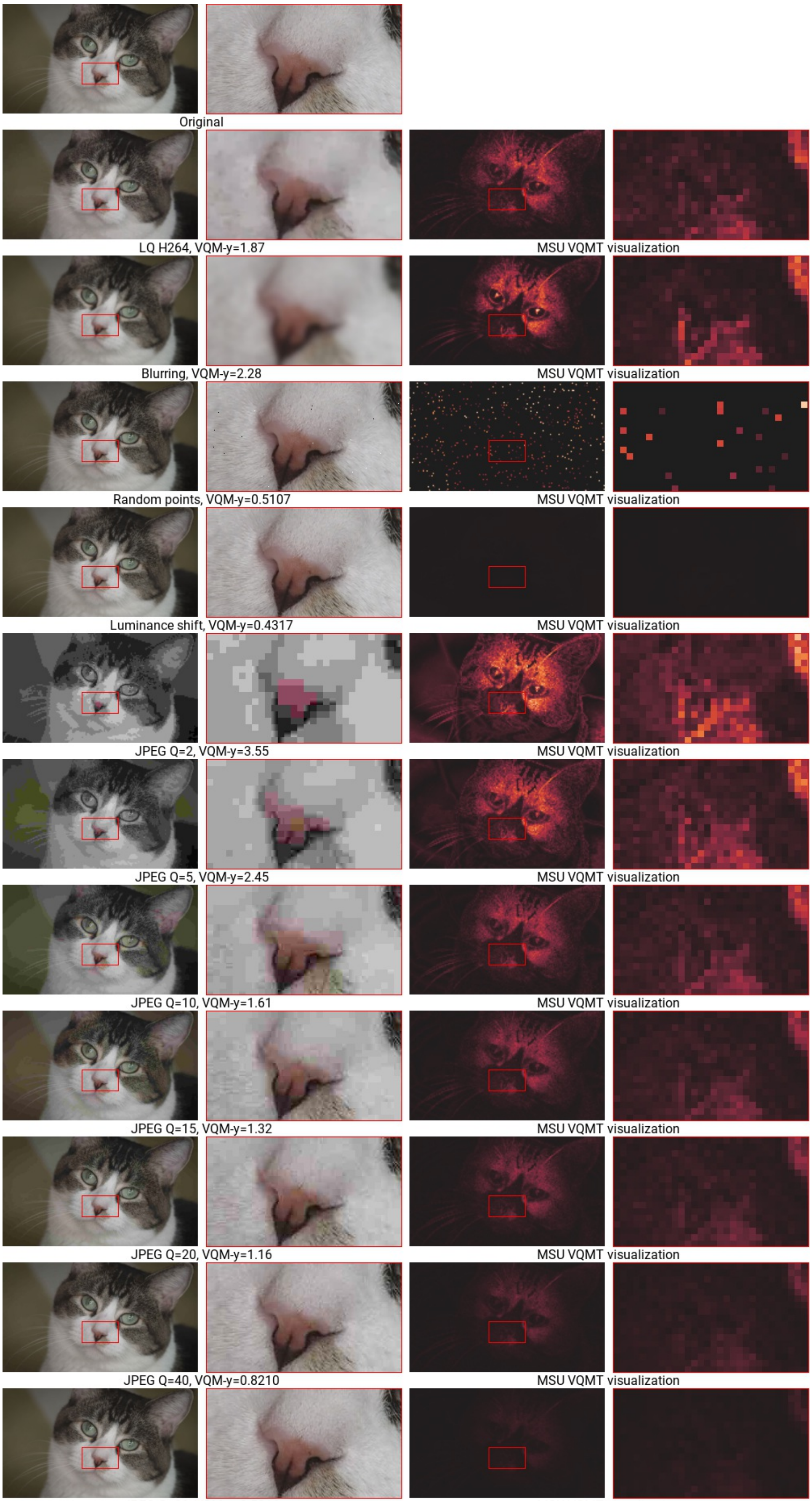
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	multithreaded	Y	HD 720p	112.46	0.01
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	52.07	0.02
VQMT 14.0 default	multithreaded	Y	4K 2160p	14.73	0.076
VQMT 14.0 default	singlethreaded	Y	HD 720p	25.64	0.041
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	12.57	0.084
VQMT 14.0 default	singlethreaded	Y	4K 2160p	3.24	0.329

Benchmark on VQMT 14.0 BETA r12792 PRO for Linux.
CPU: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64 cores
GPU: NVIDIA CUDA/TITAN RTX

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	multithreaded	Y	HD 720p	818.62	0.002
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	380.52	0.005
VQMT 14.0 default	multithreaded	Y	4K 2160p	49.1	0.035
VQMT 14.0 default	singlethreaded	Y	HD 720p	22.57	0.045
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	11.95	0.085
VQMT 14.0 default	singlethreaded	Y	4K 2160p	3.51	0.291

Benchmark on VQMT 13.1 r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 default	multithreaded	Y	HD 720p	112.93	0.01
VQMT 13.1 default	multithreaded	Y	FullHD 1080p	52.13	0.02
VQMT 13.1 default	multithreaded	Y	4K 2160p	14.53	0.076
VQMT 13.1 default	singlethreaded	Y	HD 720p	25.62	0.04
VQMT 13.1 default	singlethreaded	Y	FullHD 1080p	12.57	0.082
VQMT 13.1 default	singlethreaded	Y	4K 2160p	3.23	0.326



Original

LQ H264, VQM-y=1.87

MSU VQMT visualization

Blurring, VQM-y=2.28

MSU VQMT visualization

Random points, VQM-y=0.5107

MSU VQMT visualization

Luminance shift, VQM-y=0.4317

MSU VQMT visualization

JPEG Q=2, VQM-y=3.55

MSU VQMT visualization

JPEG Q=5, VQM-y=2.45

MSU VQMT visualization

JPEG Q=10, VQM-y=1.61

MSU VQMT visualization

JPEG Q=15, VQM-y=1.32

MSU VQMT visualization

JPEG Q=20, VQM-y=1.16

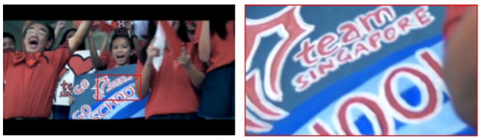
MSU VQMT visualization

JPEG Q=40, VQM-y=0.8210

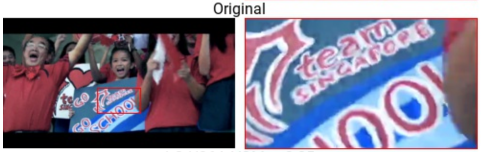
MSU VQMT visualization

JPEG Q=80, VQM-y=0.5216

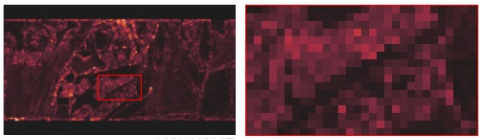
MSU VQMT visualization



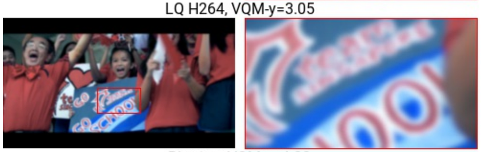
Original



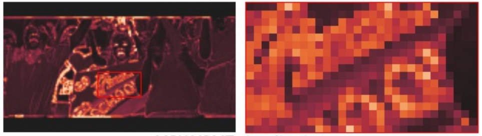
LQ H264, VQM-y=3.05



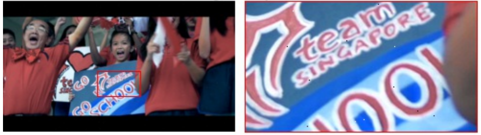
MSU VQMT visualization



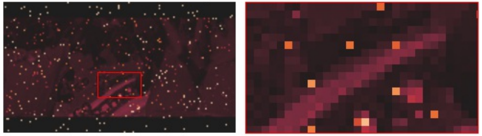
Blurring, VQM-y=4.22



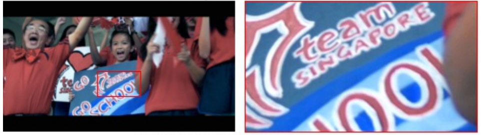
MSU VQMT visualization



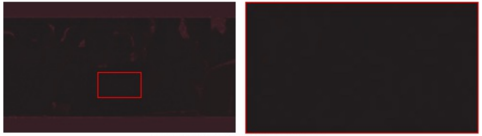
Random points, VQM-y=2.38



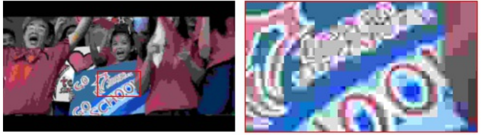
MSU VQMT visualization



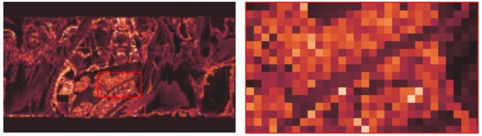
Luminance shift, VQM-y=1.02



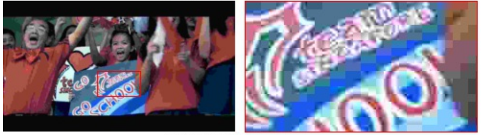
MSU VQMT visualization



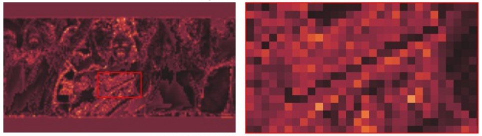
JPEG Q=2, VQM-y=4.35



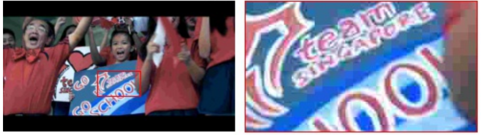
MSU VQMT visualization



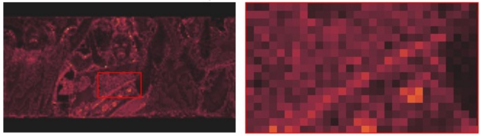
JPEG Q=5, VQM-y=4.18



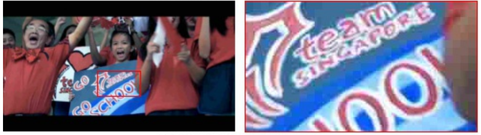
MSU VQMT visualization



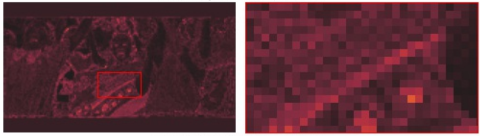
JPEG Q=10, VQM-y=2.95



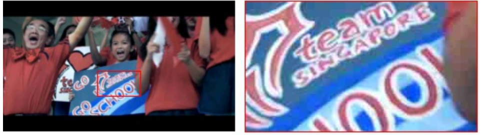
MSU VQMT visualization



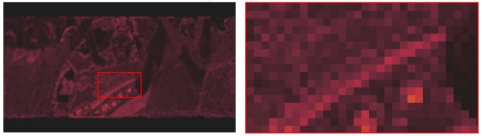
JPEG Q=15, VQM-y=2.96



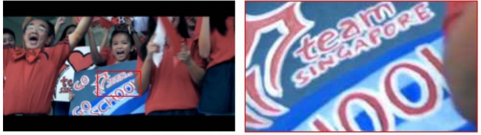
MSU VQMT visualization



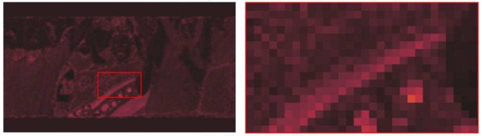
JPEG Q=20, VQM-y=2.62



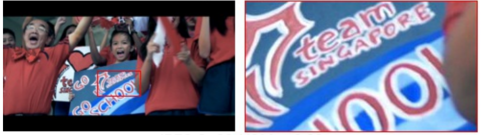
MSU VQMT visualization



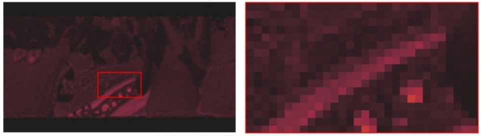
JPEG Q=40, VQM-y=2.59



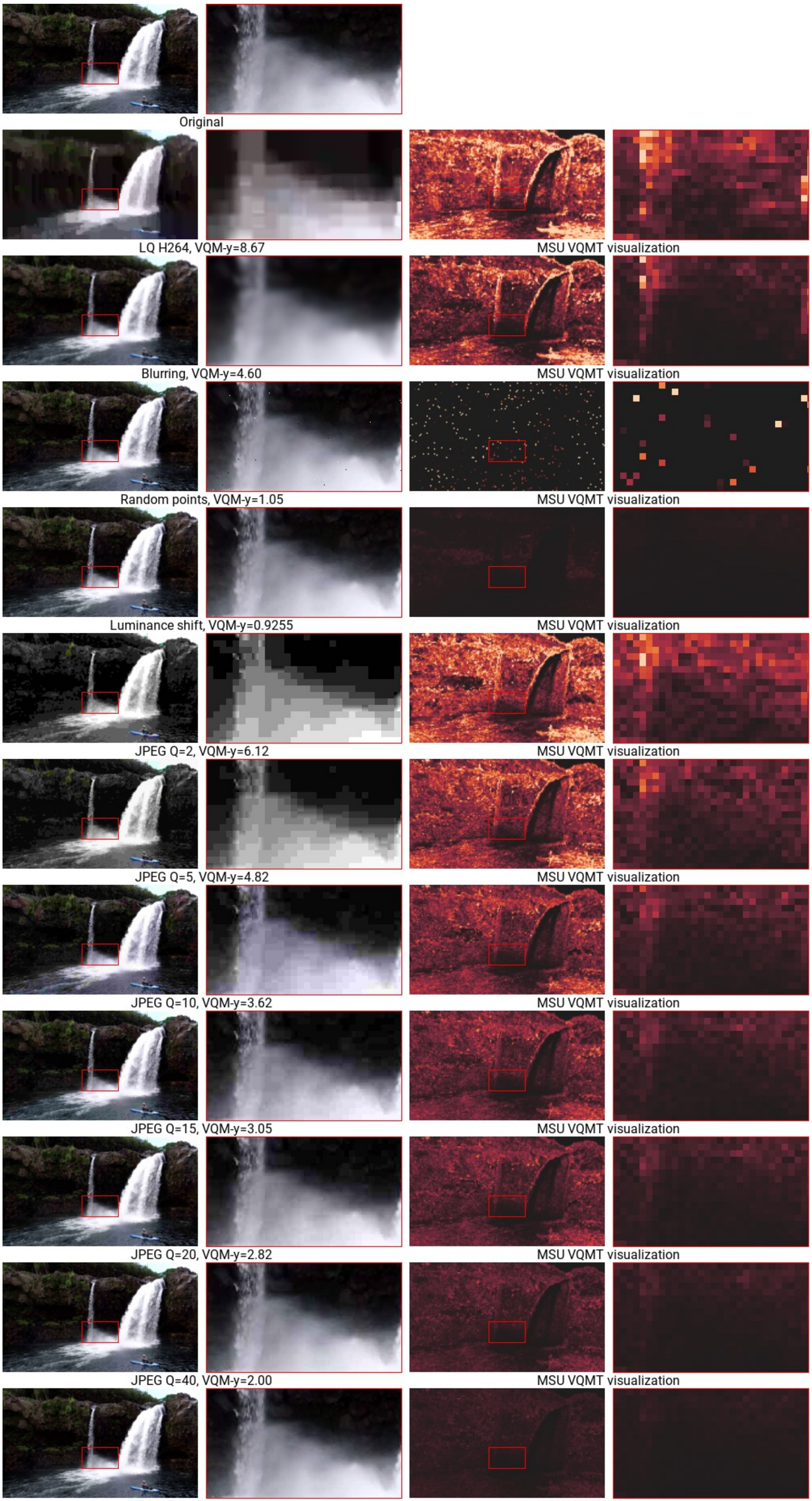
MSU VQMT visualization



JPEG Q=80, VQM-y=2.28



MSU VQMT visualization



Original

LQ H264, VQM-y=8.67

MSU VQMT visualization

Blurring, VQM-y=4.60

MSU VQMT visualization

Random points, VQM-y=1.05

MSU VQMT visualization

Luminance shift, VQM-y=0.9255

MSU VQMT visualization

JPEG Q=2, VQM-y=6.12

MSU VQMT visualization

JPEG Q=5, VQM-y=4.82

MSU VQMT visualization

JPEG Q=10, VQM-y=3.62

MSU VQMT visualization

JPEG Q=15, VQM-y=3.05

MSU VQMT visualization

JPEG Q=20, VQM-y=2.82

MSU VQMT visualization

JPEG Q=40, VQM-y=2.00

MSU VQMT visualization

JPEG Q=80, VQM-y=1.29

MSU VQMT visualization

MSU Time shift

General info

Metric type	full-reference temporal metric
Value range	depending on settings
Value interpretation	Metric value is shift in frames of distorted sequence relative to original. If n-th value is x, then n+x'th frame of distorted corresponds to n'th frame of original
MSU VQMT implementations	CPU multithreaded
MSU VQMT visualization	block-wise
Available colorspaces	Y
Output values	metric value multiple individual measument results if on
Aggregated values	standard set
MSU VQMT usages	-metr time-shift

Algorithm description

This metric performes calculation of another metric (base metric: PSNR or SSIM) between each frame of original image and several frames of distorted image. You can choose what interval will be used in settings. This metric can detect such artifacts as skipped frame, duplicated frame, small fps mismatch.

Metric considers, that the best metric value for specific original frame is the correct shift. Metric can prefer smaller shift with base metric value X to bigger shift with metric value Y if $X > Y \cdot \mathrm{threshold}$, where threshold can be set in metric settings. This helps to avoid random fluctuations.

Also, values of base metric will be smothed over adjacent frames if smoothing is on. This can help in case of very there are very similar adjacent frames in seuqence or the desired frame is absent (for example, negative shift on first frame).

Benchmark

Please note: values can vary depending on system configuration, input format and other factors.

Benchmark on VQMT 14.0 BETA r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	multithreaded	Y	HD 720p	357.01	0.004
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	137.46	0.008
VQMT 14.0 default	multithreaded	Y	4K 2160p	37.71	0.031
VQMT 14.0 default	singlethreaded	Y	HD 720p	131.66	0.009
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	54.6	0.019

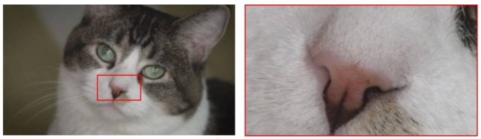
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	singlethreaded	Y	4K 2160p	17.66	0.06

Benchmark on VQMT 14.0 BETA r12792 PRO for Linux.
CPU: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64 cores
GPU: NVIDIA CUDA/TITAN RTX

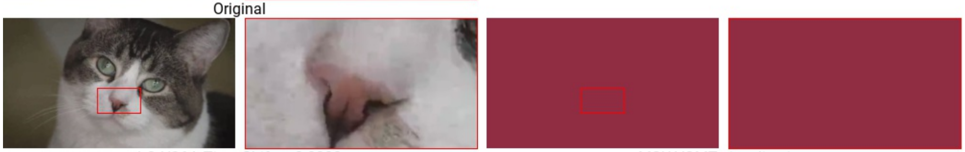
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 default	multithreaded	Y	HD 720p	1150.84	0.002
VQMT 14.0 default	multithreaded	Y	FullHD 1080p	573.55	0.004
VQMT 14.0 default	multithreaded	Y	4K 2160p	69.45	0.023
VQMT 14.0 default	singlethreaded	Y	HD 720p	85.39	0.012
VQMT 14.0 default	singlethreaded	Y	FullHD 1080p	38.37	0.026
VQMT 14.0 default	singlethreaded	Y	4K 2160p	15.18	0.068

Benchmark on VQMT 13.1 r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 default	multithreaded	Y	HD 720p	361.8	0.004
VQMT 13.1 default	multithreaded	Y	FullHD 1080p	130.18	0.009
VQMT 13.1 default	multithreaded	Y	4K 2160p	32.62	0.034
VQMT 13.1 default	singlethreaded	Y	HD 720p	134.86	0.008
VQMT 13.1 default	singlethreaded	Y	FullHD 1080p	55.41	0.019
VQMT 13.1 default	singlethreaded	Y	4K 2160p	17.17	0.061

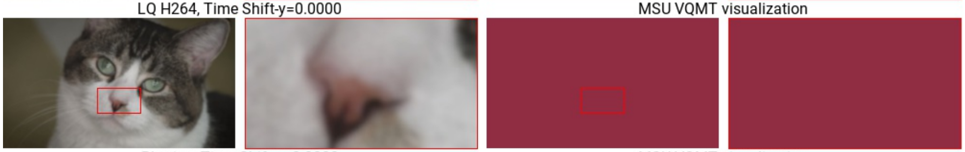


Original



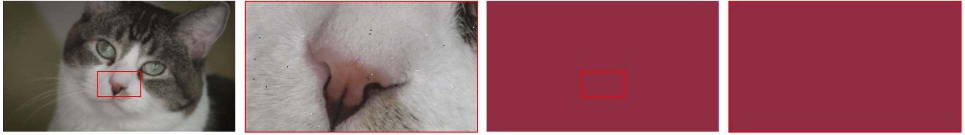
LQ H264, Time Shift-y=0.0000

MSU VQMT visualization



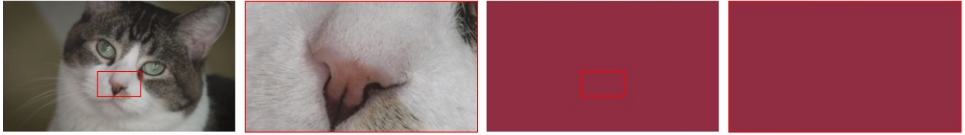
Blurring, Time Shift-y=0.0000

MSU VQMT visualization



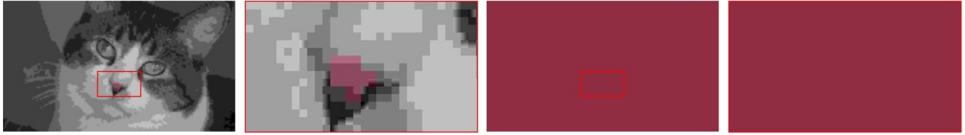
Random points, Time Shift-y=0.0000

MSU VQMT visualization



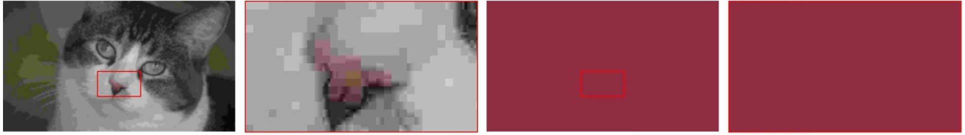
Luminance shift, Time Shift-y=0.0000

MSU VQMT visualization



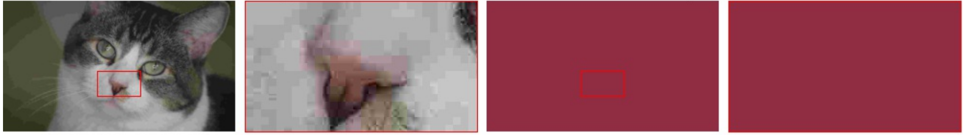
JPEG Q=2, Time Shift-y=0.0000

MSU VQMT visualization



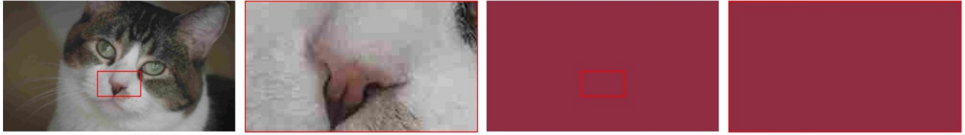
JPEG Q=5, Time Shift-y=0.0000

MSU VQMT visualization



JPEG Q=10, Time Shift-y=0.0000

MSU VQMT visualization

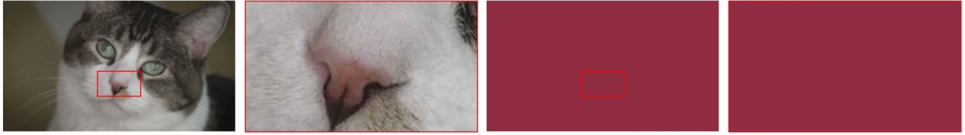


JPEG Q=15, Time Shift-y=0.0000

MSU VQMT visualization

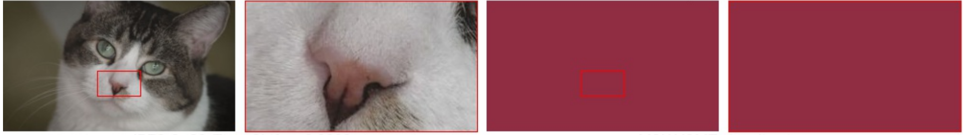
JPEG Q=20, Time Shift-y=0.0000

MSU VQMT visualization



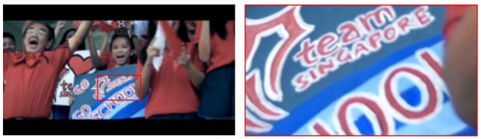
JPEG Q=40, Time Shift-y=0.0000

MSU VQMT visualization



JPEG Q=80, Time Shift-y=0.0000

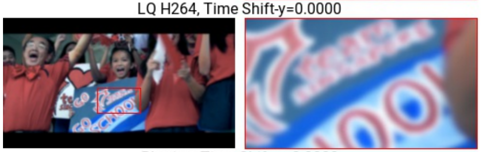
MSU VQMT visualization



Original

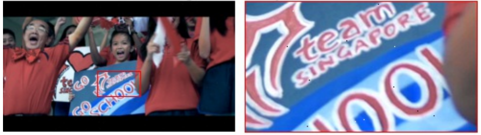
LQ H264, Time Shift-y=0.0000

MSU VQMT visualization



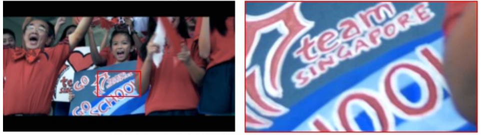
Blurring, Time Shift-y=0.0000

MSU VQMT visualization



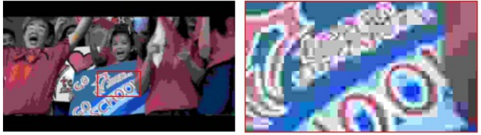
Random points, Time Shift-y=0.0000

MSU VQMT visualization



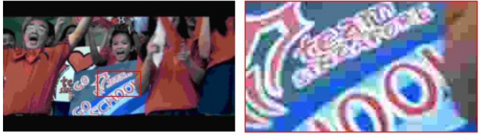
Luminance shift, Time Shift-y=0.0000

MSU VQMT visualization



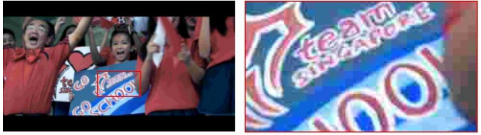
JPEG Q=2, Time Shift-y=0.0000

MSU VQMT visualization



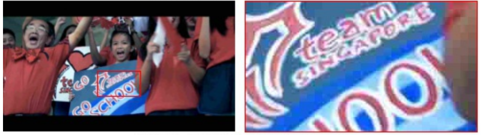
JPEG Q=5, Time Shift-y=0.0000

MSU VQMT visualization



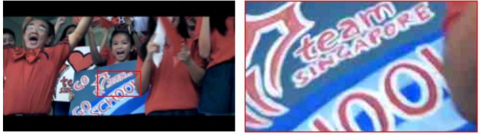
JPEG Q=10, Time Shift-y=0.0000

MSU VQMT visualization



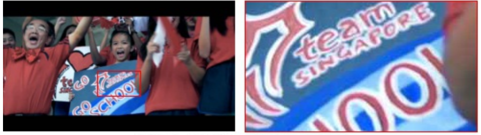
JPEG Q=15, Time Shift-y=0.0000

MSU VQMT visualization



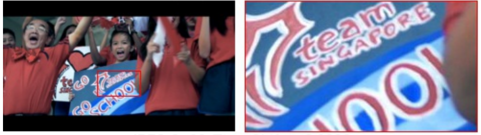
JPEG Q=20, Time Shift-y=0.0000

MSU VQMT visualization



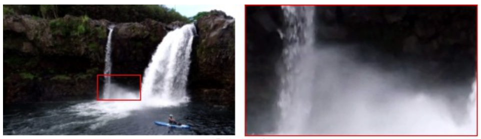
JPEG Q=40, Time Shift-y=0.0000

MSU VQMT visualization

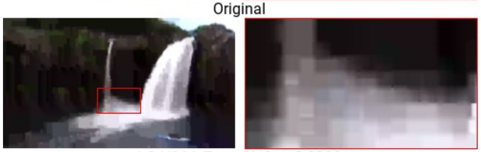


JPEG Q=80, Time Shift-y=0.0000

MSU VQMT visualization



Original



LQ H264, Time Shift-y=0.0000

MSU VQMT visualization

Blurring, Time Shift-y=0.0000

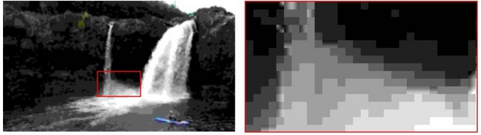
MSU VQMT visualization

Random points, Time Shift-y=0.0000

MSU VQMT visualization

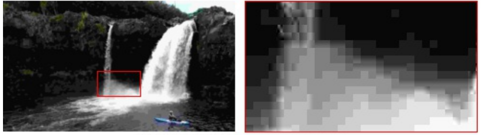
Luminance shift, Time Shift-y=0.0000

MSU VQMT visualization



JPEG Q=2, Time Shift-y=0.0000

MSU VQMT visualization

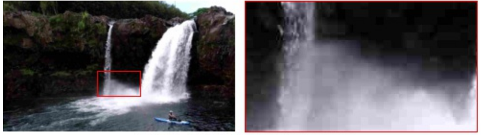


JPEG Q=5, Time Shift-y=0.0000

MSU VQMT visualization

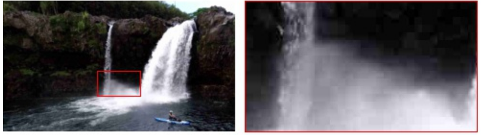
JPEG Q=10, Time Shift-y=0.0000

MSU VQMT visualization



JPEG Q=15, Time Shift-y=0.0000

MSU VQMT visualization

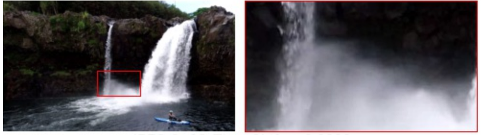


JPEG Q=20, Time Shift-y=0.0000

MSU VQMT visualization

JPEG Q=40, Time Shift-y=0.0000

MSU VQMT visualization



JPEG Q=80, Time Shift-y=0.0000

MSU VQMT visualization

No-reference informational metrics

MSU Blurring

General info

Metric type	no-reference image metric
Value range	(constant image) 0..1 (very noisy)
Value interpretation	bigger is more noise
MSU VQMT implementations	CPU multithreaded sigma (default), CPU multithreaded delta
MSU VQMT visualization	pixel-wise
Available colorspaces	Y, R, G, B for sigma Y, U, V, R, G, B for delta
Output values	metric value
Aggregated values	standard set
MSU VQMT usages	-metr blurring [over <color components>] -metr blurring_delta [over <color components>]

Algorithm description

This metric allows you to compare power of blurring of two images. If value of the metric for first picture is greater than for second, it means that second picture is more blurred, than first.

Main features: this metric is fast and doesn’t require source video.

This method estimates color variance in the neighborhood of a pixel and computes average variance. This metric has 2 variations:

- **Sigma** (default since VQMT 11). It uses 3-pixel radius neighborhood and normalized Gaussian kernel
- **Delta** (the only before VQMT 11). It uses 1-pixel radius neighborhood

Notes:

- You can’t measure blurriness on constant or gradient areas of input image. So, the value of metric is very dependent on amount of edges in images.
- This metric will detect not only compression artifacts, but natural not-infocus areas, so the value of metric is very dependent on area of focused objects in the frame. You shouldn’t use value of this metric as final blurrness index, use is only to compare blurrness of images with similar structure.

Benchmark

Please note: values can vary depending on system configuration, input format and other factors.

**Benchmark on VQMT 14.0 BETA r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti**

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 Delta	multithreaded	Y	HD 720p	983.56	0.002

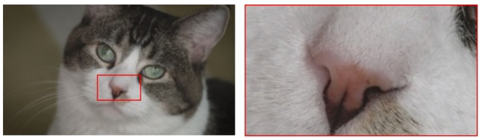
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 Delta	multithreaded	Y	FullHD 1080p	454.56	0.003
VQMT 14.0 Delta	multithreaded	Y	4K 2160p	104.73	0.011
VQMT 14.0 Sigma	multithreaded	Y	HD 720p	105.75	0.01
VQMT 14.0 Sigma	multithreaded	Y	FullHD 1080p	47.22	0.023
VQMT 14.0 Sigma	multithreaded	Y	4K 2160p	11.3	0.1
VQMT 14.0 Delta	singlethreaded	Y	HD 720p	349.35	0.004
VQMT 14.0 Delta	singlethreaded	Y	FullHD 1080p	146.94	0.008
VQMT 14.0 Delta	singlethreaded	Y	4K 2160p	33.55	0.031
VQMT 14.0 Sigma	singlethreaded	Y	HD 720p	37.04	0.028
VQMT 14.0 Sigma	singlethreaded	Y	FullHD 1080p	16.81	0.062
VQMT 14.0 Sigma	singlethreaded	Y	4K 2160p	4.16	0.255

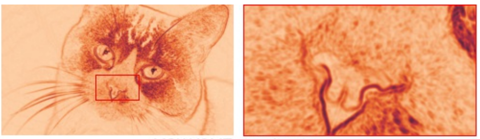
**Benchmark on VQMT 14.0 BETA r12792 PRO for Linux.
CPU: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64 cores
GPU: NVIDIA CUDA/TITAN RTX**

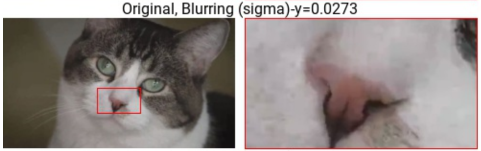
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 Delta	multithreaded	Y	HD 720p	1455.72	0.001
VQMT 14.0 Delta	multithreaded	Y	FullHD 1080p	716.32	0.003
VQMT 14.0 Delta	multithreaded	Y	4K 2160p	137.73	0.011
VQMT 14.0 Sigma	multithreaded	Y	HD 720p	390.48	0.003
VQMT 14.0 Sigma	multithreaded	Y	FullHD 1080p	161.57	0.008
VQMT 14.0 Sigma	multithreaded	Y	4K 2160p	25.87	0.054
VQMT 14.0 Delta	singlethreaded	Y	HD 720p	288.69	0.004
VQMT 14.0 Delta	singlethreaded	Y	FullHD 1080p	127.65	0.008
VQMT 14.0 Delta	singlethreaded	Y	4K 2160p	28.37	0.036
VQMT 14.0 Sigma	singlethreaded	Y	HD 720p	43.71	0.023
VQMT 14.0 Sigma	singlethreaded	Y	FullHD 1080p	23.15	0.044
VQMT 14.0 Sigma	singlethreaded	Y	4K 2160p	6.82	0.149

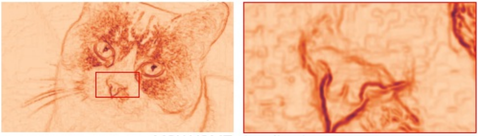
**Benchmark on VQMT 13.1 r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti**

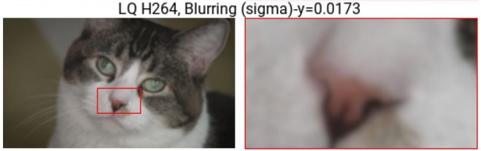
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 Delta	multithreaded	Y	HD 720p	1003.64	0.002
VQMT 13.1 Delta	multithreaded	Y	FullHD 1080p	438.04	0.003
VQMT 13.1 Delta	multithreaded	Y	4K 2160p	97.5	0.012
VQMT 13.1 Sigma	multithreaded	Y	HD 720p	105.59	0.01
VQMT 13.1 Sigma	multithreaded	Y	FullHD 1080p	46.81	0.023
VQMT 13.1 Sigma	multithreaded	Y	4K 2160p	11.25	0.099
VQMT 13.1 Delta	singlethreaded	Y	HD 720p	352.32	0.003
VQMT 13.1 Delta	singlethreaded	Y	FullHD 1080p	147.56	0.007
VQMT 13.1 Delta	singlethreaded	Y	4K 2160p	32.66	0.031
VQMT 13.1 Sigma	singlethreaded	Y	HD 720p	36.93	0.028
VQMT 13.1 Sigma	singlethreaded	Y	FullHD 1080p	16.75	0.062
VQMT 13.1 Sigma	singlethreaded	Y	4K 2160p	4.15	0.252

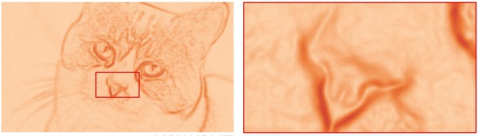

Original, Blurring (sigma)-y=0.0273

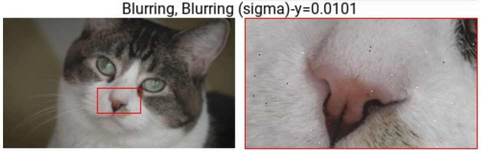

MSU VQMT visualization

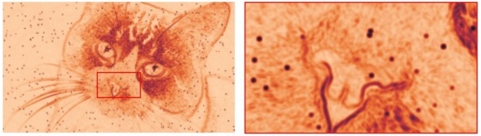

LQ H264, Blurring (sigma)-y=0.0173

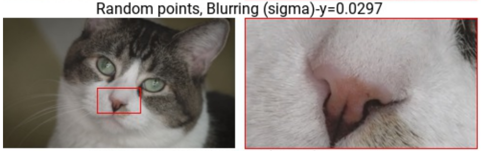

MSU VQMT visualization

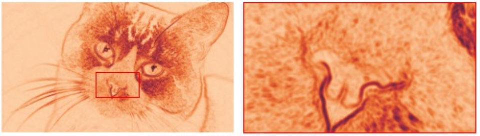

Blurring, Blurring (sigma)-y=0.0101

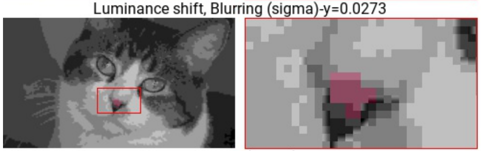

MSU VQMT visualization

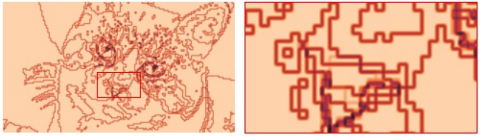

Random points, Blurring (sigma)-y=0.0297

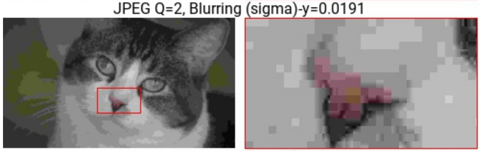

MSU VQMT visualization

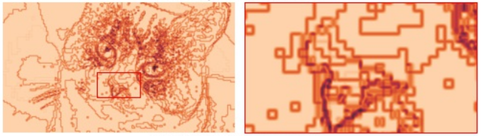

Luminance shift, Blurring (sigma)-y=0.0273

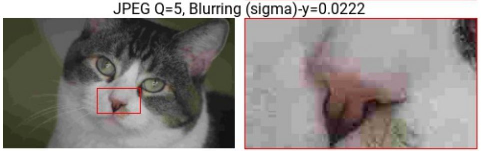

MSU VQMT visualization

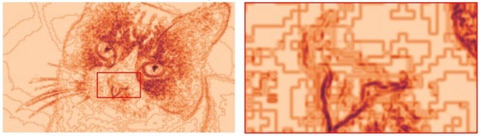

JPEG Q=2, Blurring (sigma)-y=0.0191

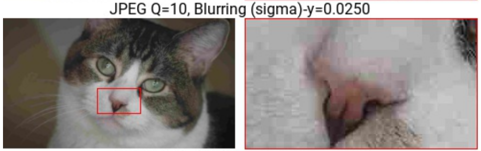

MSU VQMT visualization

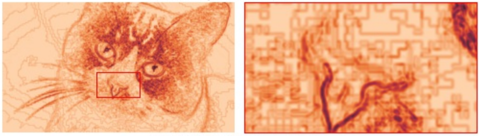

JPEG Q=5, Blurring (sigma)-y=0.0222

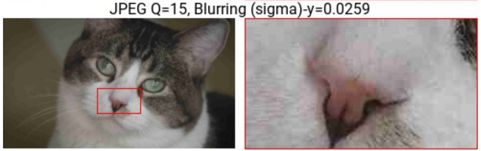

MSU VQMT visualization

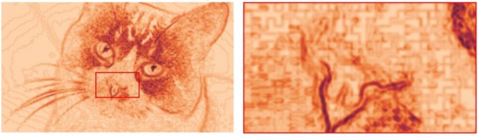

JPEG Q=10, Blurring (sigma)-y=0.0250

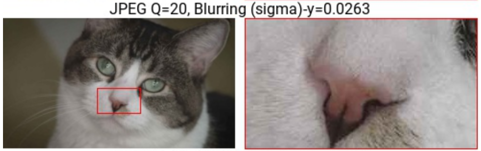

MSU VQMT visualization

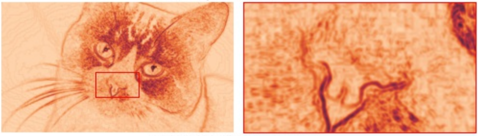

JPEG Q=15, Blurring (sigma)-y=0.0259

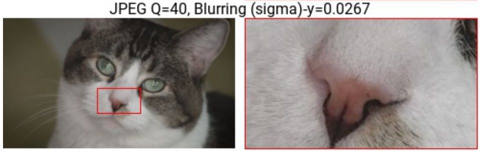

MSU VQMT visualization

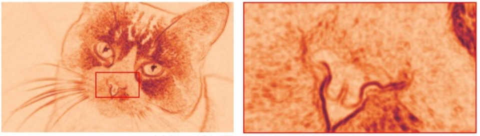

JPEG Q=20, Blurring (sigma)-y=0.0263

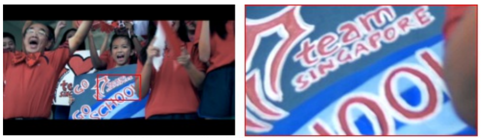

MSU VQMT visualization

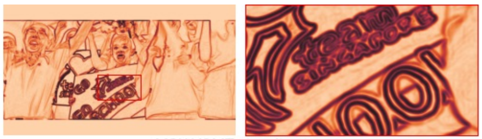

JPEG Q=40, Blurring (sigma)-y=0.0267

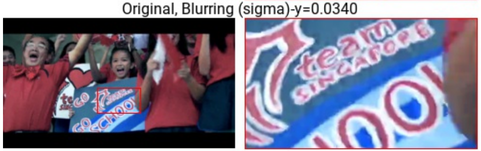

MSU VQMT visualization

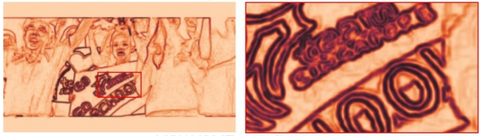

JPEG Q=80, Blurring (sigma)-y=0.0270

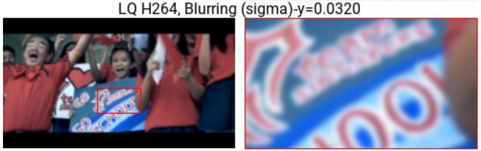

MSU VQMT visualization

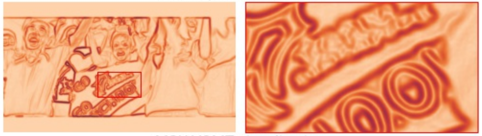

Original, Blurring (sigma)-y=0.0340

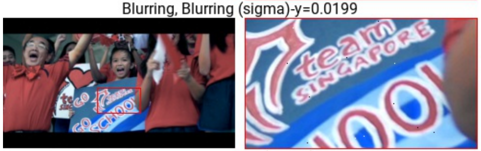

MSU VQMT visualization

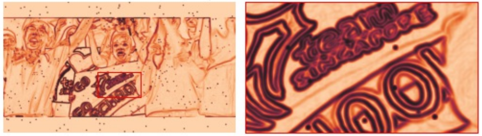

LQ H264, Blurring (sigma)-y=0.0320


MSU VQMT visualization

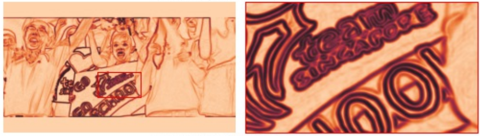

Blurring, Blurring (sigma)-y=0.0199

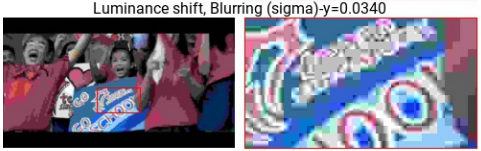

MSU VQMT visualization

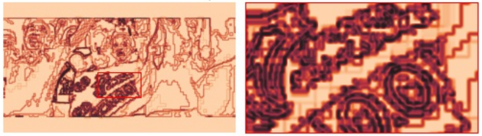

Random points, Blurring (sigma)-y=0.0371

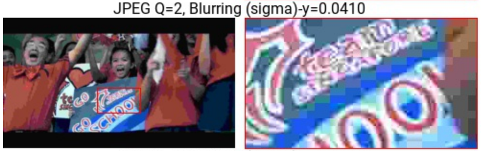

MSU VQMT visualization

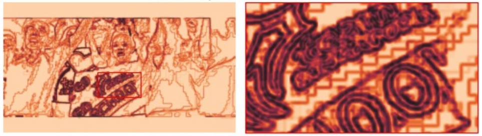
Luminance shift, Blurring (sigma)-y=0.0340

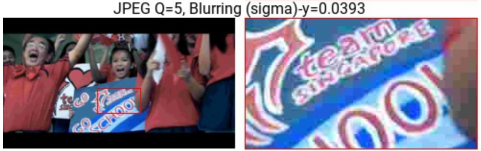

MSU VQMT visualization

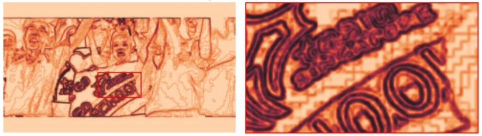

JPEG Q=2, Blurring (sigma)-y=0.0410


MSU VQMT visualization

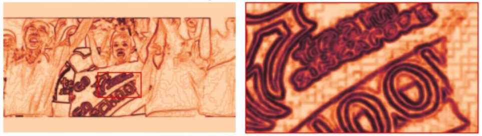

JPEG Q=5, Blurring (sigma)-y=0.0393


MSU VQMT visualization

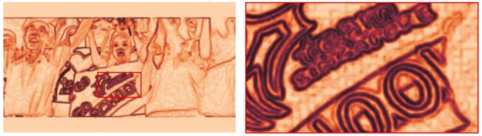

JPEG Q=10, Blurring (sigma)-y=0.0377


MSU VQMT visualization

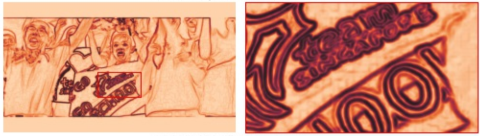
JPEG Q=15, Blurring (sigma)-y=0.0367

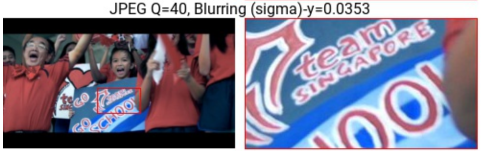

MSU VQMT visualization

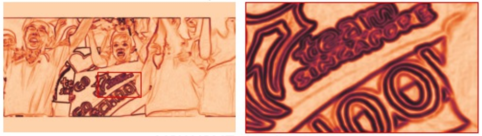
JPEG Q=20, Blurring (sigma)-y=0.0362

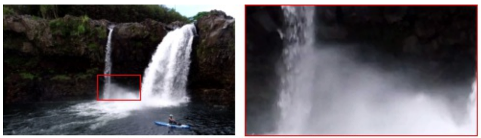

MSU VQMT visualization

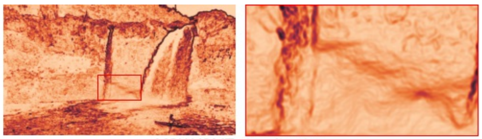
JPEG Q=40, Blurring (sigma)-y=0.0353

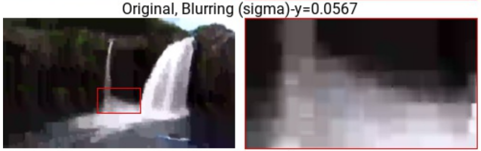

MSU VQMT visualization

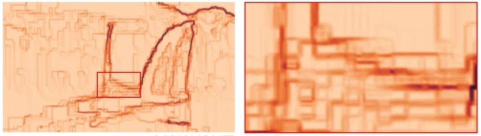

JPEG Q=80, Blurring (sigma)-y=0.0349

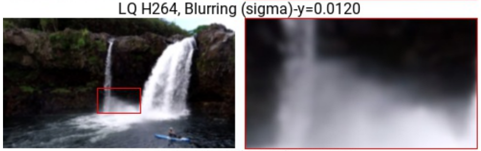

MSU VQMT visualization

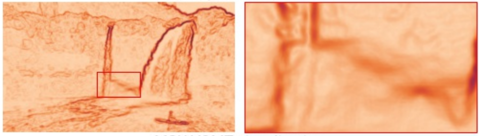

Original, Blurring (sigma)-y=0.0567

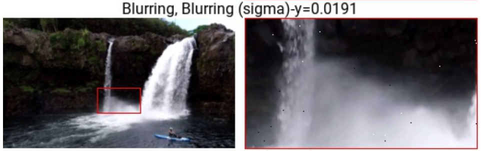

MSU VQMT visualization

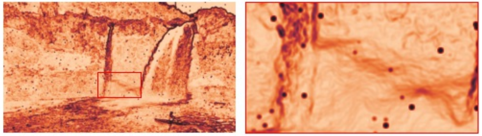

LQ H264, Blurring (sigma)-y=0.0120

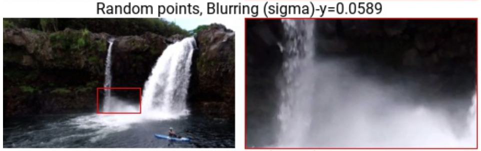

MSU VQMT visualization

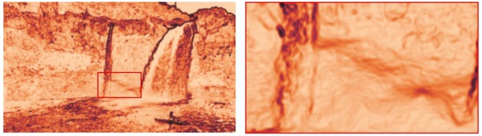

Blurring, Blurring (sigma)-y=0.0191

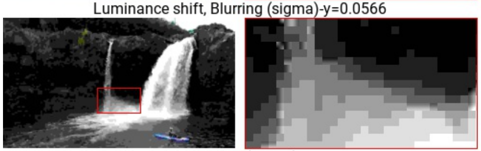

MSU VQMT visualization

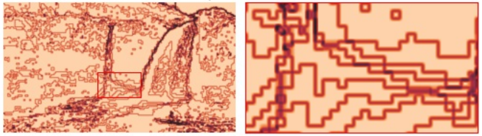

Random points, Blurring (sigma)-y=0.0589

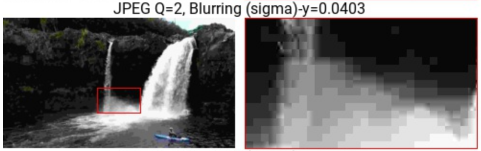

MSU VQMT visualization

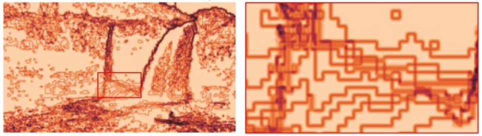

Luminance shift, Blurring (sigma)-y=0.0566

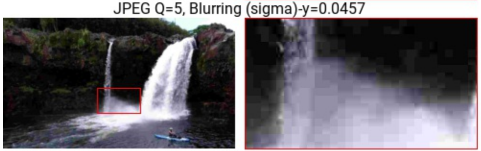

MSU VQMT visualization

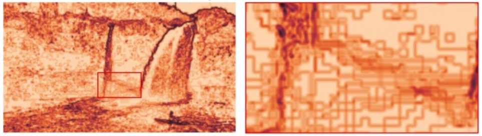

JPEG Q=2, Blurring (sigma)-y=0.0403

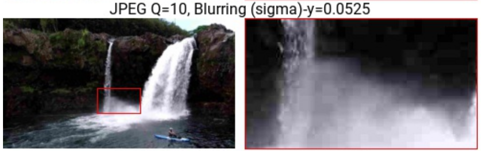

MSU VQMT visualization

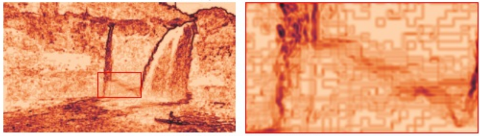

JPEG Q=5, Blurring (sigma)-y=0.0457

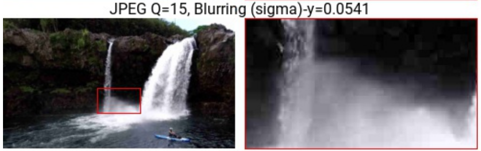

MSU VQMT visualization

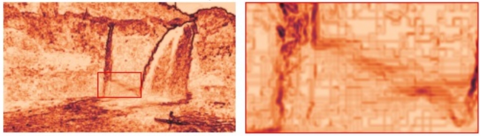

JPEG Q=10, Blurring (sigma)-y=0.0525

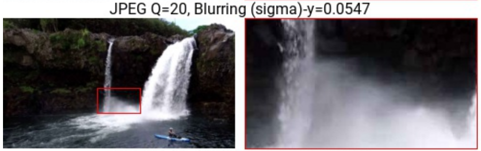

MSU VQMT visualization

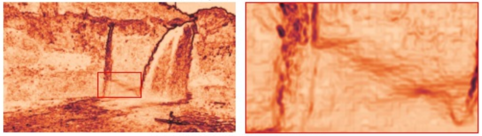

JPEG Q=15, Blurring (sigma)-y=0.0541

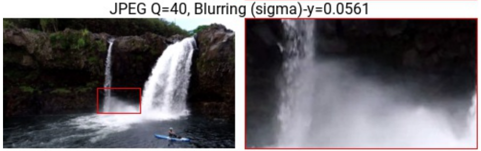

MSU VQMT visualization

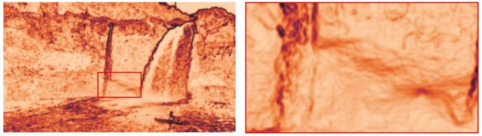

JPEG Q=20, Blurring (sigma)-y=0.0547

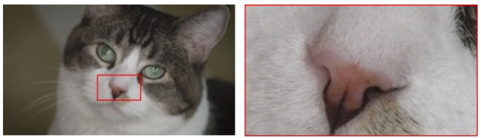

MSU VQMT visualization

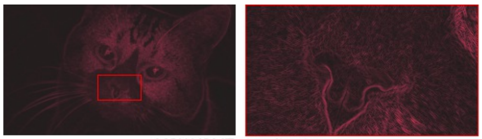

JPEG Q=40, Blurring (sigma)-y=0.0561

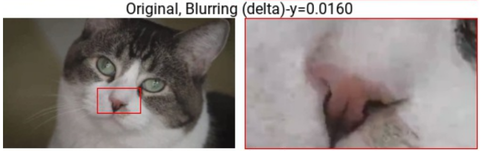

MSU VQMT visualization

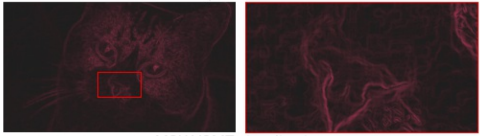

JPEG Q=80, Blurring (sigma)-y=0.0569

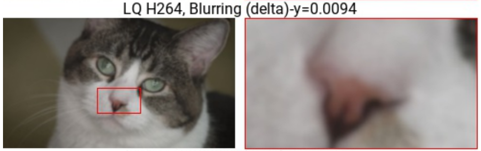

MSU VQMT visualization

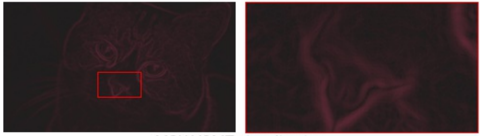

Original, Blurring (delta)-y=0.0160

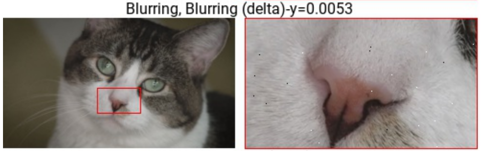

MSU VQMT visualization

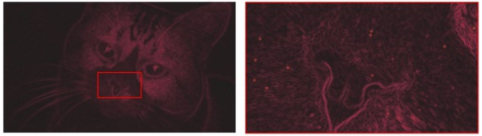

LQ H264, Blurring (delta)-y=0.0094

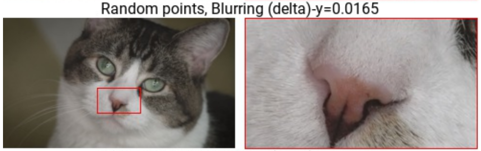

MSU VQMT visualization

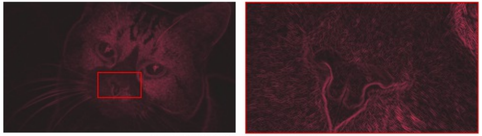

Blurring, Blurring (delta)-y=0.0053

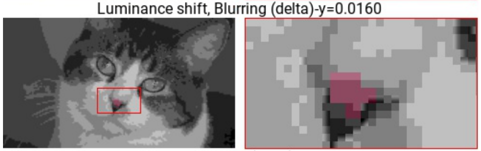

MSU VQMT visualization

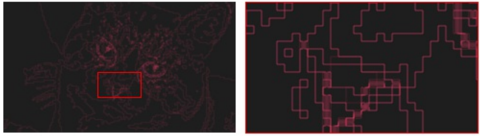

Random points, Blurring (delta)-y=0.0165

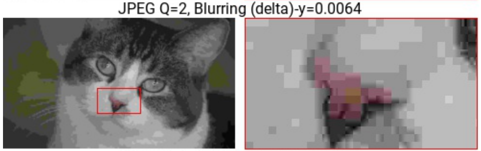

MSU VQMT visualization

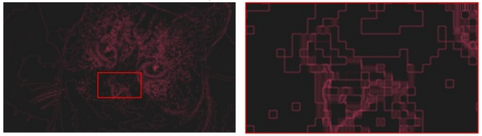

Luminance shift, Blurring (delta)-y=0.0160

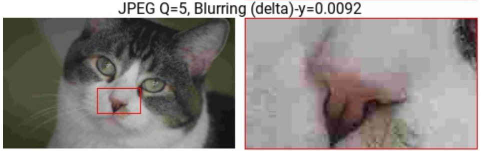

MSU VQMT visualization

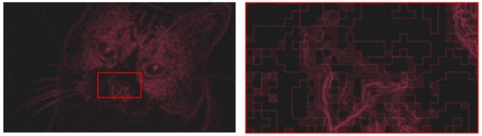

JPEG Q=2, Blurring (delta)-y=0.0064

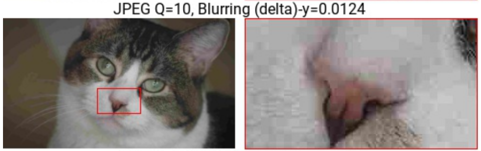

MSU VQMT visualization

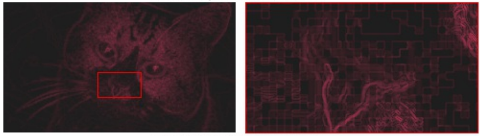

JPEG Q=5, Blurring (delta)-y=0.0092

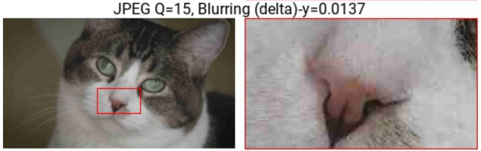

MSU VQMT visualization

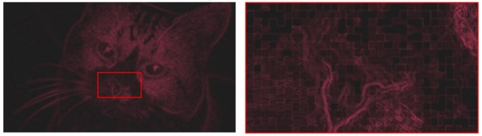

JPEG Q=10, Blurring (delta)-y=0.0124

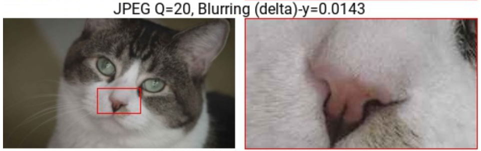

MSU VQMT visualization

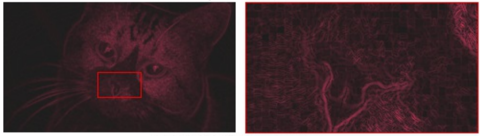

JPEG Q=15, Blurring (delta)-y=0.0137

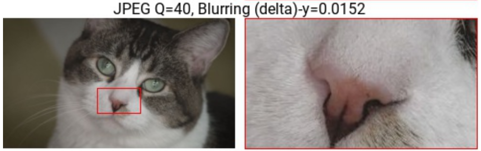

MSU VQMT visualization

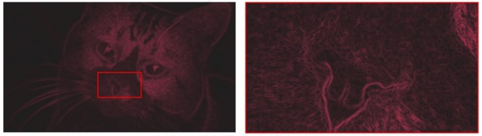

JPEG Q=20, Blurring (delta)-y=0.0143

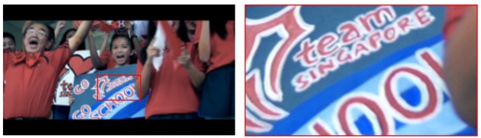

MSU VQMT visualization

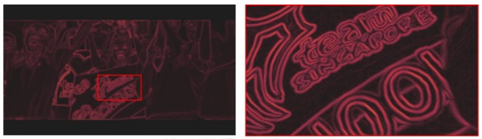

JPEG Q=40, Blurring (delta)-y=0.0152

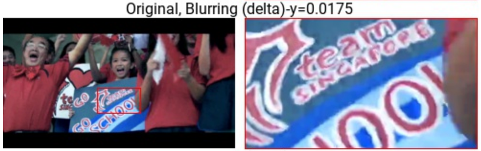

MSU VQMT visualization

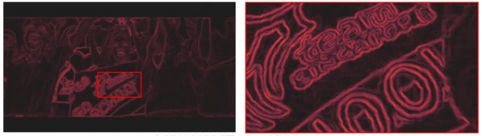

JPEG Q=80, Blurring (delta)-y=0.0157

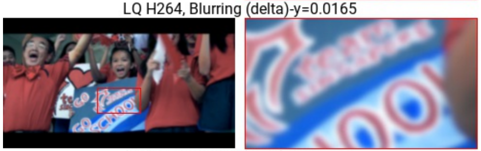

MSU VQMT visualization

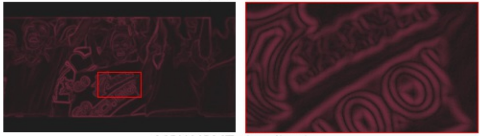

Original, Blurring (delta)-y=0.0175

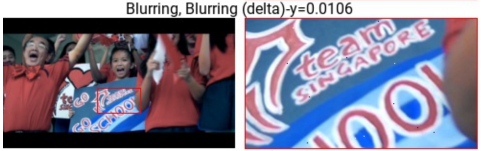

MSU VQMT visualization

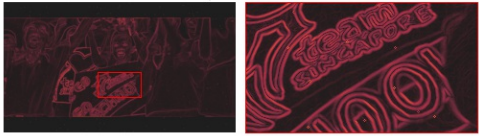

LQ H264, Blurring (delta)-y=0.0165


MSU VQMT visualization

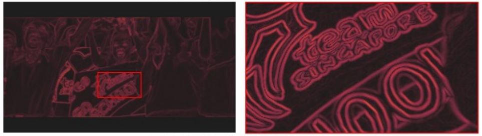

Blurring, Blurring (delta)-y=0.0106


MSU VQMT visualization

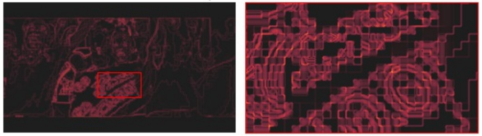

Random points, Blurring (delta)-y=0.0185

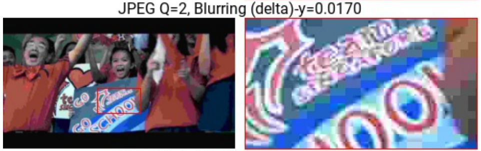

MSU VQMT visualization

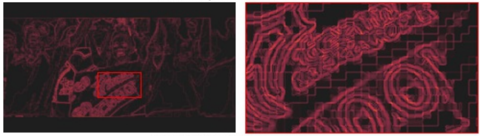
Luminance shift, Blurring (delta)-y=0.0175

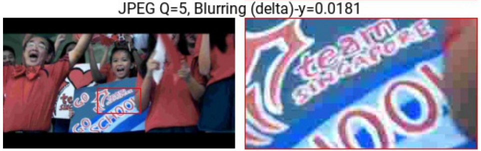

MSU VQMT visualization

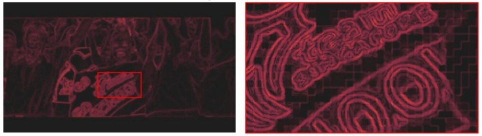
JPEG Q=2, Blurring (delta)-y=0.0170

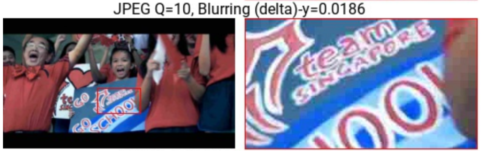

MSU VQMT visualization

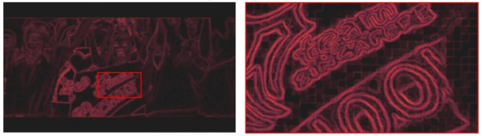

JPEG Q=5, Blurring (delta)-y=0.0181


MSU VQMT visualization

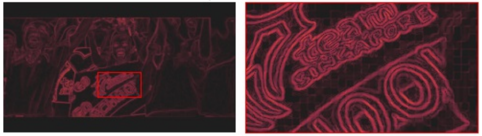

JPEG Q=10, Blurring (delta)-y=0.0186


MSU VQMT visualization

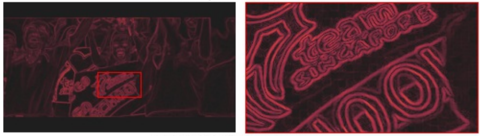

JPEG Q=15, Blurring (delta)-y=0.0185

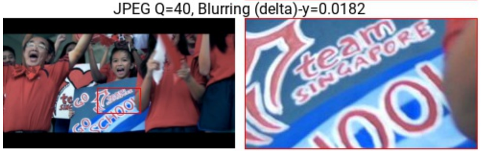

MSU VQMT visualization

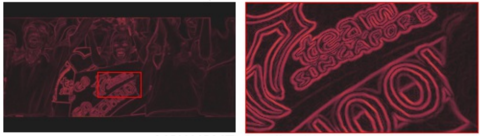
JPEG Q=20, Blurring (delta)-y=0.0184

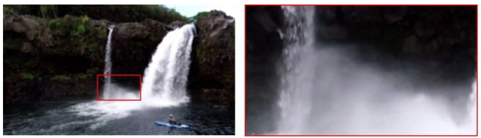

MSU VQMT visualization

JPEG Q=40, Blurring (delta)-y=0.0182

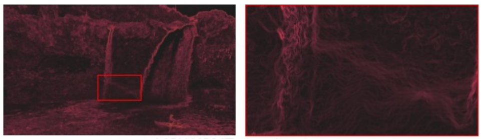

MSU VQMT visualization


JPEG Q=80, Blurring (delta)-y=0.0180


MSU VQMT visualization

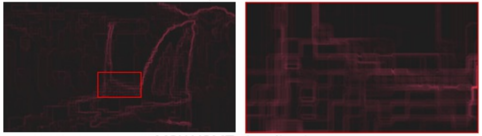


Original, Blurring (delta)-y=0.0313

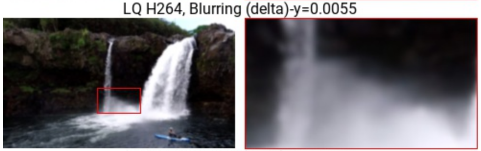


MSU VQMT visualization

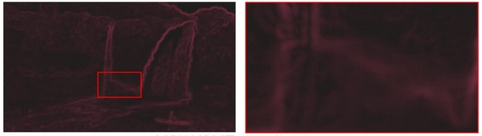
LQ H264, Blurring (delta)-y=0.0055



MSU VQMT visualization

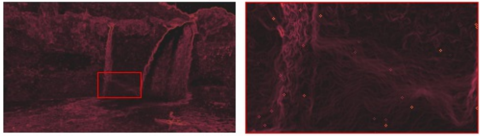


Blurring, Blurring (delta)-y=0.0102

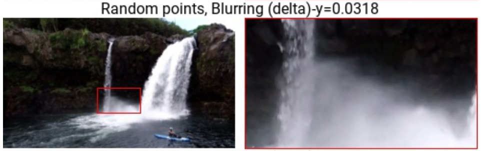


MSU VQMT visualization

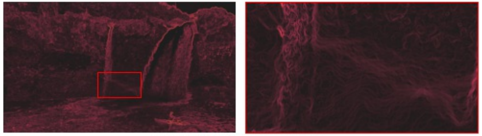
Random points, Blurring (delta)-y=0.0318



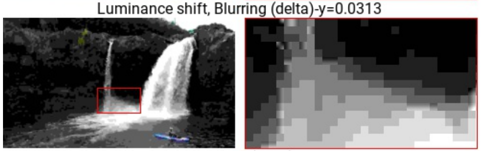
MSU VQMT visualization



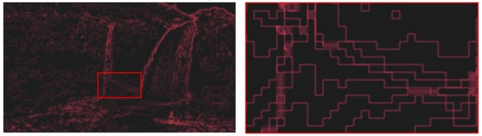
Luminance shift, Blurring (delta)-y=0.0313



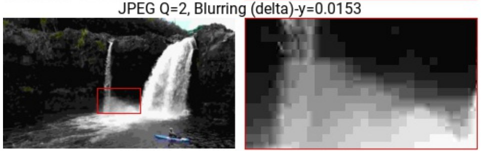
MSU VQMT visualization



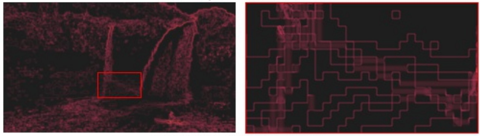
JPEG Q=2, Blurring (delta)-y=0.0153



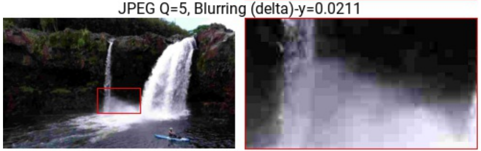
MSU VQMT visualization



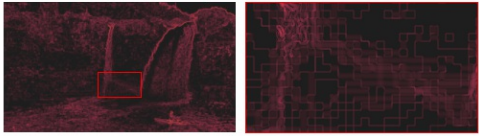
JPEG Q=5, Blurring (delta)-y=0.0211



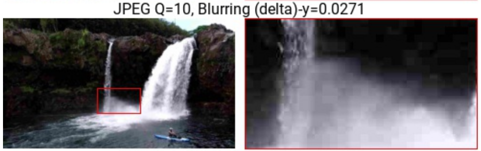
MSU VQMT visualization



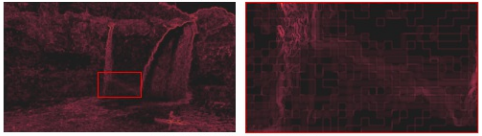
JPEG Q=10, Blurring (delta)-y=0.0271



MSU VQMT visualization

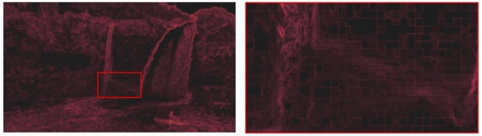


JPEG Q=15, Blurring (delta)-y=0.0291



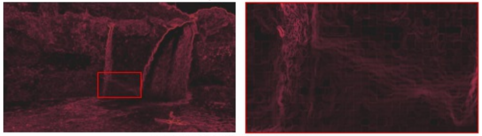
MSU VQMT visualization

JPEG Q=20, Blurring (delta)-y=0.0299



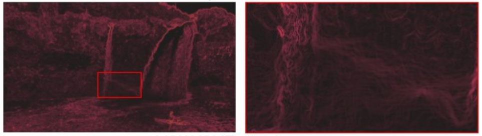
MSU VQMT visualization

JPEG Q=40, Blurring (delta)-y=0.0313



MSU VQMT visualization

JPEG Q=80, Blurring (delta)-y=0.0318



MSU VQMT visualization

MSU Blocking

General info

Metric type	no-reference image metric
Value range	(no blocks) 0..∞ (a lot of blocks)
Value interpretation	bigger is more blocks
MSU VQMT implementations	CPU multithreaded
MSU VQMT visualization	pixel-wise
Available colorspaces	Y
Output values	metric value
Aggregated values	standard set
MSU VQMT usages	-metr blocking

Algorithm description

This metric contains heuristic method for detecting objects edges, which are placed to the edge of the block. In this case metric value is pulled down, allowing to measure blocking more precisely. This metric also considers image contrast around of block and use it as weight for obtained value.

Notes:

- This algirithm considers 8x8 blocks of image, so it applicable only for I frames of video and some of encoders.

Benchmark

Please note: values can vary depending on system configuration, input format and other factors.

**Benchmark on VQMT 14.0 BETA r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti**

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 Blocking	multithreaded	Y	HD 720p	382.95	0.004
VQMT 14.0 Blocking	multithreaded	Y	FullHD 1080p	177.69	0.007
VQMT 14.0 Blocking	multithreaded	Y	4K 2160p	44.1	0.024
VQMT 14.0 Blocking	singlethreaded	Y	HD 720p	90.01	0.012
VQMT 14.0 Blocking	singlethreaded	Y	FullHD 1080p	41.76	0.025
VQMT 14.0 Blocking	singlethreaded	Y	4K 2160p	13.46	0.079

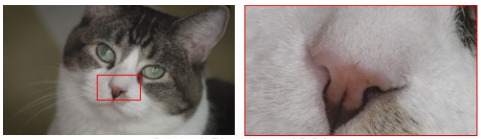
**Benchmark on VQMT 14.0 BETA r12792 PRO for Linux.
CPU: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64 cores
GPU: NVIDIA CUDA/TITAN RTX**

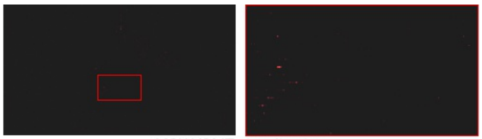
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 Blocking	multithreaded	Y	HD 720p	1242.11	0.002
VQMT 14.0 Blocking	multithreaded	Y	FullHD 1080p	645.54	0.003
VQMT 14.0 Blocking	multithreaded	Y	4K 2160p	174.17	0.01
VQMT 14.0 Blocking	singlethreaded	Y	HD 720p	51.09	0.02
VQMT 14.0 Blocking	singlethreaded	Y	FullHD 1080p	29.2	0.035
VQMT 14.0 Blocking	singlethreaded	Y	4K 2160p	9.47	0.107

Benchmark on VQMT 13.1 r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

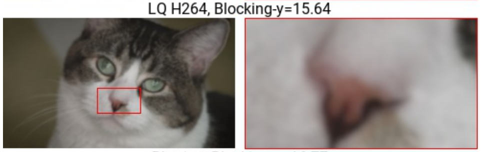
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 Blocking	multithreaded	Y	HD 720p	394.47	0.003
VQMT 13.1 Blocking	multithreaded	Y	FullHD 1080p	182.33	0.006
VQMT 13.1 Blocking	multithreaded	Y	4K 2160p	43.42	0.024
VQMT 13.1 Blocking	singlethreaded	Y	HD 720p	94.09	0.011
VQMT 13.1 Blocking	singlethreaded	Y	FullHD 1080p	43.67	0.024
VQMT 13.1 Blocking	singlethreaded	Y	4K 2160p	14.28	0.073


Original, Blocking-y=12.84


MSU VQMT visualization

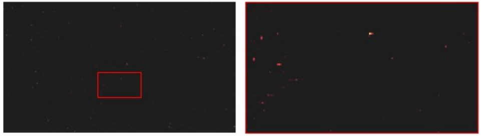
LQ H264, Blocking-y=15.64

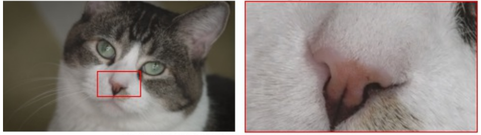
MSU VQMT visualization

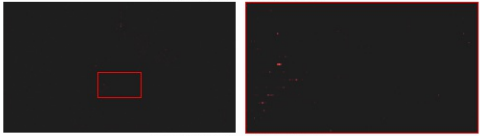

Blurring, Blocking-y=16.77

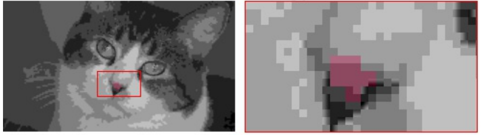
MSU VQMT visualization

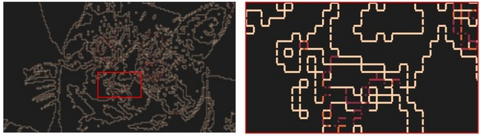
Random points, Blocking-y=18.79

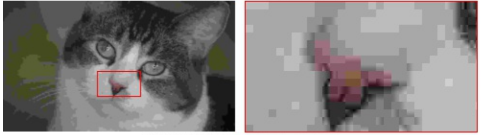

MSU VQMT visualization

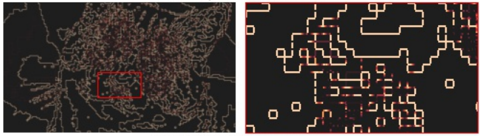

Luminance shift, Blocking-y=12.84

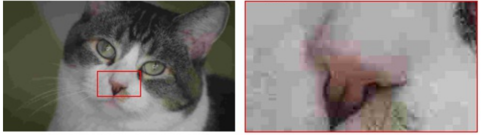

MSU VQMT visualization

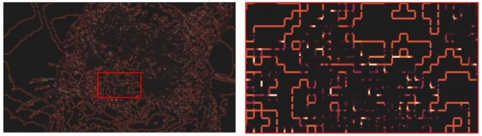

JPEG Q=2, Blocking-y=243.95

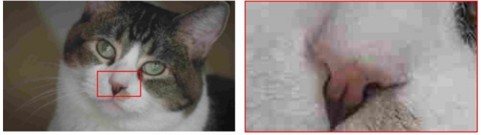

MSU VQMT visualization

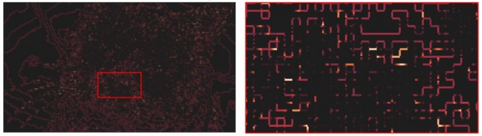

JPEG Q=5, Blocking-y=185.04

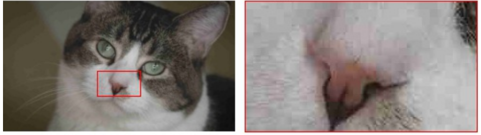

MSU VQMT visualization

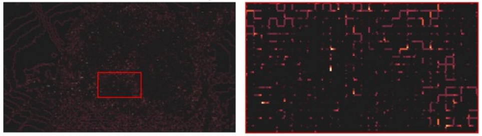

JPEG Q=10, Blocking-y=94.47


MSU VQMT visualization

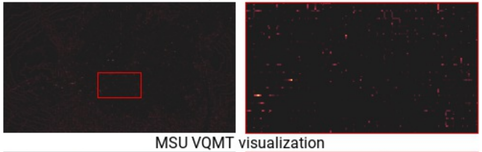

JPEG Q=15, Blocking-y=56.08

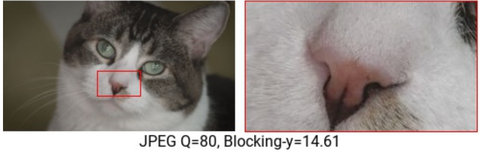

MSU VQMT visualization

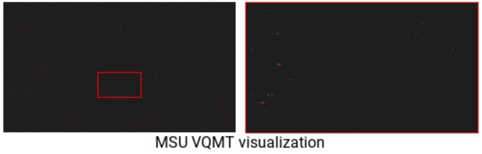

JPEG Q=20, Blocking-y=41.32

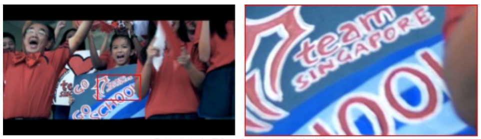

MSU VQMT visualization

JPEG Q=40, Blocking-y=24.01

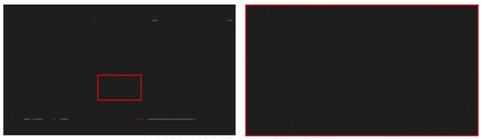

MSU VQMT visualization


JPEG Q=80, Blocking-y=14.61

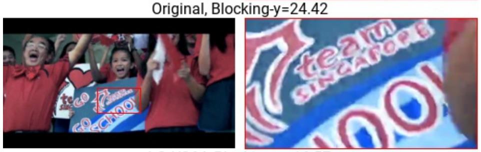

MSU VQMT visualization



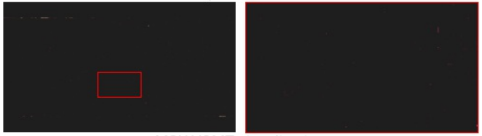
Original, Blocking-y=24.42



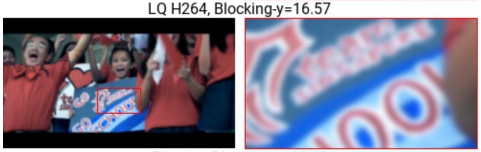
MSU VQMT visualization



LQ H264, Blocking-y=16.57

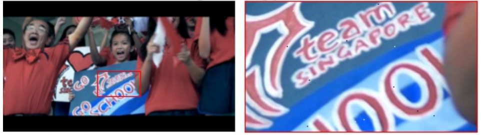


MSU VQMT visualization



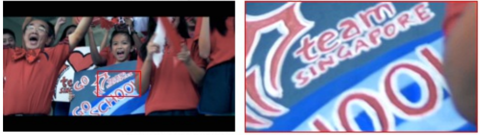
Blurring, Blocking-y=16.48

MSU VQMT visualization



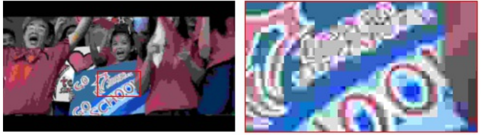
Random points, Blocking-y=30.83

MSU VQMT visualization

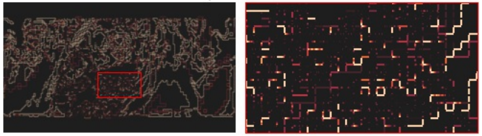


Luminance shift, Blocking-y=24.41

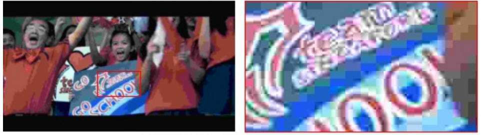
MSU VQMT visualization



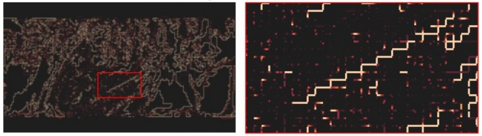
JPEG Q=2, Blocking-y=175.39



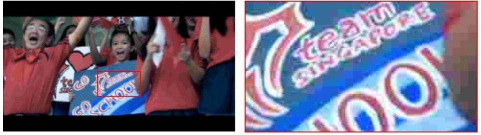
MSU VQMT visualization



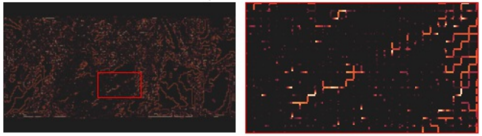
JPEG Q=5, Blocking-y=139.16



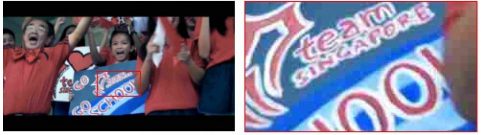
MSU VQMT visualization



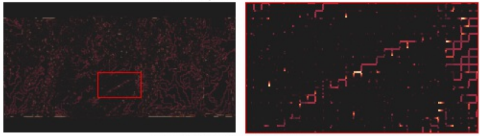
JPEG Q=10, Blocking-y=71.68



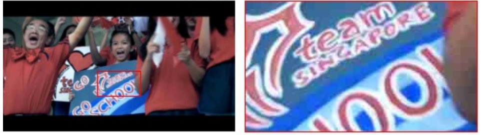
MSU VQMT visualization



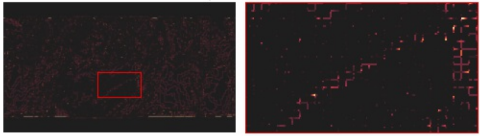
JPEG Q=15, Blocking-y=49.30



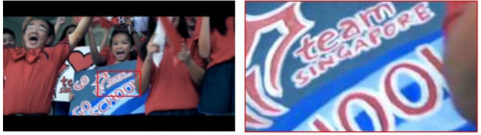
MSU VQMT visualization



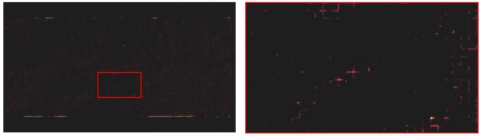
JPEG Q=20, Blocking-y=38.48



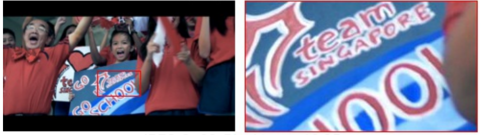
MSU VQMT visualization



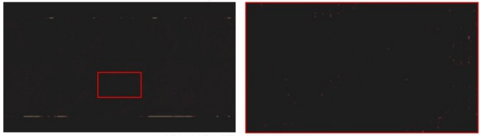
JPEG Q=40, Blocking-y=25.50



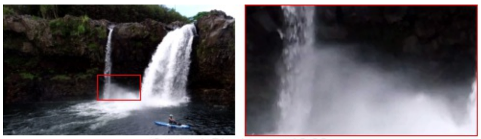
MSU VQMT visualization



JPEG Q=80, Blocking-y=19.35



MSU VQMT visualization

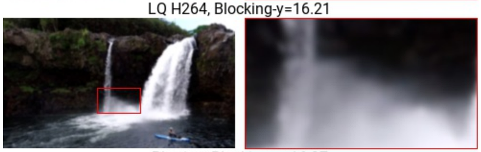


Original, Blocking-y=10.08

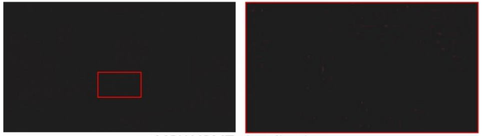
MSU VQMT visualization

LQ H264, Blocking-y=16.21

MSU VQMT visualization



Blurring, Blocking-y=16.97



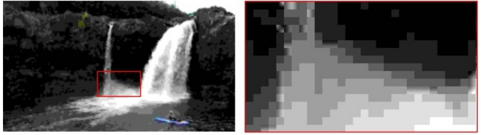
MSU VQMT visualization

Random points, Blocking-y=12.26

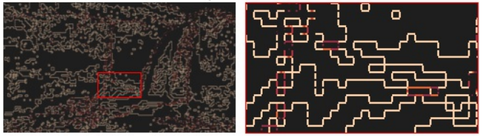
MSU VQMT visualization

Luminance shift, Blocking-y=10.07

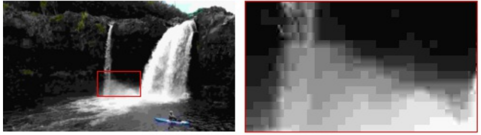
MSU VQMT visualization



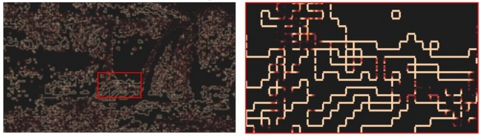
JPEG Q=2, Blocking-y=218.22



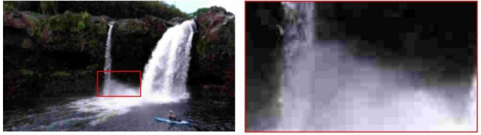
MSU VQMT visualization



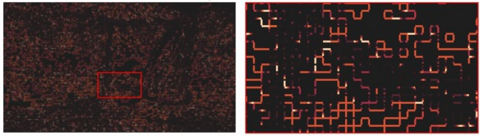
JPEG Q=5, Blocking-y=152.96



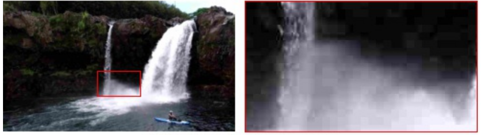
MSU VQMT visualization



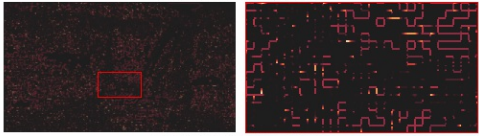
JPEG Q=10, Blocking-y=75.26



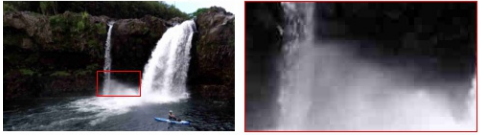
MSU VQMT visualization



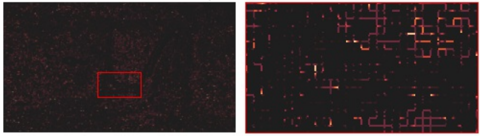
JPEG Q=15, Blocking-y=44.16



MSU VQMT visualization

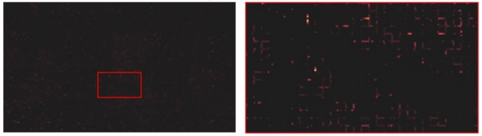


JPEG Q=20, Blocking-y=30.84

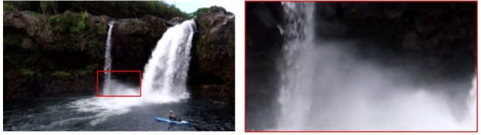


MSU VQMT visualization

JPEG Q=40, Blocking-y=16.21



MSU VQMT visualization



JPEG Q=80, Blocking-y=9.42

MSU VQMT visualization

SI (Spatial Information)

General info

Metric type	no-reference image metric
Value range	(simple, monotone frame) 0..1 (very complex frame)
Value interpretation	bigger more complex frame
MSU VQMT implementations	CPU multithreaded
MSU VQMT visualization	pixel-wise, sobel transformation
Available colorspaces	Y
Output values	metric value
Aggregated values	standard set
MSU VQMT usages	-metr si
External links	ITU-T Recommendation P.910: Subjective video quality assessment methods for multimedia applications, 1999. – 37 p.

Algorithm description

This metric measures complexity (entropy) of an input image. This metric represents simplest SI realization that takes standard deviation of sequence of pixel values (Y-component) of Sobel transformation of input image:

$$\mathrm{SI}(X) = \mathrm{std}[\mathrm{Sobel}(X(x,y))]$$

Sobel is length of vector $(\mathrm{Sobel}_h, \mathrm{Sobel}_v)$, where Sobel_h and Sobel_v are horizontal and vertical Sobel transformation. While calculating Sobel, the edge pixels will be excluded from calculation.

Benchmark

Please note: values can vary depending on system configuration, input format and other factors.

**Benchmark on VQMT 14.0 BETA r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti**

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 SI	multithreaded	Y	HD 720p	368.29	0.004
VQMT 14.0 SI	multithreaded	Y	FullHD 1080p	168.04	0.007
VQMT 14.0 SI	multithreaded	Y	4K 2160p	42.4	0.025
VQMT 14.0 SI	singlethreaded	Y	HD 720p	81.06	0.013
VQMT 14.0 SI	singlethreaded	Y	FullHD 1080p	36.8	0.028
VQMT 14.0 SI	singlethreaded	Y	4K 2160p	11.64	0.091

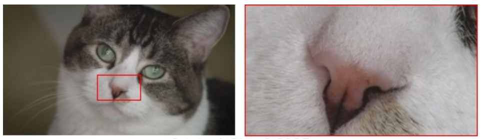
**Benchmark on VQMT 14.0 BETA r12792 PRO for Linux.
CPU: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64 cores**

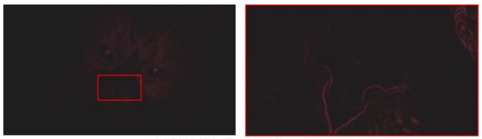
GPU: NVIDIA CUDA/TITAN RTX

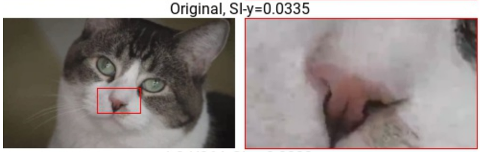
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 SI	multithreaded	Y	HD 720p	1301.78	0.002
VQMT 14.0 SI	multithreaded	Y	FullHD 1080p	693.15	0.003
VQMT 14.0 SI	multithreaded	Y	4K 2160p	176.92	0.01
VQMT 14.0 SI	singlethreaded	Y	HD 720p	83.92	0.012
VQMT 14.0 SI	singlethreaded	Y	FullHD 1080p	39.49	0.026
VQMT 14.0 SI	singlethreaded	Y	4K 2160p	14.64	0.07

Benchmark on VQMT 13.1 r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

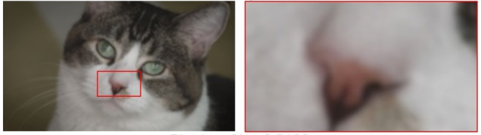
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 SI	multithreaded	Y	HD 720p	371.38	0.003
VQMT 13.1 SI	multithreaded	Y	FullHD 1080p	169.24	0.007
VQMT 13.1 SI	multithreaded	Y	4K 2160p	42.15	0.025
VQMT 13.1 SI	singlethreaded	Y	HD 720p	81.56	0.013
VQMT 13.1 SI	singlethreaded	Y	FullHD 1080p	36.85	0.028
VQMT 13.1 SI	singlethreaded	Y	4K 2160p	11.7	0.089


Original, SI-y=0.0335


MSU VQMT visualization

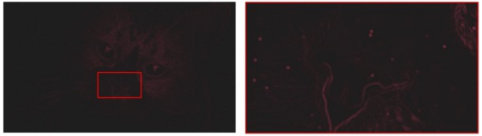

LQ H264, SI-y=0.0233

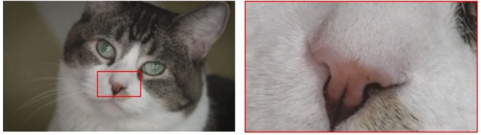
MSU VQMT visualization


Blurring, SI-y=0.0102

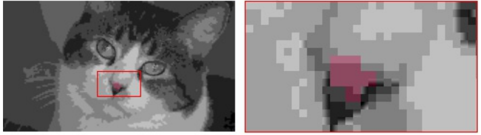
MSU VQMT visualization

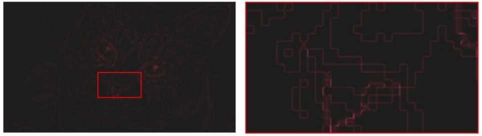
Random points, SI-y=0.0364

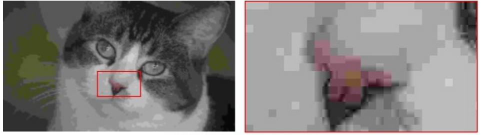

MSU VQMT visualization

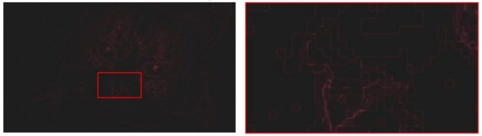

Luminance shift, SI-y=0.0335

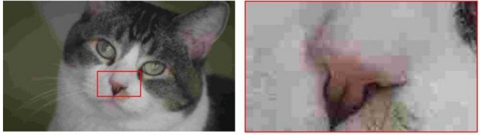
MSU VQMT visualization


JPEG Q=2, SI-y=0.0380


MSU VQMT visualization


JPEG Q=5, SI-y=0.0348

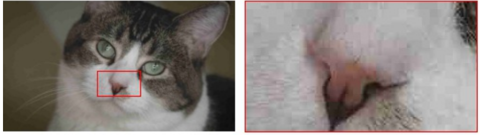

MSU VQMT visualization


JPEG Q=10, SI-y=0.0338

MSU VQMT visualization

JPEG Q=15, SI-y=0.0333

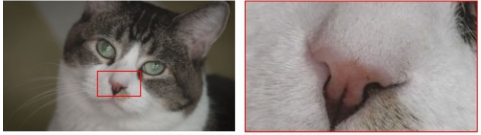
MSU VQMT visualization


JPEG Q=20, SI-y=0.0333

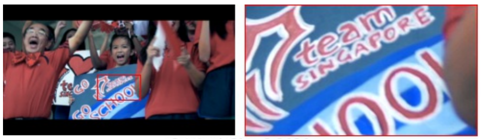
MSU VQMT visualization

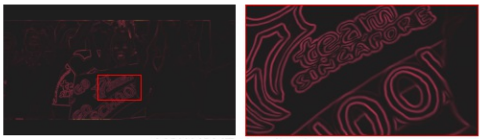
JPEG Q=40, SI-y=0.0335

MSU VQMT visualization

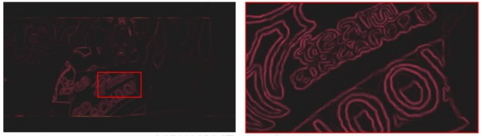

JPEG Q=80, SI-y=0.0336

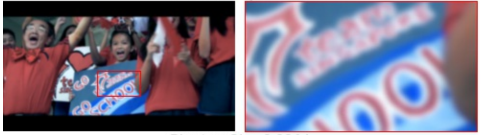
MSU VQMT visualization

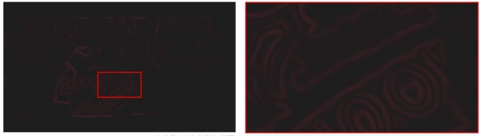

Original, SI-y=0.0626

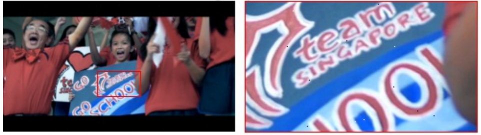

MSU VQMT visualization

LQ H264, SI-y=0.0592

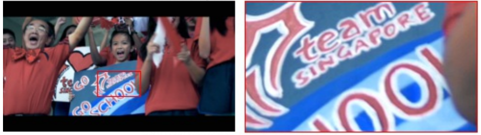

MSU VQMT visualization


Blurring, SI-y=0.0264

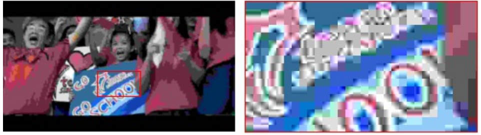

MSU VQMT visualization

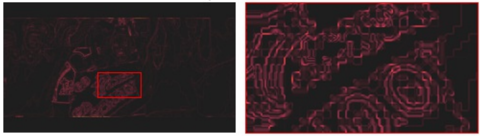

Random points, SI-y=0.0649

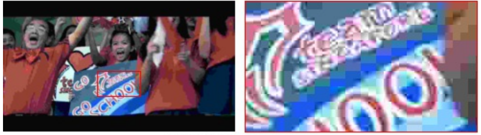
MSU VQMT visualization

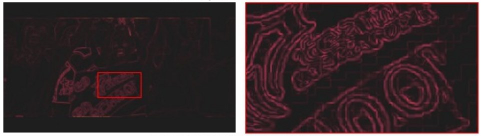

Luminance shift, SI-y=0.0626

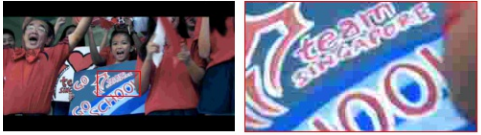
MSU VQMT visualization

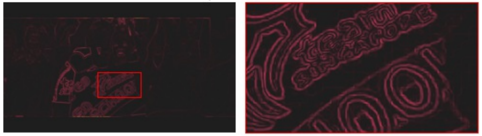

JPEG Q=2, SI-y=0.0718

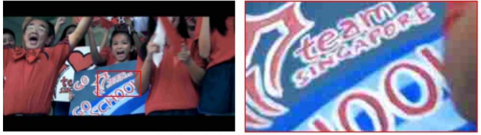

MSU VQMT visualization

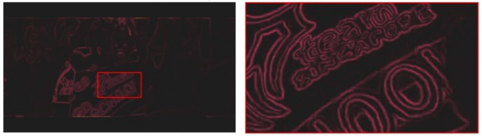

JPEG Q=5, SI-y=0.0653

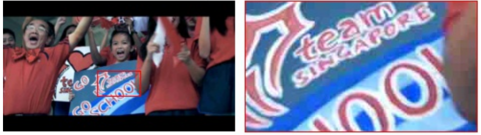

MSU VQMT visualization


JPEG Q=10, SI-y=0.0631

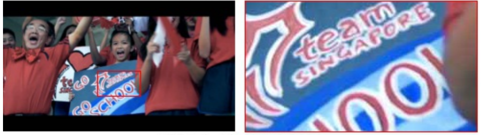

MSU VQMT visualization

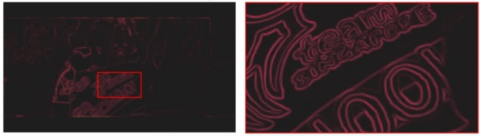

JPEG Q=15, SI-y=0.0627

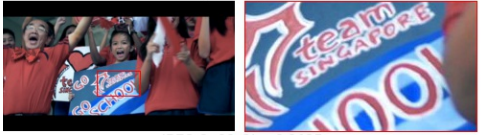

MSU VQMT visualization

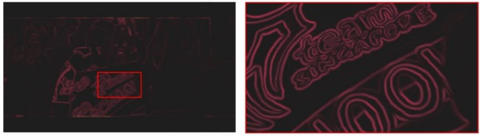

JPEG Q=20, SI-y=0.0626

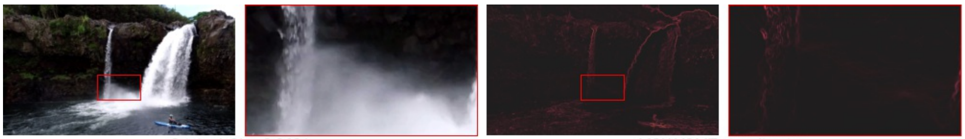
MSU VQMT visualization


JPEG Q=40, SI-y=0.0624

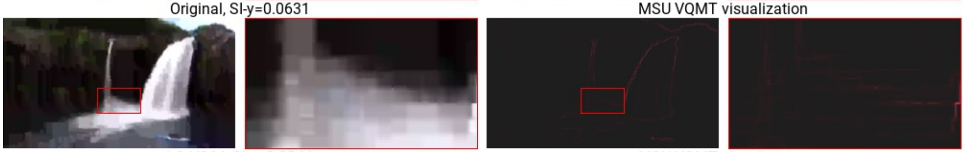

MSU VQMT visualization


JPEG Q=80, SI-y=0.0625

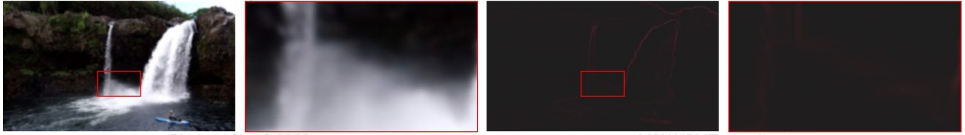

MSU VQMT visualization



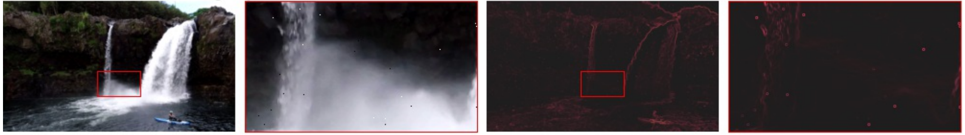
Original, SI-y=0.0631 MSU VQMT visualization



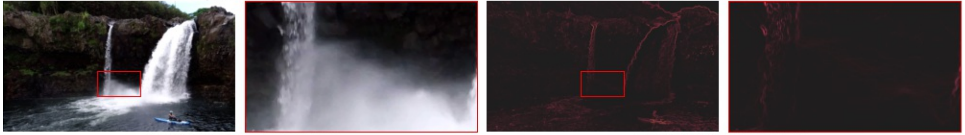
LQ H264, SI-y=0.0244 MSU VQMT visualization



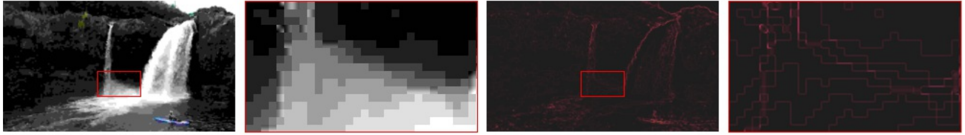
Blurring, SI-y=0.0203 MSU VQMT visualization



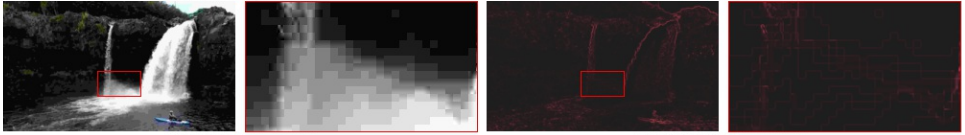
Random points, SI-y=0.0650 MSU VQMT visualization



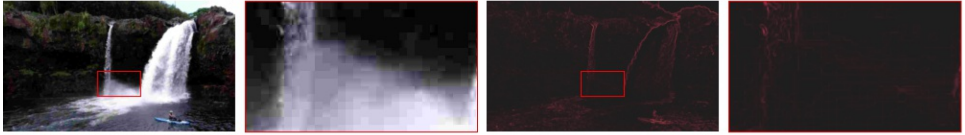
Luminance shift, SI-y=0.0631 MSU VQMT visualization



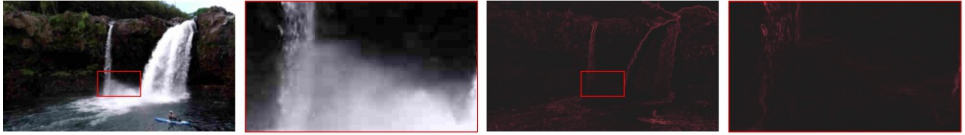
JPEG Q=2, SI-y=0.0634 MSU VQMT visualization



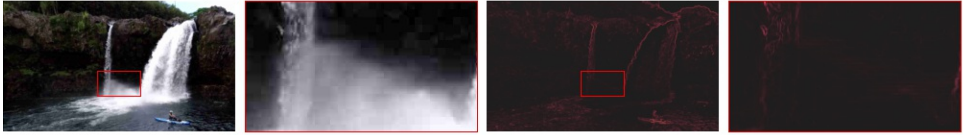
JPEG Q=5, SI-y=0.0610 MSU VQMT visualization



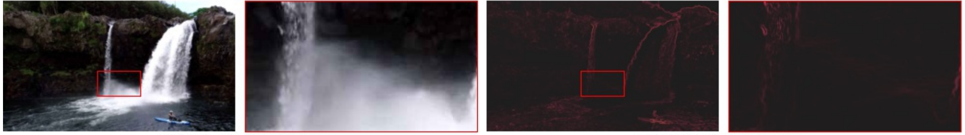
JPEG Q=10, SI-y=0.0602 MSU VQMT visualization



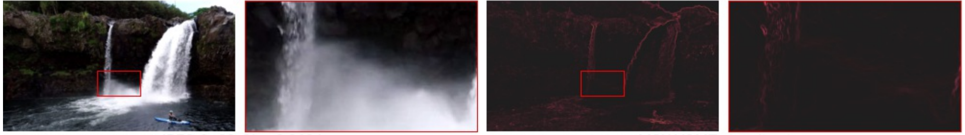
JPEG Q=15, SI-y=0.0603 MSU VQMT visualization



JPEG Q=20, SI-y=0.0603 MSU VQMT visualization



JPEG Q=40, SI-y=0.0614 MSU VQMT visualization



JPEG Q=80, SI-y=0.0624 MSU VQMT visualization

TI (Temporal Information)

General info

Metric type	no-reference temporal metric
Value range	(simple, static video) 0..1 (very diverse frames)
Value interpretation	bigger more diverse frames
MSU VQMT implementations	CPU multithreaded
MSU VQMT visualization	pixel-wise, difference between adjacent frames
Available colorspaces	Y
Output values	metric value
Aggregated values	standard set
MSU VQMT usages	-metr ti
External links	ITU-T Recommendation P.910: Subjective video quality assessment methods for multimedia applications, 1999. – 37 p.

Algorithm description

This metric measures complexity (entropy) of defference between consequent frames of input video. This metric represents simplest TI realization that takes standard deviation of sequence of differences of corresponding pixel values of a frame and previous frame:

$$\mathrm{TI}(V) = \frac{\mathrm{std}[V_n(X(x,y)) - V_{n-1}(X(x,y))]}{\max}$$

Benchmark

Please note: values can vary depending on system configuration, input format and other factors.

Benchmark on VQMT 14.0 BETA r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

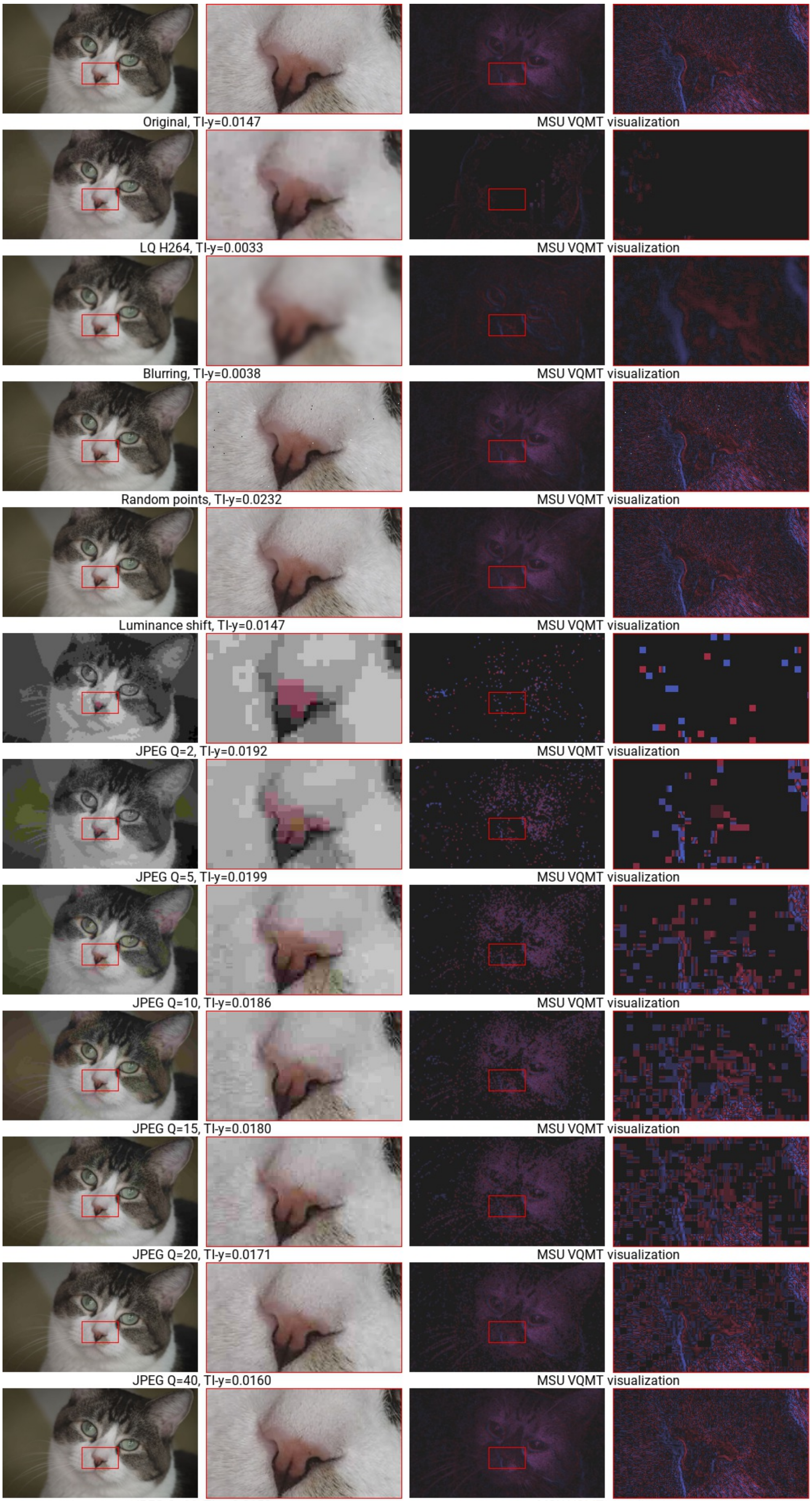
Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 TI	multithreaded	Y	HD 720p	706.82	0.002
VQMT 14.0 TI	multithreaded	Y	FullHD 1080p	290.23	0.005
VQMT 14.0 TI	multithreaded	Y	4K 2160p	70.23	0.016
VQMT 14.0 TI	singlethreaded	Y	HD 720p	166.74	0.007
VQMT 14.0 TI	singlethreaded	Y	FullHD 1080p	65.31	0.016
VQMT 14.0 TI	singlethreaded	Y	4K 2160p	21.69	0.049

Benchmark on VQMT 14.0 BETA r12792 PRO for Linux.
CPU: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 64 cores
GPU: NVIDIA CUDA/TITAN RTX

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 14.0 TI	multithreaded	Y	HD 720p	1459.05	0.001
VQMT 14.0 TI	multithreaded	Y	FullHD 1080p	729.13	0.003
VQMT 14.0 TI	multithreaded	Y	4K 2160p	127.44	0.012
VQMT 14.0 TI	singlethreaded	Y	HD 720p	135.93	0.007
VQMT 14.0 TI	singlethreaded	Y	FullHD 1080p	59.74	0.017
VQMT 14.0 TI	singlethreaded	Y	4K 2160p	21.09	0.048

Benchmark on VQMT 13.1 r12792 PRO for Windows.
CPU: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8 cores
GPU: NVIDIA CUDA/GeForce GTX 660 Ti

Implementation	System & settings	Colors	Resolution	FPS	Sec. per frame
VQMT 13.1 TI	multithreaded	Y	HD 720p	719.8	0.002
VQMT 13.1 TI	multithreaded	Y	FullHD 1080p	280.3	0.004
VQMT 13.1 TI	multithreaded	Y	4K 2160p	67.05	0.016
VQMT 13.1 TI	singlethreaded	Y	HD 720p	169.36	0.007
VQMT 13.1 TI	singlethreaded	Y	FullHD 1080p	66.09	0.016
VQMT 13.1 TI	singlethreaded	Y	4K 2160p	21.42	0.049



Original, TI-y=0.0147

MSU VQMT visualization

LQ H264, TI-y=0.0033

MSU VQMT visualization

Blurring, TI-y=0.0038

MSU VQMT visualization

Random points, TI-y=0.0232

MSU VQMT visualization

Luminance shift, TI-y=0.0147

MSU VQMT visualization

JPEG Q=2, TI-y=0.0192

MSU VQMT visualization

JPEG Q=5, TI-y=0.0199

MSU VQMT visualization

JPEG Q=10, TI-y=0.0186

MSU VQMT visualization

JPEG Q=15, TI-y=0.0180

MSU VQMT visualization

JPEG Q=20, TI-y=0.0171

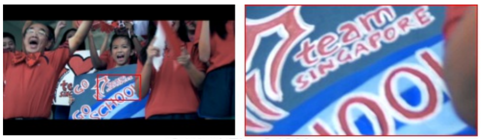
MSU VQMT visualization

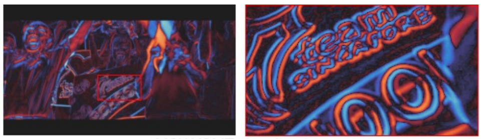
JPEG Q=40, TI-y=0.0160

MSU VQMT visualization

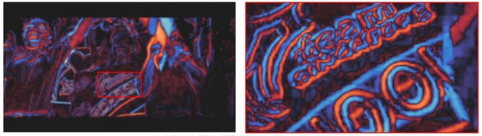
JPEG Q=80, TI-y=0.0152

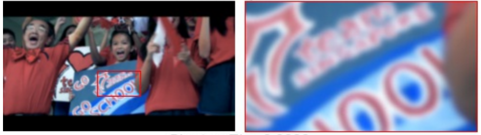
MSU VQMT visualization

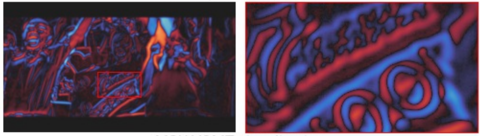

Original, TI-y=0.1066

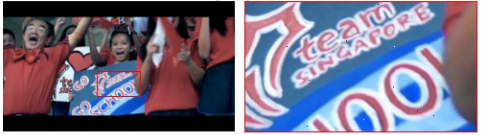

MSU VQMT visualization

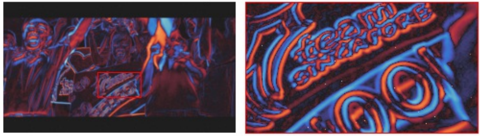
LQ H264, TI-y=0.1053

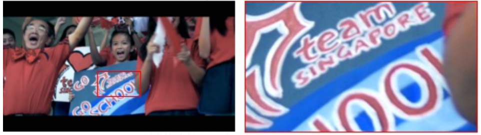

MSU VQMT visualization

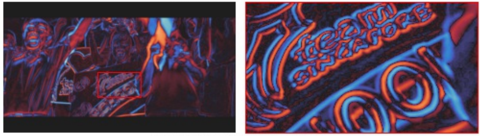

Blurring, TI-y=0.0823

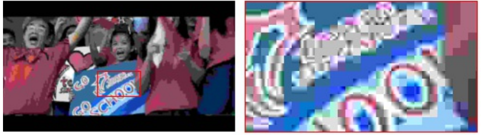

MSU VQMT visualization

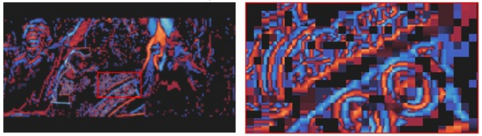

Random points, TI-y=0.1075

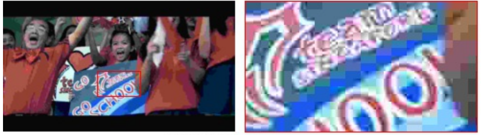

MSU VQMT visualization

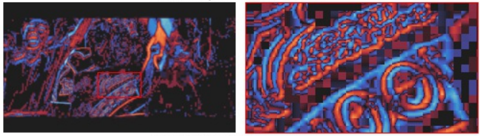

Luminance shift, TI-y=0.1065

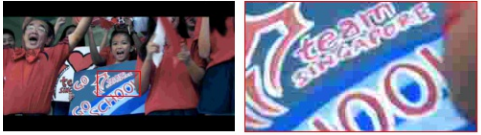

MSU VQMT visualization

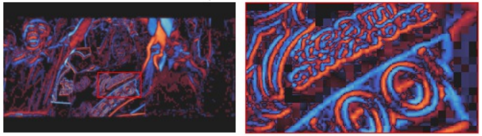

JPEG Q=2, TI-y=0.1121

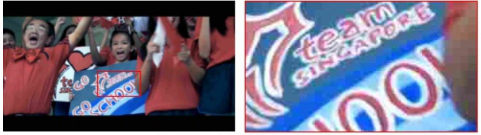

MSU VQMT visualization

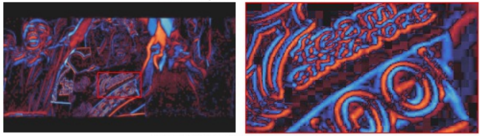

JPEG Q=5, TI-y=0.1079

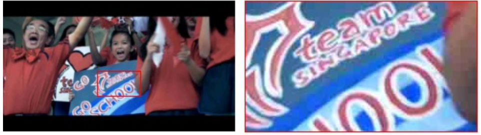

MSU VQMT visualization

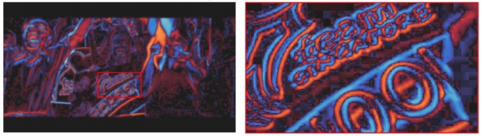

JPEG Q=10, TI-y=0.1068

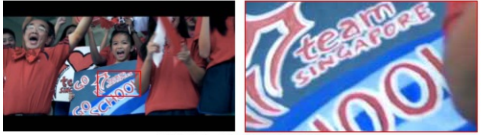

MSU VQMT visualization

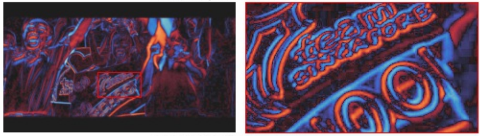

JPEG Q=15, TI-y=0.1061

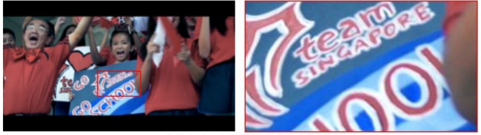

MSU VQMT visualization

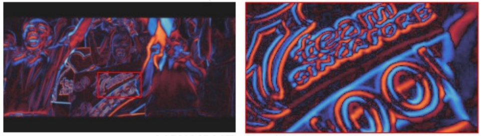

JPEG Q=20, TI-y=0.1058

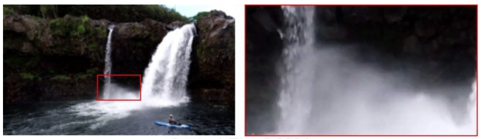

MSU VQMT visualization

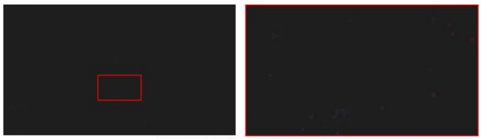

JPEG Q=40, TI-y=0.1057

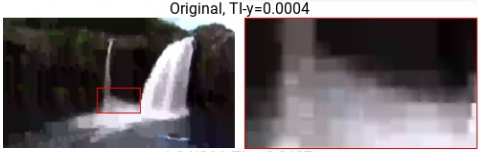

MSU VQMT visualization

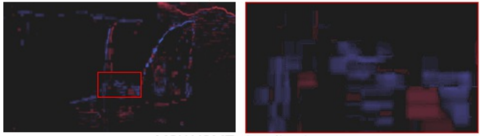

JPEG Q=80, TI-y=0.1056

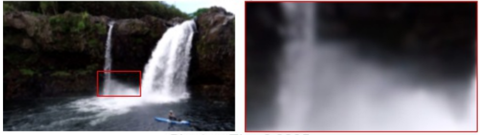

MSU VQMT visualization

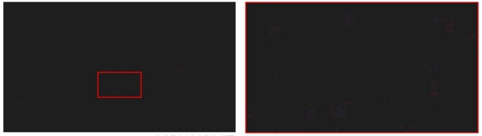

Original, TI-y=0.0004


MSU VQMT visualization


LQ H264, TI-y=0.0127


MSU VQMT visualization

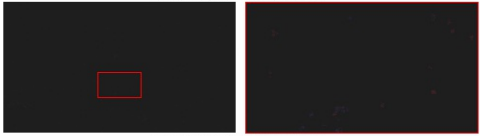

Blurring, TI-y=0.0005

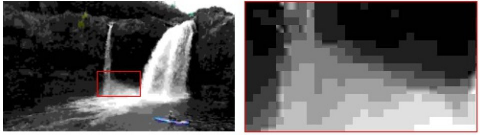

MSU VQMT visualization

Random points, TI-y=0.0206

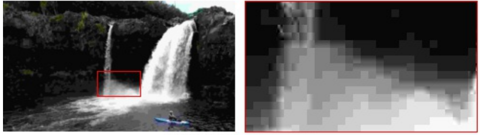
MSU VQMT visualization

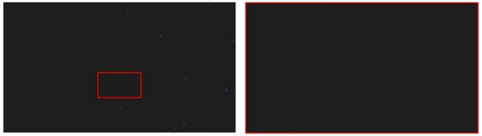
Luminance shift, TI-y=0.0004


MSU VQMT visualization


JPEG Q=2, TI-y=0.0021

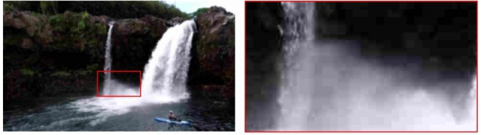
MSU VQMT visualization

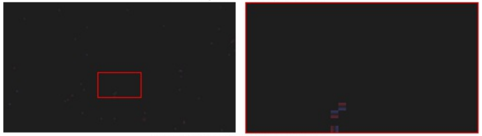

JPEG Q=5, TI-y=0.0016

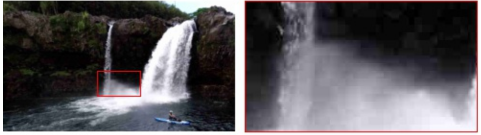

MSU VQMT visualization

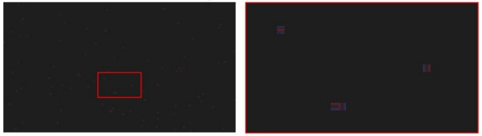
JPEG Q=10, TI-y=0.0011

MSU VQMT visualization

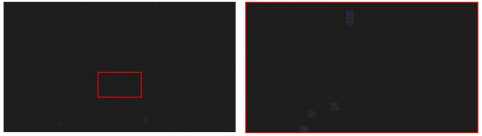

JPEG Q=15, TI-y=0.0011

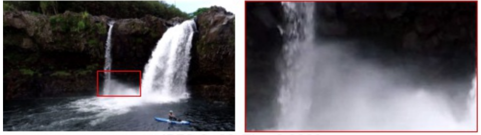

MSU VQMT visualization


JPEG Q=20, TI-y=0.0013


MSU VQMT visualization

JPEG Q=40, TI-y=0.0010


MSU VQMT visualization


JPEG Q=80, TI-y=0.0009

MSU VQMT visualization

About us

COPRESSION.RU Team initially developed MSU VQMT for the Annual codec comparison by Graphics & Media Lab Video Group.

Video Group is a part of the Graphics & Media Lab of the Computer Science Department in Moscow State University. The history of Graphics Group began at the end of 1980's. Graphics & Media Lab was officially founded in 1998. The main research directions of the Lab lie in different areas of Computer Graphics, Computer Vision, and Media Processing (audio, image, and video processing). Some of the research results were patented, other results were presented in a number of publications.

The main research directions of Graphics & Media Lab Video Group are video processing (pre-, post- and video analysis filters) and video compression (codecs' testing and tuning, quality metrics research, development of codecs).

We are really glad to work many years with companies like Intel, Samsung, Huawei. RealNetworks and others.

Mutual collaboration in areas of video processing and video compression is always interesting for us.

Images in this documentation by Compression.RU Team and MSU Video Group distributed under the terms of Creative Commons Attribution Lincense. In the examples we used materials from the following sources:

- https://vimeo.com/162777344 by AJA Video Systems
- https://vimeo.com/311282786 by Harold Aune
- https://vimeo.com/123907536 by Glenn Ng

E-mail: video-measure@compression.ru