Pós-Graduação em Ciência da Computação

"Shifted Gradient Similarity: A perceptual video quality assessment index for adaptive streaming encoding"

By

Estêvão Chaves Monteiro

M.Sc. Dissertation

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE/2016

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ESTÊVÃO CHAVES MONTEIRO

"SHIFTED GRADIENT SIMILARITY: A perceptual video quality assessment index for adaptive streaming encoding"

THIS WORK HAS BEEN SUBMITTED TO PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO OF CENTRO DE INFORMÁTICA OF UNIVERSIDADE FEDERAL DE PERNAMBUCO AS A PARTIAL REQUIREMENT FOR ACHIEVING THE DEGREE OF MASTER IN COMPUTER SCIENCE.

ADVISOR: CARLOS ANDRÉ GUIMARÃES FERRAZ

RECIFE/2016

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

M775s Monteiro, Estêvão Chaves.
Shifted gradient similarity: a perceptual video quality assessment index for adaptive streaming encoding / Estêvão Chaves Monteiro. – 2016.
116 f.: il., fig., tab.

Orientador: Carlos André Guimarães Ferraz.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2016.
Inclui referências e apêndices.

1. Processamento de imagens. 2. Qualidade visual. 3. Compressão de vídeo. I. Ferraz, Carlos André Guimarães (orientador). II. Título.

621.367 CDD (23. ed.) UFPE- MEI 2016-032

Estêvão Chaves Monteiro

Shifted Gradient Similarity: a perceptual visual quality index for adaptive streaming encoding

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação

Aprovado em: 04/03/2016.

BANCA EXAMINADORA

_____

Prof. Dr. Carlos Alexandre Barros de Mello
Centro de Informática / UFPE

_____

Prof. Dr. Celso Alberto Saibel Santos
Departamento de Informática / UFES

_____

Prof. Dr. Tsang Ing Ren
Centro de Informática / UFPE

_____

Prof. Dr. Carlos André Guimarães Ferraz
Centro de Informática / UFPE
(**Orientador**)

Aos meus pais, que dedicaram, em palavra e exemplo, caráter, integridade, devoção, sacrifício e humildade. Professores da vida e do espírito. Tudo o que sou, serei, fiz, farei e não farei, carregam seus ensinamentos e amor.

Às minhas três irmãs e meus três irmãos, que vi nascer e crescer, com os quais aprendi convivência, tolerância, generosidade e liderança.

A Rafael, irmão de coração e da vida, sempre presente a dividir as dificuldades e comemorar as vitórias.

A Ludmila, espírito de luz e amor que me acalma e me inflama, que me guia e se perde comigo, sempre a oferecer um sorriso fácil e sincero.

ACKNOWLEDGEMENTS

To Prof. Carlos Ferraz, for his humble and honest friendship, for his good spirits and patience, for offering incentive, orientation and investment. An exemplary professor.

To Profs. Tsang and Roberto Barros, for believing in my work, for their advisory and patience, and for their personal investments in my undertaking.

To Ricardo Scholz, for our friendship, teamwork and sharing of experiences.

To the Development Department of Serviço Federal de Processamento de Dados – SERPRO, especially represented by Simone Ramos, Ézio Oliveira and Patrícia Batista, for the unshakable belief in my competence, my success, and the value of my research, and for the efforts in conciliating my time dedicated to the department with the time dedicated to the university.

To all collaborators of the Centro de Informática whom directly or indirectly contributed for this work, particularly the professionalism of the Pós-Graduação office, and the investment of the coordination to the presentation of my work in the exterior.

To the fine professionals of Nanyang Technical University of Singapore, for their impeccable courtesy, patience, professionalism and good spirits.

"Through our eyes, the universe is perceiving itself. Through our ears, the universe is listening to its harmonies. We are the witnesses through which the universe becomes conscious of its glory, of its magnificence."

Alan W. Watts

RESUMO

Cada vez mais serviços de *streaming* de vídeo estão migrando para o modelo adaptativo, devido à crescente diversidade de dispositivos pessoais conectados à Web e à popularidade das redes sociais. Limitações comuns na largura de banda de Internet, velocidade de decodificação e potência de baterias disponíveis em tais dispositivos desafiam a eficiência dos codificadores de conteúdo para preservar a qualidade visual em taxas de dados reduzidas e abrangendo uma ampla gama de resoluções de tela, tipicamente comprimindo para menos de 1% da massiva taxa de dados bruta. Ademais, o sistema visual humano não percebe uniformemente as perdas de informação espacial e temporal, então um modelo objetivo físico simples como a média do erro quadrático não se correlaciona bem com qualidade perceptível. Técnicas de avaliação e predição objetiva de qualidade perceptível de conteúdo visual se aprimoraram amplamente na última década, mas o problema permanece em aberto.

Dentre as métricas de qualidade psicovisual mais relevantes estão muitas versões do índice de *similaridade estrutural (Structural Similarity – SSIM)*. No presente trabalho, várias das mais eficientes métricas baseadas em SSIM, como o *Multi-Scale Fast SSIM* e o *Gradient Magnitude Similarity Deviation (GMSD)*, são decompostas em suas técnicas-componentes e recombinadas para se obter medidas e entendimento sobre a contribuição de cada técnica e se desenvolver aprimoramentos à sua qualidade e eficiência. Tais métricas são aplicadas às bases de dados *LIVE Mobile Video Quality* e *TID2008* e os resultados são correlacionados aos dados subjetivos incluídos naquelas bases na forma de *escores de opinião subjetiva (mean opinion score – MOS)*, de modo que o grau de correlação de cada métrica indique sua capacidade de predizer qualidade perceptível. Investiga-se, ainda, a aplicabilidade das métricas à recente e relevante implementação de *otimização psicovisual de distorção por taxa (psychovisual rate-distortion optimization – Psy-RDO)* do codificador x264, ao qual atualmente falta uma métrica de avaliação objetiva ideal.

O índice *"Shifted Gradient Similarity" (SG-Sim)* é proposto com uma técnica aprimorada de realce de imagem que evita uma perda não-pretendida de informação

de análise, comum em índices baseados em SSIM, assim alcançando correlação consideravelmente maior com MOS comparado às métricas existentes investigadas neste trabalho. Também são propostos filtros de consolidação espacial mais eficientes: o filtro gaussiano de inteiros 1-D decomposto e limitado a dois desvios padrão e o filtro "box" subamostrado baseado na imagem integral, os quais retém, respectivamente, 99% e 98% de equivalência e obtém ganhos de velocidade de, respectivamente, 68% e 382%. O filtro subamostrado também promove escalabilidade, especialmente para conteúdo de *ultra-alta definição*, e define a versão do índice "*Fast SG-Sim*". Ademais, verifica-se que o SG-Sim aumenta a correlação com *Psy-RDO*, indicando-se uma métrica de qualidade de codificação ideal para o *x264*. Finalmente, os algoritmos e experimentos usados neste trabalho estão implementados no software "*Video Quality Assessment in Java*" (*jVQA*), baseado nas plataformas *AviSynth* e *FFmpeg* e que é projetado para personalização e extensibilidade, suportando conteúdo *ultra-alta definição* "4K" e disponibilizado como código-fonte aberto e livre.

Palavras-chave: Qualidade de imagem digital. Compressão de vídeo. Predição objetiva de qualidade subjetiva. Escore de opinião subjetiva – *MOS*. Otimização de distorção por taxa – *RDO*. Eficiência de algoritmo. *Streaming* adaptativo. Similaridade estrutural – *SSIM*. Padrão *MPEG AVC/H.264*.

ABSTRACT

Adaptive video streaming has become prominent due to the rising diversity of Web-enabled personal devices and the popularity of social networks. Common limitations in Internet bandwidth, decoding speed and battery power available in such devices challenge the efficiency of content encoders to preserve visual quality at reduced data rates over a wide range of display resolutions, typically compressing to lower than 1% of the massive raw data rate. Furthermore, the human visual system does not uniformly perceive losses of spatial and temporal information, so a simple physical objective model such as the mean squared error does not correlate well with perceptual quality. Objective assessment and prediction of perceptual quality of visual content has greatly improved in the past decade, but remains an open problem.

Among the most relevant psychovisual quality metrics are the many versions of the Structural Similarity (SSIM) index. In this work, several of the most efficient SSIM-based metrics, such as the Multi-Scale Fast SSIM and the Gradient Magnitude Similarity Deviation (GMSD), are decomposed into their component techniques and reassembled in order to measure and understand the contribution of each technique and to develop improvements in quality and efficiency. The metrics are applied to the LIVE Mobile Video Quality and TID2008 databases and the results are correlated to the subjective data included in the databases in the form of mean opinion scores (MOS), so each metric's degree of correlation indicates its ability to predict perceptual quality. Additionally, the metrics' applicability to the recent, relevant psychovisal rate-distortion optimization (Psy-RDO) implementation in the x264 encoder, which currently lacks an ideal objective assessment metric, is investigated as well.

The "Shifted Gradient Similarity" (SG-Sim) index is proposed with an improved feature enhancement by avoiding a common unintended loss of analysis information in SSIM-based indexes, and achieving considerably higher MOS correlation than the existing metrics investigated in this work. More efficient spatial pooling filters are proposed, as well: the decomposed 1-D integer Gaussian filter limited to two

standard deviations, and the downsampling Box filter based on the integral image, which retain respectively 99% and 98% equivalence and achieve speed gains of respectively 68% and 382%. In addition, the downsampling filter also enables broader scalability, particularly for Ultra High Definition content, and defines the "Fast SG-Sim" index version. Furthermore, SG-Sim is found to improve correlation with Psy-RDO, as an ideal encoding quality metric for x264. Finally, the algorithms and experiments used in this work are implemented in the "Video Quality Assessment in Java" (jVQA) software, based on the AviSynth and FFmpeg platforms, and designed for customization and extensibility, supporting 4K Ultra-HD content and available as free, open source code.

Keywords: Digital image quality. Video compression. Objective prediction of subjective quality. Mean opinion score – MOS. Rate-distortion optimization – RDO. Algorithm efficiency. Adaptive streaming. Structural similarity – SSIM. MPEG AVC/H.264 standard.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

1080p	1080 progressive horizontal lines.
1-D	Unidimensional.
2-D	Bidimensional.
3-SSIM	Three-Component Structural Similarity.
4K	Four thousand vertical lines.
4-SSIM	Four-Component Structural Similarity.
720p	720 progressive horizontal lines.
API	Application programming interface.
AQ	Adaptive quantization.
AVC	Advanced Video Coding.
AVI	Audio/Video Interleave.
AVS	AviSynth (script).
B-frame	Bidirectional predicted frame.
CDN	Content delivery network.
CLI	Command-line interface.
CPU	Central processing unit.
CSIQ	Categorical Subjective Image Quality Database.
CSS	Cascading Style Sheets.
CSV	Comma-separated values.
DASH	Dynamic Adaptive Streaming over HTTP.
DCT	Discrete cosine transform.
DLL	Dynamic linked library.
DMOS	Differential mean opinion score.
DRM	Digital Rights Management.
DVD	Digital Video Disc.
FOSS	Free and open-source software.
FR	Full reference.
FWVGA	Full Wide Video Graphics Array.
GMSD	Gradient Magnitude Similarity Deviation.
G-SSIM	Gradient Structural Similarity.
GPU	Graphics processing unit.
GUI	Graphical user interface.
HD	High definition.
HDTV	High definition television.
HEVC	High Efficiency Video Coding.
HTML	Hypertext Markup Language.
HTTP	Hypertext Transport Protocol.
HVS	Human visual system.
IDR-frame	Instantaneous decoder refresh frame.
IEC	International Electrotechnical Commission.
I-frame	Intra-predicted frame.
IQA	Image quality assessment.
ISO	International Standards Organization.
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector.
JNA	Java Native Access.
jNAvi	Java Native Access for Avisynth.

JNI	Java Native Interface.
JS	JavaScript.
JVM	Java Virtual Machine.
jVQA	Video Quality Assessment in Java.
LCC	Linear correlation coefficient.
LIVE	Laboratory for Image and Video Engineering.
ME	Motion estimation.
MOS	Mean opinion score.
MOVIE	Motion-based Video Integrity Evaluation
MSE	Mean squared error.
MS-SG-Sim	Multi-Scale Shifted Gradient Similarity.
MS-SSIM	Multi-Scale SSIM.
MSU	Moscow State University.
MPEG	Motion Pictures Expert Group.
NAT	Network address translation
OO	Object-oriented.
OS	Operating system.
P-frame	Predictive frame.
PNG	Portable Network Graphics.
PRR	Pixel ratio root.
PSNR	Peak signal-to-noise ratio.
Psy-RDO	Psychovisual rate-distortion optimization.
qHD	Quarter High Definition.
QP	Quantization parameter.
QVGA	Quarter Video Graphics Array.
RCC	Rank correlation coefficient.
RDO	Rate-distortion optimization.
RGB	Red-Green-Blue.
RMSE	Root mean squared error.
RTP	Real-Time Transport Protocol.
RTCP	Real-Time Transport Protocol Control Protocol.
RTSP	Real Time Streaming Protocol.
SD	Standard definition.
SG-Sim	Shifted Gradient Similarity.
SNR	Signal-to-noise ratio.
SSIM	Structural Similarity.
ST-VSSIM	Spatio-Temporal Video Structural Similarity.
SVC	Scalable Video Coding.
TID2008	Tampere Image Database 2008.
UML	Unified Modeling Language.
VAQ	Variance-based adaptive quantization.
VBV	Video buffer verification.
VfW	Video for Windows.
VGA	Video Graphics Array.
VOD	Video on demand.
VQA	Video/visual quality assessment.
VQEG	Video Quality Experts Group.
VQI	Visual quality index.

VQM	Video Quality Model.
VQMT	Video Quality Measurement Tool.
WXGA	Wide Extended Graphics Array.
Y4M	YUV4MPEG format.
$Y'C_BC_R$	Digital luma and differential blue and red chroma.
YUV	Analog luma and differential blue and red chroma.
Y'V12	8-bit $Y'C_BC_R$ with chroma subsampled to 4:2:0 (12 total bits per pixel).

LIST OF SYMBOLS

Δ	Difference.
∇	Gradient.
μ	Mean.
σ	Standard deviation.
$\sum$	Summation.
C_B	Digital differential blue chroma.
C_R	Digital differential red chroma.
dB	Decibel.
GB	Gigabyte.
kbit	Kilobit.
Mbit	Megabit.
MiB	Mebibyte.
s	Second.
U	Analog differential blue chroma.
V	Analog differential red chroma.
Y′	Luma.

CONTENTS

1. INTRODUCTION

In 2014, 64.4% of all consumer Internet traffic constituted of video (CISCO, 2015). Streaming of video and audio accounted in 2015 for over 70% of North American downstream traffic in the peak evening hours on fixed access networks, increased from a mere 35% in 2010 (SANDVINE, 2015); of that share, the Netflix service accounts for 37.1%, YouTube 17.9%, and Amazon Video 3.1%, while BitTorrent file sharing is down to 5% from 7% in a single year. The popular video hosting website YouTube serves over a billion users[1] — nearly one-third of all the users of the Internet — and achieves quarterly a gross revenue of U$ 17 billion[2]; also noteworthy, more than half of YouTube views come from mobile devices. Netflix, in turn, serves 75 million users and achieves quarterly a gross revenue of 1.6 billion[3]. However, **Internet data rates** are severely *constrained* compared to High Definition Television (HDTV) broadcasting and optical media, so streaming video over the Internet imposes several challenges to *quality of experience* (QoE), the foremost of which being **efficient lossy compression**. For instance, typical bandwidths for broadcasting and optical media for HD[4] content are 18 Mbit/s and 40 Mbit/s, respectively (WAGGONER, 2010), whereas most Web streaming services range from 3 to 8.5 Mbit/s (APPLE, 2014; AARON et al., 2015; PATTERSON, 2012), averaging less than 1% of the raw data rates, and still retain good *visual quality*.

The increasingly popular Web streaming services must not only contend with limited and diverse bandwidth, but also with a myriad of **devices**: desktop computers, mobile computers, tablet computers, "smart" mobile phones, "smart" television sets, console video games and microconsoles. Such devices offer a wide array of **media decoding and display capabilities** that must be appropriately considered to maximize the quality of experience of streaming to them, since most services attempt to reach as many devices as possible in order to expand business. For example, the **H.264 Baseline Profile** standard is currently the most ubiquitous

1 <http://www.youtube.com/yt/press/statistics.html>.
2 <http://uk.businessinsider.com/stats-on-googles-revenues-from-youtube-and-google-play-2015-7>.
3 <http://ir.netflix.com/results.cfm>.
4 High Definition.

video format, providing a practical balance between compression efficiency and complexity, depending simply on ubiquitous embedded hardware decoders and reasonable battery power. Higher-end "HD-ready" devices, however, further support the **High Profile** of H.264, which achieves higher compression for the same visual quality, and recent high-end devices also support H.265 for Ultra-HD content. Even within "HD-ready" devices, resolutions vary from HD (1280×720), through WXGA[5] (1366×768), to Full HD (1920×1080); and simpler devices mainly range from QVGA[6] (320×240), through VGA (640×480), to FWVGA[7] (854×480) or qHD[8] (960×540) (WAGGONER, 2010; PATTERSON, 2012; STATCOUNTER, 2015). Modern Web streaming systems account for reasonable subsets of these display resolutions and decoding formats; considering WXGA devices simply display HD content, FWVGA and qHD are often considered redundant to each other, and Ultra HD is not yet widely supported, a typical version set consists of the five remaining resolutions, each with the appropriate compression format. In order to serve so many resolutions, decoders, and bandwidths, and with the relatively low cost of storage in Web servers, most video-on-demand (VOD) services, such as Netflix and YouTube, produce several optimized versions of their content beforehand and **stream adaptively**. Adaptive streaming is not strictly defined, however, by adaptation for different decoders, but primarily by adaptation to bandwidth fluctuations.

Digital compression is based on mathematical techniques for codifying data in such a way to minimize redundancy, thus increasing the entropy or "randomness" of the codified data. Video is a particularly redundant subject for compression, especially natural video, because visual information tends towards significant redundancy both in the spatial dimensions and the temporal dimension (WAGGONER, 2010). Furthermore, the human visual system does not uniformly perceive losses of spatial and temporal information, so some information are more important than others, enabling lossy compression to discard unnecessary data by considering psychovisual priorities. This is how compression rates of 1:100 for video

5 Wide Extended Graphics Array.
6 Quarter Video Graphics Array.
7 Full Wide Video Graphics Array.
8 Quarter High Definition.

are possible without necessarily degrading subjective opinion. However, it also follows that objective prediction and assessment of the perceptual quality is far from trivial, and simple physical objective metrics such as the bits per pixel or the mean squared error are ineffective (WANG; BOVIK, 2009). **Perceptual objective visual quality assessment (VQA)** has greatly improved in the past decade, but could hardly be said to have achieved an ideal stage and remains an active topic of scientific research.

Among the most relevant perceptual VQA metrics are the many versions of the **Structural Similarity (SSIM)** index. In this work, several of the most efficient SSIM-based metrics, such as the *Multi-Scale Fast SSIM* (CHEN; BOVIK, 2011) and the *Gradient Magnitude Similarity Deviation* (XUE et al., 2014), are decomposed into their component techniques and reassembled in order to measure and understand the contribution of each technique and to develop improvements in quality and efficiency. This work presents the **"Shifted Gradient Similarity" (SG-Sim)** index as a version of SSIM with an improved feature enhancement operation. Furthermore, more efficient spatial pooling filters are proposed, as well: the decomposed 1-D integer Gaussian filter limited to two standard deviations **(2σ Gaussian)**, and the **downsampling Box filter** based on the integral image, which constitutes the **Fast SG-Sim** version.

For convenient and efficient handling of video resources and batch experiments, the SSIM-based metrics studied in this work were implemented in the **Video Quality Assessment in Java (`jVQA`)** toolset. `JVQA` is inspired by the Moscow State University Video Quality Measurement Tool[9] and designed for assembling VQA component techniques into full, experimental metrics. Virtually any publicly available video format is supported by means of the FFmpeg decoders[10] and the AviSynth frameserver[11], and computing efficiency is a primary requirement.

Human testing of image and video quality metrics in laboratory is a very resource-demanding enterprise. For this reason, many research groups publish not only the results for their metrics, but also the image and video datasets employed

9 <http://compression.ru/video/quality_measure/video_measurement_tool_en.html>.
10 <http://ffmpeg.org/general.html#Supported-File-Formats_002c-Codecs-or-Features>.
11 <http://avisynth.nl>.

and the associated subjective metrics, such as the *mean opinion score (MOS)* and the **differential mean opinion score (DMOS)**. Thus, other researchers may independently test their own objective metrics' effectiveness at predicting subjective quality by computing their correlation to the provided MOS or DMOS. In this work, comparative experiments are conducted over SSIM-based metrics in order to determine the contribution and efficiency of each of the interchangeable VQA techniques that constitute SSIM-based metrics, such as the "shifted gradient" feature enhancement and the downsampling Box pooling filter.

This research is further concerned with studying the **behavior** of SSIM-based metrics, such as the amplitude of their responses, the significance of such values, and how these traits change throughout different resolutions. Finally, the metrics are tested on their correlation to the two classic **rate-distortion optimization (RDO)** modes, PSNR-RDO[12] and SSIM-RDO (WANG., S., 2012), against their correlation to the recent *Psychovisual RDO (Psy-RDO)* (WAGGONER, 2010). A new set of reference videos is produced in order to test the metrics not only against "clean" natural video, but also against other types of common content: film grain noise and classic animation, as well as high action and low action. This new dataset is encoded following common recommendations for adaptive streaming for the Web, with versions for each RDO mode. The literature on perceptual VQA stipulates that SSIM-RDO should produce better quality than PSNR-RDO, and Psy-RDO should improve upon SSIM-RDO, so an ideal perceptual VQA metric should reflect this ranking in addition to achieving high correlation to MOS. Thus, a new dataset and methodology for comparing VQA metrics is presented to complement the traditional MOS correlation methodology. Thus, this work presents and compares many state-of-the-art perceptual video quality techniques both for encoding and for assessment, with an emphasis on computing efficiency for low-latency applications.

This dissertation is, hence, structured as follows. Chapter 2 contextualizes the challenges and opportunities for adaptive streaming over HTTP, which typically targets HTML5 decoders, and defines the scope for research, including modern Psy-RDO in encoders such as `x264` and `x265`. Chapter 3 describes in depth the VQA

12 Peak signal-to-noise ratio.

techniques involved in SSIM-based metrics and presents the shifted gradient feature enhancement for achieving higher quality, and the 2σ Gaussian and downsampling Box filters for achieving higher computation efficiency. Chapter 4 presents the `jVQA` toolset for comparing SSIM-based metrics. Chapter 5 reports the experimental methodologies and results, with commentaries on VQA techniques' contributions and behaviors. Finally, Chapter 6 concludes the work, clarifies its limitations, and recommends future investigations.

2. *PERCEPTUAL QUALITY IN WEB VIDEO CODING*

In this chapter, a contemporary picture of video encoding for the Web and its technical challenges is described. Currently deployed devices impose very particular constraints to achieving visual quality as perceived by the human visual system (HVS). The broad diversity of decoders, displays, and network connections demand scalable, adaptive and inexpensive technologies (ZAMBELLI, 2009). The most effective solution for such scenario has become streaming adaptively, both in regards to network bandwidth and to decoding capabilities such as display resolution and coding complexity on common HTTP[13] platforms. HTML5[14] decoders are also becoming ubiquitous, gravitating towards the H.264 standard (WAGGONER, 2010). Web bandwidths, however, are considerably lower than those in broadcasting and optical media, requiring highly effective lossy compression techniques. All these parameters and constraints must be balanced for maximizing the quality of experience in streaming systems.

2.1. Adaptive streaming over HTTP

Video streaming services have traditionally employed specialized communication protocols, such as RTSP, RTP and RTCP, that maintain open sessions with frequent data exchanges (ZAMBELLI, 2009; WAGGONER, 2010; LEVKOV, 2011). Recently, however, streaming services are shifting to HTTP, taking advantage of ubiquitous infra-structure such as proxies, caches, firewalls, network address translation (NAT) and content-delivery networks (CDNs), which require no further application-specific configuration, are available at commodity costs and require less system resources, being a stateless protocol (ZAMBELLI, 2009; SODAGAR, 2011).

Initially, streaming over HTTP was accomplished by simple progressive downloads that could be played during data transfer. This model, however, soon proved inefficient due to the fluctuations in available bandwith, typical of consumer

13 Hypertext Transport Protocol.
14 Hypertext Markup Language.

devices, especially mobile devices, resulting in buffer underruns and interrupted playback; also, unnecessary data consumption occurs when playback is paused by the user (ZAMBELLI, 2009; WAGGONER, 2010; MARQUES; BETTENCOURT; FALCÃO, 2012). Adaptive HTTP streaming proposes to mitigate these problems by segmenting the content into few-seconds-long streams, then encoding several versions of varying data rates and keeping the buffer always full by switching to lower-data-rate versions when bandwidth becomes reduced (STOCKHAMMER, 2011; LEDERER, 2012; PATTERSON, 2012; APPLE, 2014). Depending on the media container format, each segment in a version may be a separate file or the same file with downloads by bit ranges. However, while traditional constant-rate streaming was performed with key frame intervals of 10 seconds or more, adaptive streaming uses intervals of 2 to 4 seconds for fast switching (ZAMBELLI, 2009; LEVKOV, 2010 & 2011; STOCKHAMMER, 2011), and this shorter interval reduces *compression efficiency*, as discussed below in Section 2.3. Thus, segmented content allows for fast version switching, increasing *bandwidth efficiency* and *quality of experience*, but presents a greater compression challenge to the encoder to preserve visual quality.

The reduced cost of HTTP infra-structure is especially appropriate within the present context of a myriad of streaming client devices: desktop computers, mobile computers, tablet computers, "smart" mobile phones, "smart" television sets, console video games and microconsoles. These devices offer wide variations of **display resolutions, decoding capabilities** and **Internet bandwidths**, requiring ample **content versioning** in streaming services and increasing infrastructure cost. Such scenario demands a visual quality index (VQI) that is **scalable** for objective, automated quality control over multiple resolutions. To illustrate, tables 2.1-2.3 reproduce three complete arrays implemented by Apple (2014), Netflix (AARON et al., 2015) and Youtube[15,16].

15 <http://support.google.com/youtube/answer/1722171>.
16 <http://en.wikipedia.org/wiki/YouTube#Quality_and_formats>.

Table 2.1 — Recommended array of video data rates and resolutions for Apple HTTP Live Streaming Media.

Data rate (kbit/s)	Coding resolution	Pixel total	Frame rate	Bits/ pixel	Keyframe interval (s)	H.264 Profile
200	416×234	97,344	12	0.18	3	Baseline
400	480×270	129,600	15	0.21	3	Baseline
600	640×360	230,400	29.97	0.09	3	Baseline
1200	640×360	230,400	29.97	0.17	3	Baseline
3500	960×540	518,400	29.97	0.23	3	Main
5000	1280×720	921,600	29.97	0.18	3	Main
6500	1280×720	921,600	29.97	0.24	3	Main
8500	1920×1080	2,073,600	29.97	0.14	3	High

Table 2.2 — Typical array of video data rates and resolutions for balanced and high quality streaming in Netflix, based on MPEG-DASH.

Data rate (kbit/s)	Coding resolution	Pixel total	Frame rate	Bits/pixel	Pixel aspect ratio	Display resolution
235	320×240	76,800	23.976	0.13	4:3	426×240
375	384×288	110,592	23.976	0.14	4:3	512×288
560	512×384	196,608	23.976	0.13	4:3	682×384
750	512×384	196,608	23.976	0.16	4:3	682×384
1050	640×480	307,200	23.976	0.14	4:3	854×480
1750	720×480	345,600	23.976	0.21	32:27	854×480
2350	1280×720	921,600	23.976	0.11	1:1	1280×720
3000	1280×720	921,600	23.976	0.14	1:1	1280×720
4300	1920×1080	2,073,600	23.976	0.09	1:1	1920×1080
5800	1920×1080	2,073,600	23.976	0.12	1:1	1920×1080

Table 2.3 — Typical array of video data rates and resolutions streamed by Youtube, based on MPEG-DASH.

Data rate (kbit/s)	Coding resolution	Pixel total	Frame rate	Bits/pixel	H.264 Profile
300	426×240	102,240	30	0.10	Main
400	640×360	230,400	30	0.06	Main
1000	854×480	518,400	30	0.06	Main
1500	1280×720	921,600	30	0.05	Main
3000	1920×1080	2,073,600	30	0.05	High

Adaptive streaming over HTTP has consolidated its market presence and is used by major streaming services such as Youtube, Amazon and Netflix[17]. It is implemented in several proprietary streaming platforms, such as Apple HTTP Live Streaming, Microsoft Smooth Streaming and Adobe HTTP Dynamic Streaming. However, each of these implementations are said to be 80% similar and 100% incompatible (LAW, 2012), with no common whole specification of manifest file format, media container and segmentation formats, and media players. In order to standardize such methods, the MPEG developed the Dynamic Adaptive Streaming over HTTP (DASH) specification (SODAGAR, 2011), which ignores specific client and playback implementations, focusing on the methods for indexing and formatting the content resources. The reference implementations of MPEG-DASH are the `libdash` open source library (LEDERER, 2013) and the Dash.js JavaScript client[18]. DASH is implemented in relevant software such as most popular Web browsers, Microsoft Azure Media Services, Adobe Primetime, Akamai CDN, VLC Media Player, Helix Universal Server, GPAC framework and GStreamer. However successful or not DASH comes to be, the fact remains that adaptive streaming over HTTP platforms in general are widely adopted and under development, and demand an effective, efficient and scalable objective visual quality index.

17 <http://blog.eltrovemo.com/1218/mpeg-dash-ecosystem-status>.
18 <http://github.com/Dash-Industry-Forum/dash.js>.

2.2. Web video decoders

Video playback in Web clients has traditionally been carried out by proprietary plug-ins such as Adobe Flash and Microsoft Silverlight (WAGGONER, 2010; SHIKARI, 2010a). However, these have been often and strongly criticized for security and stability concerns, aggravated by their closed-source nature, while Web video has become ubiquitous and considerably relevant. To address such concerns, Web standards organizations have developed the *HTML5* standard, which includes video playback and animations[19,20]. These technologies are now integrated within every relevant Web browser and are gradually substituting the traditional plug-ins. Apple, which provides one of the most relevant streaming platforms and mobile product lines, refuses to support plug-in-based players, claiming security and stability flaws (JOBS, 2010). Other major providers such as Mozilla, Google, Netflix and Opera have likewise been pushing towards substituting proprietary plug-ins by HTML5. Microsoft has also recently announced they plan to discontinue Silverlight, shifting to HTML5 (PARK; WATSON, 2013). HTML5 players may be formatted with Cascading Style Sheets (CSS) and programmed with JavaScript (JS), which are also ubiquitous in Web clients. Indeed, there are several such open-source players available[21], including DASH players. The main obstacle remaining for wider HTML5 adoption is the lack of more sophisticated technology standards such as cryptography and digital rights management (DRM), which are under development (PARK; WATSON, 2013).

As mentioned in the previous section, the most relevant Web video format today is the ITU-T standard **H.264**, also standardized by ISO/IEC as MPEG **Advanced Video Coding (AVC)** (Waggoner, 2010). Its Baseline Profile enjoys *ubiquitous* support in contemporary video-capable devices, often by embedded hardware decoders. HD-ready devices also support the High Profile, which achieves greater compression and quality while demanding more system resources. Although the use of H.264 requires payment of royalties, these are forfeit for free services

19 <http://www.w3.org/TR/html5>.
20 <http://www.w3.org/TR/media-source>.
21 <http://html5video.org/wiki/HTML5_Player_Comparison>.

(WAGGONER, 2010), and Cisco recently made freely available their OpenH264 software implementation of the Baseline Profile, paying the licensing costs themselves for all downloads of the binary packages[22], and this has been adopted by Mozilla. H.264 also enjoys one of the most efficient and highest-quality software encoders, **`x264`**, which is also free and open-source (FOSS), although it may be subject to payed licenses depending on the specific application (MERRITT, 2006; WAGGONER, 2010; VATOLIN, 2012).

Other relevant formats for HTML5 include the FOSS Theora, VP8 and VP9, advocated by Google, Mozilla and Opera[23]. Theora achieves lower quality than H.264, whereas VP8 achieves comparable quality, between Baseline and High Profile, and VP9 exceeds the quality achieved by H.264, competing with the new H.265 format (also standardized as MPEG High Efficiency Video Coding – HEVC; WAGGONER, 2010), which is targeted at Ultra-HD content. H.265 also enjoys an efficient FOSS encoder, `x265` (MULTICOREWARE, 2015). None of these formats, however, enjoy wide native support, or even wide software support, particularly in systems by Apple and Microsoft. Considering this scenario, this work will focus on H.264: High Profile for HD, and Baseline Profile for lower resolutions.

2.3. Web video quality

Video represents a considerable amount of information, which impose several technical constraints that affect visual quality. Most video applications today are digital, where quality becomes an especially complex subject, involving spatial sampling of colors, temporal sampling, sensor motion blur, lossy compression and particularities of the human visual system (HVS) such as luminance masking (WAGGONER, 2010). Furthermore, Web streaming is achieving popularity comparable to television's, and is likely to exceed it in the next decades (CISCO, 2015), but compared to broadcast and optical media, suffers yet from **severely constrained data rates**. For instance, DVD[24] media typically offers 720×480 (Standard

22 <http://blogs.cisco.com/collaboration/open-source-h-264-removes-barriers-webrtc>.
23 <http://www.webmproject.org/about/faq>.
24 Digital Video Disc.

Definition, SD) resolution at an average 8 Mbit/s (maximum 9.8 Mbit/s), and Blu-ray typically offers 1920×1080 (Full High Definition, FHD) resolution at an average 20 Mbit/s (maximum 40 Mbit/s) (WAGGONER, 2010), whereas popular Web services such as Youtube and Netflix stream FHD constrained at a range of mere 3-6 Mbit/s (see Tables 2.1-2.3). Although it has been stated that the high data rates in optical media are excessive in order to accommodate low-quality encoding implementations (SHIKARI, 2008c), as well as virtually any kind of content, typical data rates for Web streaming are significantly low and prone to **blurring** and **artifacts** such as banding, blocking and ringing, requiring efficient coding implementations and **visual quality assessment (VQA)** algorithms to achieve the best possible perceptual quality.

Digitizing video from the analog world first requires sampling the continuous physical signal into discrete information: this is the *resolution* of the digital video. Each information sample is a color point, a pixel (picture element), that must be quantized, typically within 24 bits, which allows over 16 million colors (whereas humans can discriminate 10 million) (WAGGONER, 2010). Quantization must refer to a *color space*, which defines how colors are mathematically recorded. The most common color space for displays is RGB, which encodes the red, green and blue light signals in three separate 8-bit channels. However, the most common color space for encoding, storage, streaming and broadcast is the planar scheme $Y'C_BC_R$, where Y′ stands for the luma signal, which contains most of the relevant information, C_B stands for blue chroma and C_R stands for red chroma. $Y'C_BC_R$ recognizes that the greyscale Y′ plane condenses most of the visual information, allowing the chroma planes to encode only differential information, with green being implicit as the widest range of the spectrum that is visible to humans. $Y'C_BC_R$ is most commonly encoded with *4:2:0 chroma subsampling*, by which the chroma resolutions are reduced so each chroma pixel corresponds to a luma region of 2×2 pixels. This is the first important implementation strategy to reduce the large amount of video data, achieving a 50% reduction, from 24 to 12 bits per pixel, and has been used for decades in analog equipment predating digital video encoding (though not with bits). 4:2:0 $Y'C_BC_R$ is popularly called **Y′V12**, as a confusion with the distinct analog colorspace YUV (WAGGONER, 2010); in fact, files containing such raw video use the extension

"yuv".

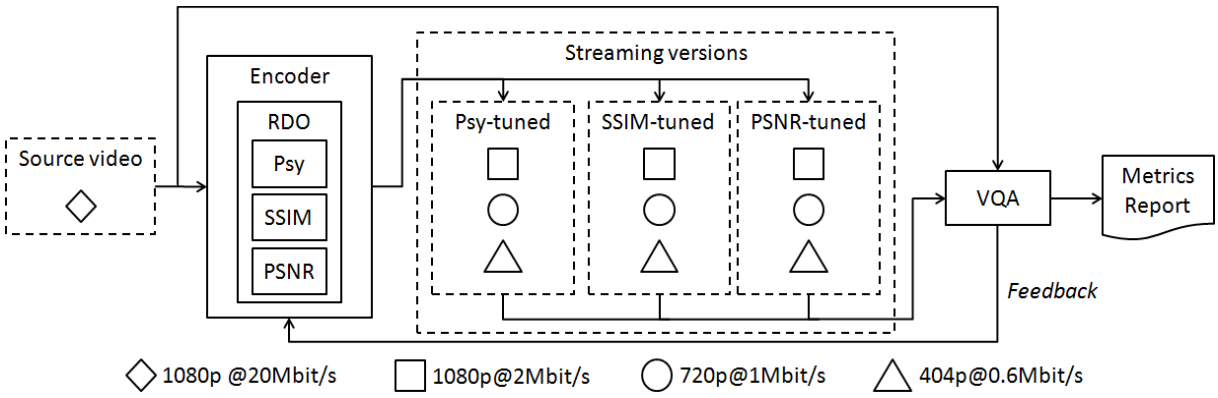
Even with Y'V12, 720×480 at a typical 24 frames per second (Hz) results in 99.5 Mbit/s, 12 times larger than the 8 Mbit/s of DVD. This is achieved by **lossy compression**, which not only applies lossless compression algorithms to treat redundant data and increase entropy, but first discards less perceptually relevant visual information. Most digital video coding is based on the discrete cosine transform (DCT), a specialization of the Fourier transform, which transforms the spatial information into the frequency domain and allows for more efficient compression and a more specific level of quantization. In fact, the terms quantization and *quantizer* in video encoding contexts most often refers to the precision of the DCT, or the amount of information discarded. This quantization is the cause for the common blurring and artifacts mentioned above; the lower the data rate, the higher the blurring and artifacting. Optical media formats such as DVD and Blu-ray, as well as digital TV, offer high enough data rates to rarely incur in loss of quality due to encoding (transmission signal quality is a much higher concern), but this is not the case in Web streaming. SD is typically streamed at around 1 Mbit/s (see Tables 2.1-2.3), which is a mere **1%** of the uncompressed rate, whereas DVD uses 8 times as much. Even considering the key frame interval for DVD is 0.5 second (for speed of random seeking within the stream), and 1 second for Blu-Ray, the difference is considerable enough to demand high quality encoding.

2.4. Perceptual quality metrics in encoders

Lossy encoders must perform data rate control to preserve the most perceptual quality possible. Reducing the rate produces a distortion, which must be measured for **rate-distortion optimization (RDO)** (WANG, S. et al., 2012, 2013). Such decision algorithms require visual quality indexes. The most trivial measure for video quality is the amount of bits per second. However, this is the least informative. Content with a high degree of movement (temporal complexity) or high frequency details (spatial complexity) require more bits to achieve similar quality with simpler content. Pixel-by-pixel comparison by the mean squared error (MSE) is, by far, more effective.

However, MSE and its derivative **peak signal-to-noise ratio (PSNR)** still fail to account for human *perceptual bias* of the distortions (WANG, Z.; Bovik, 2009). Many perceptual visual quality metrics have been proposed (LIN; KUO, 2011), one of the most relevant being the **structural similarity (SSIM)** index (Wang, Z. et al., 2004). Perceptual metrics are evaluated by correlation with subjective scores from laboratory tests with human subjects. SSIM has been extended and adapted by several techniques, such as in Multi-Scale SSIM (WANG, Z.; SIMONCELLI; BOVIK, 2003), Gradient SSIM (CHEN; YANG; XIE, 2006), Gradient-weighted SSIM (LI; BOVIK, 2010a & 2010b), Fast SSIM (CHEN; BOVIK, 2011) and Gradient Magnitude Similarity Deviation (GMSD) (XUE et al., 2014), in order to increase subjective correlation; these will be the primary subjects of this work's investigation, elaborated in Chapter 3, implemented in Chapter 4 and tested in Chapter 5.

Fig. 2.1 — Example of low-data-rate streaming versions produced by different encoder RDO modes and compared to the source by VQA metrics.



Source: MONTEIRO et al., 2015.

An effective perceptual metric is important for RDO during encoding and for evaluating the product of the encoding. This also allows an encoding service to manage and report the quality of produced content. Such workflow is illustrated by Fig. 2.1. However, SSIM values are not consistent throughout different resolutions; in this regard, SSIM is not scalable. In an adaptive streaming system, it is important to maintain consistent quality among resolutions; reduced resolutions are naturally blurred as a result of the low-pass filter required for adequate downsampling, but artifacting should be proportionally consistent to the higher-sampled version.

Another application-critical constraint is that the metrics used in RDO do not significantly increase **encoding time**, as to not disproportionally increase the processing latency in encoding systems, so the quality index must be computationally efficient. For this reason, some relevant SSIM-based or related metrics are out of the scope of this investigation, such as those that account for temporal information, such as in Wang, Lu and Bovik (2004), Video Quality Model (VQM) (PINSON; WOLF, 2004), Spatio-Temporal Video SSIM (MOORTHY; BOVIK, 2009a), Motion-based Video Integrity Evaluation (MOVIE) (SESHADRINATHAN; BOVIK, 2009), Motion-Compensated SSIM (MOORTHY; BOVIK, 2010), and SSIMplus (REHMAN; ZENG; WANG, 2015); as well as more complex algorithms, such as Complex Wavelet SSIM (WANG; SIMONCELLI, 2005), PSNR-HVS (EGIAZARIAN et al., 2006; PONOMARENKO et al., 2009), Fixation SSIM (MOORTHY; BOVIK, 2009b), Information-Weighted SSIM (WANG; LI, 2011), and Feature Similarity (ZHANG et al., 2011). **Fast SSIM** and **GMSD** are the fastest SSIM-based metrics, so the primary objective of this work is to develop an SSIM-based index of *comparable speed* and *higher perceptual quality*.

A particularly effective implementation of perceptual RDO, that improves upon SSIM-based RDO, is in the `x264` and `x265` encoders, and is called **Psychovisual RDO (Psy-RDO)** (SHIKARI, 2008a; SHIKARI, 2009a; WAGGONER, 2010; PATTERSON, 2012; AVIDEMUX, 2012; MEWIKI, 2012). Psy-RDO attempts to preserve the visual energy in each DCT block, whereas SSIM and PSNR-based RDO tend towards blurring. Encoding with Psy-RDO is considered of higher perceptual quality but lowers both SSIM and PSNR for the produced content. A more effective perceptual quality index is expected to correlate better with Psy-RDO. Achieving **higher correlation** with Psy-RDO with the new SSIM-based index is the secondary objective of this work.

2.5. Closing remarks

In this chapter, the fundamental goals and constraints for quality of experience in video coding for the Web are layed out. Content must be available at several data rates for optimal bandwidth efficiency, as well as several display resolutions and decoding complexities. Adaptive streaming over HTTP achieves these goals with inexpensive equipment, but reduces compression efficiency. HTML5 is becoming the standard decoding platform in the Web, and mostly uses the H.264 video format, although royalty-free formats are available on some platforms. The `x264` encoder has implemented effective psychovisual optimizations which have not yet been reflected by common video quality metrics, which indicates that the existing metrics may be further improved, for both evaluating the encoded versions as well as assisting the encoders in rate-distortion coding decisions. Such a new algorithm must achieve increased correlation to subjective perceptual quality without increasing encoding latency beyond that of common SSIM data rate control implementations.

3. *IMPROVING VIDEO QUALITY ASSESSMENT TECHNIQUES WITH A SHIFTED GRADIENT*

The **structural similarity (SSIM)** index (WANG, Z. et al., 2004) is a relevant image quality assessment (IQA) metric that has been modified and extended by several techniques (XUE et al., 2014), each aiming to improve the metric's correlation to perceptual quality as represented by the **distortion mean opinion score (DMOS)**. Investigating such techniques in several combinations is paramount to understand their contributions and propose further improvements. SSIM has become an important metric for VQA during and after encoding; popular modern video encoders such as `x264` (Merritt, 2006) and `libvpx`[25] compute SSIM in their **rate-distortion optimization (RDO)** controls and also output the produced version's index. VQA tools such as the Moscow State University Video Quality Measurement Tool (MSU VQMT)[26] compute several versions of SSIM.

As discussed in Section 2.4, there are SSIM-based metrics that account for the *temporal dimension* in video information besides the common frame-by-frame approach. However, such techniques are more complex, increasing computing latency and demanding more resources, and for these reasons are not as widely implemented, especially in streaming and encoders and comparisons of those (VATOLIN et al., 2012; DAEDE; MOFFITT, 2015), and are not investigated in this work.

SSIM-based indexes are algorithms consisting of six components, each of which being implemented by particular techniques:

1. spatial feature enhancement;
2. spatial feature pooling;
3. the similarity index that produces the quality map;
4. quality map pooling;
5. input resolution scaling (optional);
6. luma similarity index (optional).

25 <http://www.webmproject.org/about/faq>.
26 <http://compression.ru/video/quality_measure/video_measurement_tool_en.html>.

The primary component of an SSIM index is the spatial feature enhancement method for the compared images, identified as signal x and signal y, which may be implemented by techniques such as the covariance and/or gradient of the pixel color levels, discussed in Section 3.1. Such features emphasize spatial relations between neighboring pixels, accounting for contrast and edges and are especially sensitive to blurring, blocking, and ringing artifacts, typical of low-data-rate lossy encoding. Enhancement is performed on each channel or plane of the image's color space. For the most common inputs, i.e., JPEG or MPEG-based images, which are coded in the Y'V12 ($Y'C_BC_R$ 4:2:0) planar color space, it is commonly sufficient to compare only the luma (Y') plane.

After enhancement, most versions of SSIM perform local spatial pooling in order to consolidate each pixel's features with those of their neighbors, improving comparison coherency in a manner consistent with the human visual system (HVS). Feature pooling is achieved through a spatial filter such as the Box filter or the Gaussian filter, detailed in Section 3.2, and contributes to quality but increases the computation cost.

The pooled features are then compared in the quadratic division to produce a similarity index (1). Because variance-based feature results range from -1 to 1 multiplied by the dynamic range, the comparison expression may be nullified on results of 0, distorting the behavior of the index. In most versions of SSIM, this distortion is treated by adding a stabilizer constant C to both the numerator and the denominator. This is commonly proposed even when working only with gradients, even though the results of which are limited to the original dynamic range instead. Such stabilization also introduces distortion, so alternatives have been proposed, described in 3.3. After each pixel is compared by these three components, a quality map is produced, with a discrete SSIM index at each position. All results are then consolidated to an overall index, usually by the mean of results, though other methods such as the standard deviation have also been proposed, as commented in Section 3.4.

$$SSIM(x,y) = \frac{2\sigma_{xy} + C_1}{\sigma_x^2 + \sigma_y^2 + C_1} \quad (1)$$

The four core IQA techniques in SSIM are complemented by scaling of the input and also by comparing the images' luma signals. Most versions perform dyadic downsampling before computing SSIM according to the HVS resolution for the display dimension and distance from the observer, and there are also effective multi-scaled approaches (WANG; SIMONCELLI; BOVIK, 2003; CHEN; BOVIK, 2011). Depending on the application, simply comparing the luma levels in addition to the images' structures may also be relevant: without a luma comparison, processed images that are darkened, lightened or inverted to negatives may not reflect these distortions on the index, compromising its effectiveness. These additional techniques are discussed in Sections 3.5 and 3.6.

3.1. Enhancing spatial features

Wang and Bovik's original universal IQA index (2002), later developed into SSIM, proposes spatial features enhancement by computing the covariance of the color levels in each of the compared images. This feature enhancement represents the image's contrast and structure and requires either spatial pooling of the neighboring pixels, or the mean color level for the whole frame for each enhanced yet unfiltered pixel. Let x be the 1-dimensional vector signal of the original image, y be the signal of the processed image, and C be a small stabilizing constant; σ_x^2 is, thus, the variance of the color levels of x, representing the contrast, whereas σ_{xy} is the covariance between the color levels of x and y, representing the structure. Then, the similarity index is defined in (1).

Chen, Yang and Xie (2006) propose to first compute the gradient magnitudes of the images being compared by the 3×3 Sobel operators (Fig. 3.1), then compute the regular covariance SSIM of these gradients. This **Gradient SSIM** was found slightly more effective than simple covariance SSIM. Li and Bovik (2010) later published the **3-SSIM** and **4-SSIM** indexes, which may in turn be described as the opposite

approach regarding the combination of covariance and gradients: the covariance SSIM is computed normally, then the spatial results are weighted according to the Sobel gradients of the input images. This also improved DMOS correlation. Both versions are more demanding in computation than the original SSIM, although it is clear that enhancing by gradients is effective to improve SSIM.

Fig. 3.1 — Horizontal and vertical Sobel operators for gradient magnitude.

1	2	1
0	0	0
-1	-2	-1

1	0	-1
2	0	-2
1	0	-1

Source: the Author.

The gradient magnitude is a Euclidean quadratic distance defined in (2), where ∇i is the response to the horizontal spatial gradient operator and ∇j is the response to the vertical operator for each pixel n. Gradient SSIM proposes to approximate this expression by (3), a common simplification.

$$\nabla_n = \sqrt{\nabla i_n^2 + \nabla j_n^2} \quad (2)$$

$$\nabla_n = |\nabla i_n| + |\nabla j_n| \quad (3)$$

Chen and Bovik (2011) propose **Fast SSIM** as a less complex alternative, specifically for video on mobile devices. This version of the index compares only the gradients of the images, which are produced by the faster 2×2 Roberts operators (Fig. 3.2), ignoring any consideration of covariance.

Fig. 3.2 — Roberts operators for gradient magnitude.

1	0
0	-1

0	1
-1	0

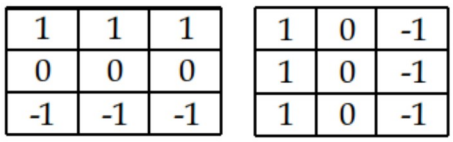
Source: the Author.

The gradient magnitude in Fast SSIM is approximated by (4), which produces results closer to (2) than (3) does. Fast SSIM is shown to be over 2.5 times faster to compute while within 99.7% of the prediction of quality of covariance SSIM.

$$\nabla_n = \max(|\nabla i_n|, |\nabla j_n|) + {}^1\!/_4 \min(|\nabla i_n|, |\nabla j_n|) \quad (4)$$

Building upon Fast SSIM, Xue et al. (2014) propose the **Gradient Magnitude Similarity Deviation (GMSD)** index, based on gradients per 3×3 Prewitt operators (a compromise between Sobel and Roberts, Fig. 3.3) and unfiltered, instead of locally pooled. This index is shown to outperform all other versions of SSIM in both computing speed and subjective quality prediction. GMSD indicates that an index based only on gradients, without covariance, may give better results without spatial pooling. However, this is an oversimplification: as discussed in Section 3.3, further below, the stabilization component is a greater factor for improving the index.

Fig. 3.3 — Horizontal and vertical Prewitt operators for gradient magnitude.



Source: the Author.

3.1.1. The Shifted Gradient Similarity index

Experimentation with stabilization techniques, as described in 3.3, produced a spatial structure feature enhancement original to this work: the "**shifted gradient**" (MONTEIRO et al., 2015). This is a mathematically trivial adjustment to the approximate gradient magnitude of Fast SSIM, adding 1, as per (5). However, because the dynamic range of gradient magnitude is similar to the dynamic range of the processed color levels, such as 0 to 255 (although the magnitude might exceed this to 320, about 25% greater), without negative values, this changes the dynamic range to 1-256, allowing to discard any further division stabilization, thus reducing distortion. This original technique, together with the right combination of the other components, outperforms GMSD in both computing speed and prediction of perceptual quality, as shown in Chapter 5.

$$\nabla_n = \max(|\nabla i_n|, |\nabla j_n|) + {}^1\!/_4 \min(|\nabla i_n|, |\nabla j_n|) + 1 \quad (5)$$

Let ∇S be the gradient magnitude for the source image, and ∇V its counterpart for the low-data-rate version, as defined in (5). Spatial pooling, represented by μ for N pixels, is defined in (6) and (7) (see Section 3.2). The **Shifted Gradient Similarity index (SG-Sim)** is defined in (8), and substitutes (1), above. SG-Sim requires no division stabilization, as defined in Section 3.3, but its DMOS correlation may benefit from arithmetic stabilization, as investigated in Chapter 5.

$$\mu_{\nabla S} = \frac{1}{N}\sum_{i=1}^{N} \nabla S_i \quad (6)$$

$$\mu_{\nabla SV} = \frac{1}{N}\sum_{i=1}^{N} (\nabla S_i \, \nabla V_i) \quad (7)$$

$$SG\text{-}Sim(S,V) = \frac{2\mu_{\nabla SV}}{\mu_{\nabla S}^2 + \mu_{\nabla V}^2} \quad (8)$$

Although each of the gradient operators Roberts, Prewitt and Sobel may appear to produce very similar information, experiments reported in Chapter 5 reveal significant differences in visual quality assessment. Surprisingly, the Prewitt operator is found to perform significantly better than the more precise Sobel, which in turn performs significantly better than Roberts, as expected.

3.2. Pooling spatial features

The universal IQA index proposed by Wang and Bovik that would become SSIM employed an **8×8 Box filter** for spatial pooling of variances and covariance. This simple mean of each pixel and its neighbors is effective, yet produces blocking artifacts in the quality map. In order to improve this result, SSIM instead proposes a **Gaussian filter**, which is defined by greater weights at the center that decrease by the normal distribution in a circular-symmetric pattern, thus softening the edges of the quality map and producing a more natural image, with improved isotropy. The Gaussian filter in SSIM is based on a standard deviation σ = 1.5 and, as usual, the radius includes 3σ, thus an 11×11 filter, illustrated by Fig. 3.4. Naturally, convolving

this filter with the image is slower to compute than the 8×8 Box filter, which requires only 53% as many operations and allows more effective computational optimizations.

Fig. 3.4 — 11×11 Gaussian filter for $\sigma = 1.5$ represented by integers to facilitate visualization.

0	1	5	16	31	**39**	31	16	5	1	0
1	8	**39**	117	229	286	229	117	**39**	8	1
5	**39**	183	556	1084	1353	1084	556	183	**39**	5
16	117	556	1690	**3292**	4111	**3292**	1690	556	117	16
31	229	1084	**3292**	6412	**8007**	6412	**3292**	1084	229	31
39	286	1353	4111	**8007**	10000	**8007**	4111	1353	286	**39**
31	229	1084	**3292**	6412	**8007**	6412	**3292**	1084	229	31
16	117	556	1690	**3292**	4111	**3292**	1690	556	117	16
5	**39**	183	556	1084	1353	1084	556	183	39	5
1	8	**39**	117	229	286	229	117	**39**	8	1
0	1	5	16	31	**39**	31	16	5	1	0

The 7×7 core is emphasized to show the greater density at the center. The emphasized coefficients illustrate the circular symmetry. **Source**: the Author.

Fast SSIM identifies the spatial pooling of image features as the greatest computational cost in SSIM-based indexes (CHEN; BOVIK, 2011). Because digital video streams at least 24 frames per second, efficiency in computations is a primary concern. To minimize the impact of convolutions, Fast SSIM reduces the filter to 8×8, while attempting to retain an approximation of the Gaussian weights substituting precise floating-point coefficients by proportional rounded integer coefficients, which must be normalized at each pixel by dividing by their total; see Fig. 3.5.

Fig. 3.5 — Fast SSIM integer filter.

0	0	0	1	1	0	0	0
0	0	1	**2**	**2**	1	0	0
0	1	**2**	4	4	**2**	1	0
1	**2**	4	**8**	**8**	4	**2**	1
1	**2**	4	**8**	**8**	4	**2**	1
0	1	**2**	4	4	**2**	1	0
0	0	1	**2**	**2**	1	0	0
0	0	0	1	1	0	0	0

The greyed out coefficients are null, therefore wasted. **Source**: the Author.

SSIM-based indexes may instead ignore spatial feature pooling altogether. In this case, the quality map is produced pixel by pixel with no consideration for

neighboring pixels. GMSD, for instance, computes no feature pooling. This technique could potentially give more precise results, especially in the case of high definition content, because Box and Gaussian filters are, in fact, blurring operators. Such potential notwithstanding, actual DMOS prediction from most **unfiltered** SSIM versions (except GMSD) is considerably poor, as discussed in Chapter 5. Unfiltered SSIM is, however, 876% faster to compute.

3.2.1. Optimizing the Gaussian filter

The coefficients chosen for Fast SSIM evidently are significantly imprecise and 37.5% null, wasting operations and/or information. Because a Gaussian filter is simply a *bidimensional normal distribution* (GONZALEZ; WOODS, 2007), 3σ includes 99.7% of all values and 2σ includes 95.5%, as illustrated by Figs. 3.8 and 3.9, so we propose a more effective approximation by trimming the 3.3σ filter of 11×11 to an exact **2σ filter of 7×7**. To obtain the integer approximation, all coefficients are divided by the smallest coefficient, as illustrated in Fig. 3.6. With $\sigma = 1.5$, the 7×7 filter includes 96.6% of the weights of the 11×11 filter, using merely 40.5% as many coefficients. Further, a 5×5 filter includes 1.3σ and 86% of the weights requiring merely 20.7% of the coefficients. In this case, simply dividing by the lesser coefficient may distort proportions, so the coefficients may be adjusted so that the centroid is 10 or greater (Fig. 3.7).

Fig. 3.6 — 7×7 Gaussian integer filter.

1	3	6	8	6	3	1
3	9	**18**	23	**18**	9	3
6	**18**	36	**45**	36	**18**	6
8	23	**45**	56	**45**	23	8
6	**18**	36	**45**	36	**18**	6
3	9	**18**	23	**18**	9	3
1	3	6	8	6	3	1

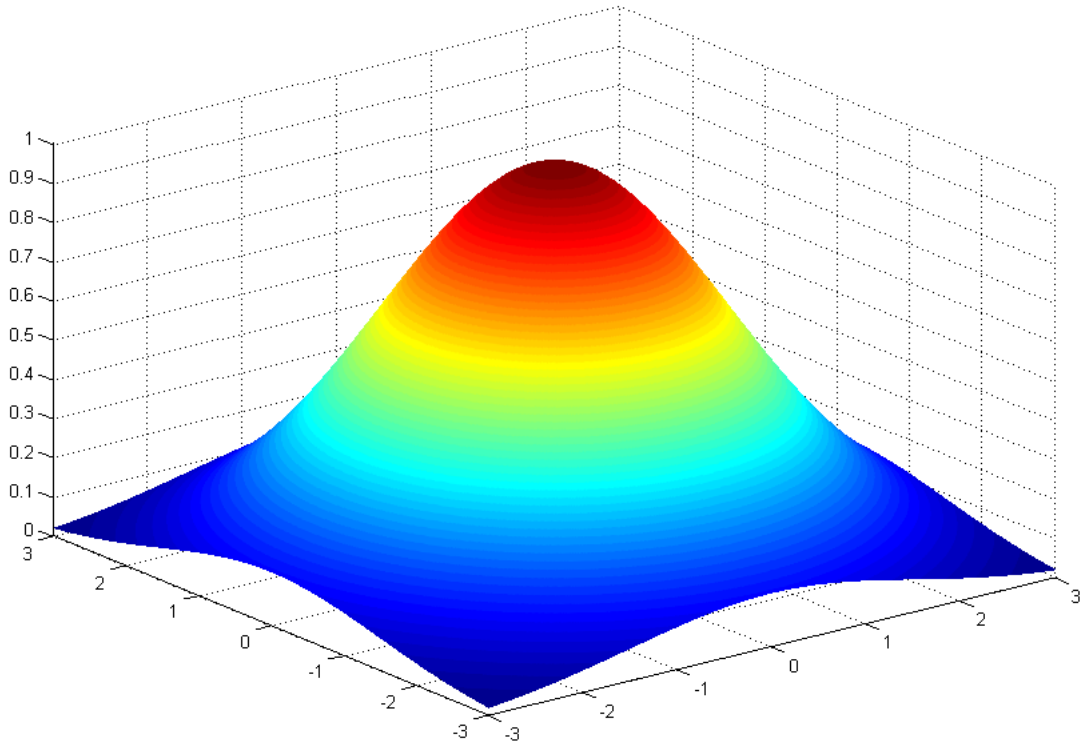
Source: the Author.

Fig. 3.7 — 5×5 Gaussian integer filter.

2	**4**	5	**4**	2
4	8	**10**	8	**4**
5	**10**	12	**10**	5
4	8	**10**	8	**4**
2	**4**	5	**4**	2

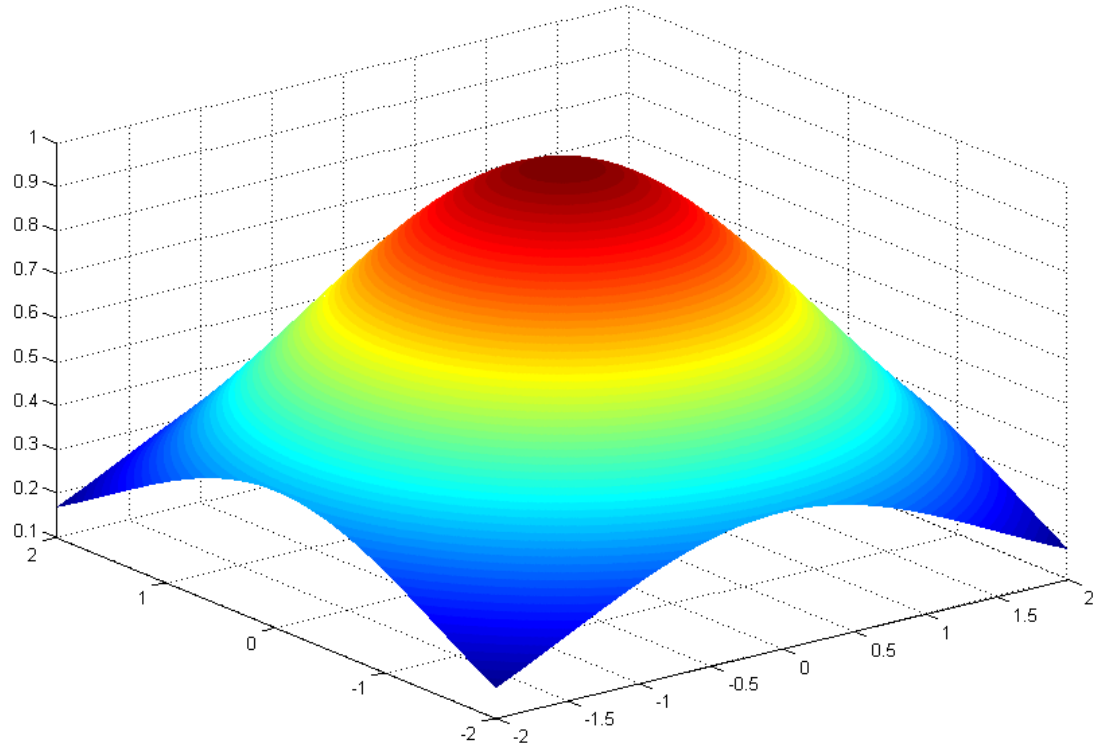
Source: the Author.

Fig. 3.8 — 3-D representation of the Gaussian filter of σ = 1.5 and radius = 3σ.



Source: the Author.

Fig. 3.9 — 3-D representation of the Gaussian filter of σ = 1.5 and radius = 2σ.



Source: the Author.

Fig. 3.10 — 1-D Gaussian integer filter of length 7.

2	6	12	15	12	6	2

Source: the Author.

Fig. 3.11 — 1-D Gaussian integer filter of length 5.

4	8	10	8	4

Source: the Author.

The bidimensional Gaussian filter is also decomposable into two unidimensional filters perpendicular to each other (Figs. 3.10 And 3.11), which require yet again less convolution operations, improving the computing speed of the 11x11 filter by 36%. Preliminary experiments evidenced that, due to the spatial coherency of natural images, the index responses to the approximations in 7×7 and 5×5 are over 99% similar to the response from the full 11×11 2-D filter, while computing respectively 68% and 89% faster (amounting to 24% and 30% faster than the 11×11 1-D filter pair).

3.2.2. Improving efficiency with a downsampling filter

Because VQA applications must balance accuracy with efficiency, spatial coherency also allows to replace the 5×5 Gaussian filter by a **5×5 Box filter**, which retains over 98% similarity and computes 128% faster than the 11×11 2-D Gaussian filter (68% faster than the 1-D filter pair) when implemented using an *integral image* (CHEN; BOVIK, 2011). An integral image or summed area table is a technique consisting of a data structure and algorithm that allows fast computing of, for instance, the means of spatial regions of an image, as in a Box filter. For a filter window size of n, a Box window requires n^2 -1 additions where a 2-D Gaussian window additionally requires n^2 multiplications, as well as one division at each pixel, whereas the integral image requires simply three additions and one subtraction, besides one addition per pixel when the summed area table is built.

Fig. 3.12 is the summed area table computed from a 5×5 table filled with values of 1, for ease of demonstration. For each pixel (x, y), the summed area is its value plus the values of pixels (x-1, y) and (x, y-1) minus the value of (x-1, y-1). The

grey area, in turn, is an example of any arbitrary area that may be quickly computed by a simple operation of its vertices, emphasized in bold: add the values in the first and last vertices and subtract the second and third vertices: 1 + 12 - 3 - 4 = 6 is the area.

Fig. 3.12 — Sample summed area table.

1	2	3	**4**	5
2	4	6	8	10
3	6	9	**12**	15
4	8	12	16	20
5	10	15	20	25

Source: the Author.

Instead of sliding the convolution window, a 5×5 or 7×7 Box filter can be used to segment and **downsample** each region of the image, still retaining 98% similarity and computing 382% faster than the 11×11 2-D Gaussian filter (255% faster than the 1-D filter pair), instead of 128%. Besides considerably improving computing speed, this pooling technique is also effective in that all original color levels are uniformly condensed by the resulting mean, so each original pixel provides equal contribution to the index. In this case, it is especially important to avoid a window dimension of 8 or 16, to avoid overlap with the image's spatial coding blocks, which may impair the detection of blocking artifacts. This technique is comparable to the Box sampling used by Wang, Lu and Bovik (2004).

The **downsampling Box filter** technique also promotes *scalability*: for an arbitrarily larger video frame, a proportionally larger window downsamples to the same size for assessment so that the only increases in computation cost are during the gradient filtering and box downsampling. This method may be particularly effective for Full HD and Ultra HD content.

3.3. Computing the similarity and stabilizing the quality map

The similarity index in SSIM is defined by a division of squared values for the image features, as defined earlier in (1). The index varies between -1.0 (diametrically opposite) to 1.0 (exactly equal). Enhancing, pooling and computing the similarity index for each pixel produces the *quality map* (also called the *error map*), which fundamentally is simply a processed image. To facilitate visual analysis, the values may be normalized to a greyscale dynamic range, such as 0 to 255, where darker pixels indicate lower similarity than brighter ones. This quality map is an effective tool for validating the techniques as well as the implementations: blocking artifacts in the quality map produced by a Box filter result in, however slightly, decreased subjective quality prediction; replacing the Box filter by a Gaussian filter softens hard borders and improves quality prediction. By the same token, inadequate division stabilization techniques produce black artifacts that stand out in the quality map and result in decreased quality prediction. Fig. 3.14 at the end of this section gives representative quality maps for the investigated SSIM-based techniques.

Clearly, an adequate stabilization technique for the similarity division is paramount for the index's quality, by preserving numeric proportions between the features. SSIM achieves stabilization by adding the constant C_1 to both the numerator and denominator of the similarity division, which may be called **arithmetic stabilization** or **stabilization by constant**. C_1 is defined by (9), where K_1 is an experimental small constant and L is the dynamic range of the image, typically 2^8-1 = 255. In the GMSD reference implementation, however, C_1 has been adjusted to (10) as a result of that particular research.

$$C_1 = (K_1 L)^2 = (0.03 \times 255)^2 = 58.5225 \quad (9)$$

$$C_1 = K_1 L^2 = 0.0026 \times 256^2 = 170.3936 \quad (10)$$

Suppose an original variance or gradient magnitude of 1 that is distorted to 2: with the usual stabilization constant of 58.5, the index produced is 0.9843; whereas without constants, the index is 0.8. Evidently, the second index is more proportional

to the input than the first. When such a magnitude of 1 is distorted to 8, the index is 0.6032 with constants and 0.2462 without. On the other hand, any distortion at higher magnitudes affects less the index: 150 distorted to 200 produces 0.96 with both methods. This means that, at lower magnitudes, the stabilizing constant reduces the weight of differences, which could have a negative impact on quality prediction, although this has not been confirmed by the experiments in Chapter 5. As variance and gradients may be considered analogous to contrast for MPEG-based IQA purposes, increasing the response to differences in lower magnitudes may better correlate to light adaptation and improve the index's effectiveness.

Rouse and Hemami (2008a) offer similar criticism and propose **stabilizing by a logical treatment**, as exemplified by listing 3.1 (modified and optimized for logical precedence). These operations allow to discard the stabilizing constant, retaining the full range of intensities for comparison and full numeric proportions between (most of) the intensities. A secondary contribution is that the amplitudes between similar indexes are generally widened, which further facilitates comparison. This technique improves adherence to Weber's law of light adaptation, as proposed by Z. Wang et al. (2004). Indeed, they admit the constant stabilizer's value to be "somewhat arbitrary".

Listing 3.1 — Logical treatment for quality map stabilization.

```
if σx = 0 and σy = 0 then
      SSIM = 1; //Signals considered equal, even if 0.
else
      if σxy = 0 then
            SSIM = 0; //One signal is 0, the other is not.
      else
            SSIM = 2σxy /(σx² + σy²); //Structural similarity index.
```

Even with logical treatment, however, the most critical problem remains: if either value from the feature enhancements is zero, the index value will be zero, whether the other value is 1 or 255. In this case, there still occurs loss of numeric proportions, and worse, useful information for comparison is destroyed. This is especially the case with gradients, although less so with covariance.

When comparing gradients, the magnitude is always 0 or a positive value, comparable to the dynamic range of the color levels in the original image. Because there are no negative values, gradient features allow a shift of +1 to the magnitudes, preventing any 0 value and thus preserving a linear numeric proportion between all possible values while requiring no particular division stabilization. Thus, this modification of feature enhancement allows to ignore division stabilization altogether, while meeting all the requirements. This is the justification for the **shifted feature (gradient) magnitude** proposed in this work.

Table 3.1 illustrates the distortions in arithmetic stabilization and logical stabilization, as well as the greater mathematical consistency of the shifted gradient method. Three stabilization techniques are presented along with pairs of proportional gradient magnitudes. The proportions between the pairs (0, 1) and (0, 8) are supposed to be similar to those of respectively (1, 2) and (1, 9), but this is not the case for C_1=0, and, although with C_1=58.5 the indexes are proportional, they are also significantly higher than the absolute differences in magnitudes. Finally, the pair (150, 200) are of such higher magnitudes that the index shows highly discrete sensitivity to the differences, suggesting there is a magnitude threshold beyond which it becomes futile to compute the index; this may be an additional opportunity to optimize implementations of SSIM-based indexes.

Table 3.1 — SSIM value samples for different stabilization methods.

∇S	∇V	*SSIM* for C_1=58.5	*SSIM* for C_1=0	*SSIM* for C_1=0, ∇S+1, ∇V+1
0	1	0.9832	0.0000	0.8000
1	2	0.9843	0.8000	0.9231
0	8	0.4776	0.0000	0.2195
1	9	0.5445	0.2195	0.2289
150	200	0.9600	0.9600	0.9604

Fig. 3.13 presents (a) a sample high quality image detail and (b) the same image compressed in H.264 format to a very low data length, such that much visual information is lost by blurring, and new edges are added as blocking artifacts. With

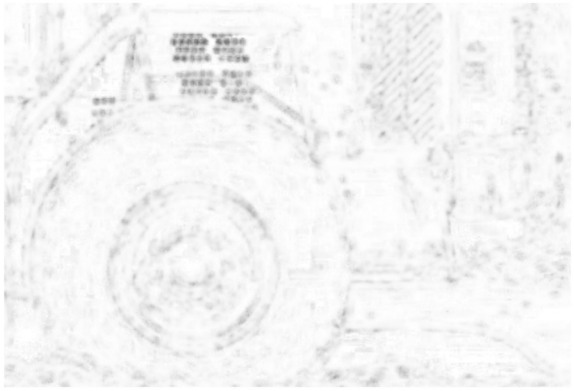
the `jVQA` tool presented in Chapter 4, quality maps are produced using several representative SSIM-based metrics. In Fig. 3.14, The original SSIM quality map (a) directly reflects the original elements that become blurred, as well as the new edges from blocking; however, it is particularly monotonic, which implies that most error information is uniformly weighted rather than perceptually weighted. On the other hand, the quality map (b), produced by Fast SSIM, clearly produces a wider range of gray tones, suggesting it may be more perceptually-weighted than SSIM. However, when the stabilizing constant is removed, the quality map (c) reveals black artifacts which result from the loss of information that has also occurred in (b) but was masked by the arithmetic stabilization. These artifacts are also evidenced in the quality map (d) produced by that same metric without spatial pooling. However, without any stabilization, the SG-Sim quality map (e) neither produces loss of information, nor introduces any distortion, and presents a wide dynamic range which is predicted to improve perceptual correlation. This is further evidenced by the unfiltered version (f).

Fig. 3.13 — Detail of a source image and its compressed counterpart.

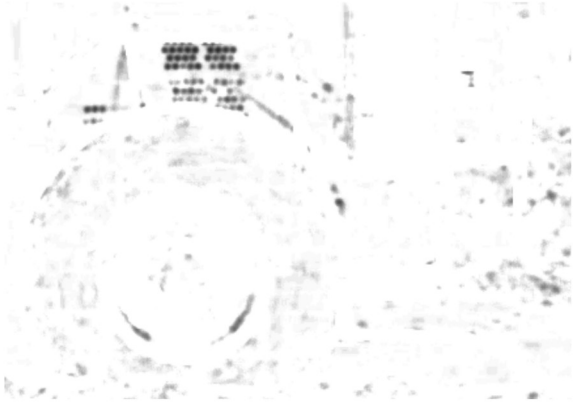
(a) Source image. (b) H.264-compressed image.

Source: the Author.

Fig. 3.14 — Quality maps for several VQA metrics.

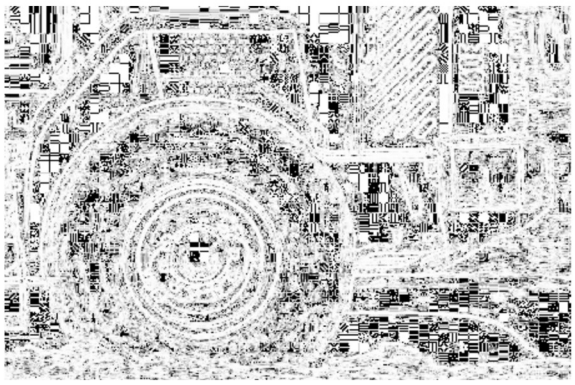


(a) SSIM quality map.

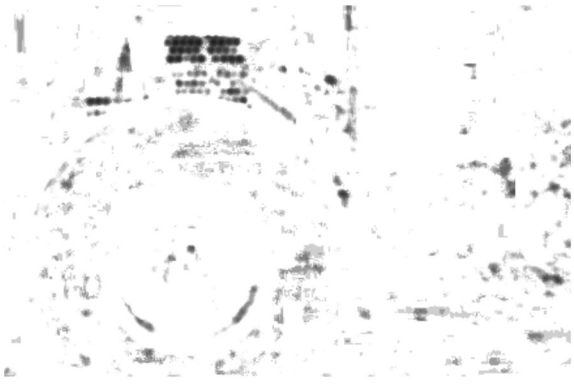


(b) Fast SSIM quality map.

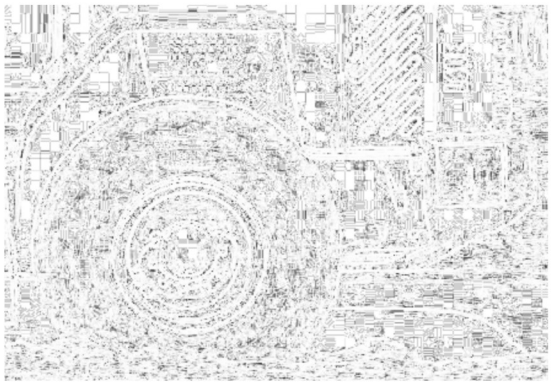
(c) Logical-stabilized Fast SSIM quality map.



(d) Unfiltered logical Fast SSIM quality map.



(e) SG-Sim quality map, without stabilization.



(f) Unfiltered SG-Sim quality map.

Source: the Author.

3.4. Pooling the quality map

In order to obtain a quality index for an entire image (or video frame), usually the **mean** of each pixel's SSIM index is computed. This is the simplest technique, effective and fast to compute. Other techniques have been proposed (WANG; SHANG, 2006; MOORTHY; BOVIK, 2009b) that are more sophisticated, thus slower.

Because of the low latency requirement for encoding and streaming applications, the only other quality map pooling technique considered in this work was the **standard deviation**, exclusively as an essential part of the GMSD index. Computing the standard deviation of the quality map is appropriate for this index due to its lack of feature pooling, which would otherwise increase complexity and stabilize the results.

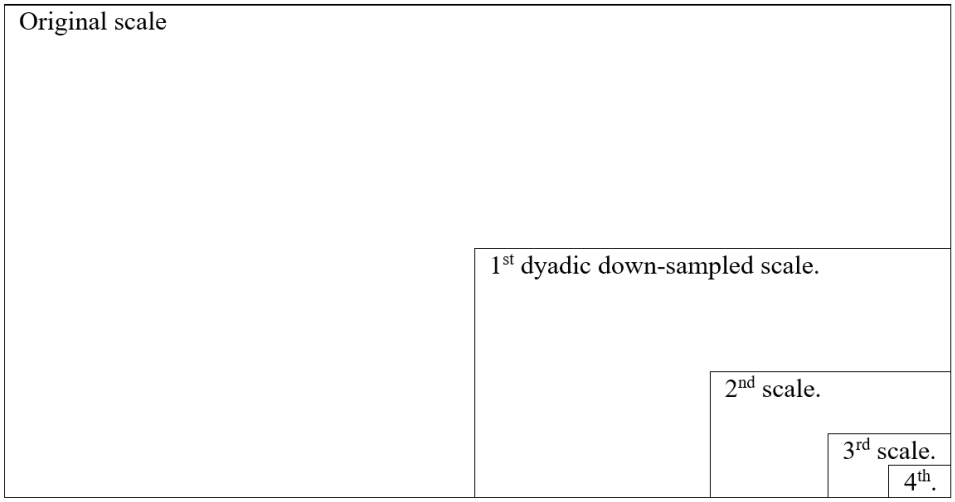
3.5. Downscaling the input resolution

Most research on SSIM and related metrics work upon images of 768×512 or similar resolutions. According to display resolution and viewing distance, they recommend to downsample the images to 384×256 by a 2×2 Box filter **(dyadic downsampling)** before computing the index, which achieves greater DMOS correlation. A more general recommendation is given by Z. Wang et al. (2011): for a viewing distance of 3 to 5 times the image's height and width, downsample by an integer factor so the resulting lower resolution's least dimension approximates 256. Of course, when comparing only SSIM-based metrics applied to content of uniform resolution, that distinction becomes moot. It follows that our comparative investigation of SSIM techniques disregards this single downsampling, working instead on the full original resolutions.

Experiments suggest that lower resolutions produce higher SSIM values and these values become more compressed (flattened), in such a way that in several cases the correlation to DMOS after nonlinear regression improves. This is explained because downsampling is an operation involving a low-pass filter, which smooths and blurs visual information, in both the source and processed videos, which then

become less differentiated. Thus, the SSIM index naturally rises at lower resolutions, especially for high-noise content such as cinema film grain content. This is a *scalability* problem, particularly for *adaptive streaming*, because an index at a 1080p resolution will usually correlate to a lower perceptual quality than the same index at a 480p resolution, for instance.

Fig. 3.15 — The four dyadic downsampling scales of MS-SSIM.



Source: the Author.

Natural images typically include objects of several scales. For example, a picture of a car has a large-scale object, the car, against a background, and the car has smaller-scale features such as windows and wheels. The HVS perceives quality in many such scales. This is the motivation for the **Multi-scale SSIM (MS-SSIM)** (WANG; SIMONCELLI; BOVIK, 2003), which downsamples the content four times by the dyadic method (see Fig. 3.15), then computes SSIM for each scale, including the original, and finally produces an aggregate index with different exponential weights for each scale: the powers of 0.0448, 0.2856, 0.3001, 0.2363, and 0.1333. For example, a 768×512 image is processed in MS-SSIM at the scales of 768×512, 384×256, 192×128, 86×64, and 43×32, and the SSIM index for each is computed and adjusted by the power of the weight of the respective scale. The exponential weights increase their respective indexes in inverse proportion, so the first scale is the most increased, and the third scale is the least.

MS-SSIM significantly improves DMOS correlation and is one of the most

effective SSIM-based indexes. Such increase may be due to the index's tendency to improve correlation at lower resolutions, which could mean the gain is more a matter of resulting numeric proportions than a matter of effectiveness of image processing techniques. This consideration, however, requires more thorough investigation than has been conducted in this work, remaining an open line of inquiry. **Fast MS-SSIM** (CHEN; BOVIK, 2011) further proposes to ignore the original resolution, which provides the least contribution, in order to decrease complexity while retaining significant quality. This technique allows for twice the computing speed of full-scale SSIM and four times that of five-scale SSIM.

3.6. Computing luma similarity

The original SSIM includes not only contrast and structure similarity, but also a luma similarity term. In an universal IQA index, the luma term is important because a distorted image may be structurally identical yet darker or brighter (or of different hues, in the case of chroma planes). This should not typically be the case in mature video encoders, however. Because of its low contribution, luma similarity in MS-SSIM is only computed for the smallest scale, which is a mere 43×32 image in the case of that paper. Rouse and Hemami (2008b), and later GMSD as well, recommend to disregard this component entirely.

$$l(x,y) = \frac{2\mu_x\mu_y + C_2}{\mu_x^2 + \mu_y^2 + C_2} \quad (11)$$

A description of SSIM techniques would not be complete, however, without the least consideration of the luma similarity component, which may be referred to as *l*. In the usual implementations of SSIM, that are applied simply to the luma plane of a Y'V12 image, the image itself is the very luma signal, requiring no pre-processing. However, the luma similarity index proposed in SSIM is concerned with local pooled luma, so the input signal is blurred by either a Box or Gaussian filter, according to the methods discussed in Section 3.2. In (11), μ_x is the locally-pooled luma for image *X*, and C_2 is a constant for index stabilization as discussed in Section 3.3.

3.7. Closing remarks

This chapter reviewed the component techniques of many SSIM-based indexes of interest to video quality assessment in the context of adaptive streaming over HTTP and its particular constraint of low processing latency. This investigation allowed to compose a new, more effective index, the **Shifted Gradient Similarity** (SG-Sim, at Section 3.1.1), as well as to propose optimized feature pooling filters for improved computing speed: the **2σ-Gaussian 1-D filter pair** (Section 3.2.1) and the **downsampling box filter** (Section 3.2.2).

The next chapter discusses in detail the implementation of all these VQA techniques in the `jVQA` software. `JVQA` is presented as a toolset for customizing SSIM-based indexes, applying them to encoded versions of video content and studying their behavior and contributions. In this, it proves faster, more practical and flexible than experimenting in Matlab with pure Y'V12 streams.

4. THE VIDEO QUALITY ASSESSMENT IN JAVA TOOLSET

In order to properly compare the SSIM-based video quality assessment (VQA) techniques described in Chapter 3, a customizable software tool is required, which should also be practical for batch processing. Most academic implementations of VQA metrics are published in the form of MATLAB code (XUE et al., 2014). MATLAB is a relevant general-purpose numerical computing environment commonly available in academic research institutions and provides the required flexibility. However, an ideal VQA "laboratory" software should have computing performance similar to practical applications, with application-specific optimizations, and work with common video formats; MATLAB's performance is limited by being interpreted code, instead of natively-executed compiled code, and input is restricted to raw $Y'C_BC_R$ video (YUV) files, which are considerably heavier for handling than compressed video (including lossless formats). In addition, MATLAB is proprietary, and the programs written are not quite portable. For these reasons, a more practical software is required for this work.

Moscow State University (MSU) publishes the Video Quality Measurement Tool (VQMT)[27], a proprietary Windows application that allows processing full-reference metrics such as MSE, PSNR, SSIM, MS-SSIM, 3-SSIM and ST-VSSIM, as well as no-reference metrics from MSU for measuring blurring, blocking, noise etc. VQMT supports as input Audio-Video Interleave (AVI), AviSynth script (AVS), YUV4MPEG2 (Y4M), YUV and bitmap (BMP). AviSynth[28] is a frame server that interprets scripts and forwards the commands to the underlying operating system's decoders, such as the Microsoft DirectShow and Video for Windows (VFW) APIs, or even standalone decoders such as FFmpeg[29] (WAGGONER, 2010). VQMT provides a powerful VQA laboratory, including plots and graphics, as well as CPU and GPU code optimizations. However, it is proprietary and requires a license to enable processing of HD content, which is one of the requirements for this investigation.

Inspired by VQMT, the new Video Quality Assessment in Java (`jVQA`)

27 <http://compression.ru/video/quality_measure/video_measurement_tool_en.html>.
28 <http://avisynth.nl>.
29 <http://ffmpeg.org/general.html#Supported-File-Formats_002c-Codecs-or-Features>.

software laboratory has been developed for this work, addressing the aforementioned requirements. `JVQA` is designed as a flexible, reusable, extensible and efficient open-source toolset for SSIM-based indexes, with decoding provided by the AviSynth and FFmpeg open-source platforms. Implemented in Java, it allows multi-platform usage, and supports 4K (3840×2160) Ultra-HD content.

The design requirements for the `jVQA` software are described in Section 4.1. Discussion on methods for calling native software decoders from Java is the topic of Section 4.2. Considerations on optimized image processing tools are given in Section 4.3. Design specifications for `jVQA` are presented in Section 4.4. Finally, computing performance considerations are given in Section 4.5.

4.1. JVQA design requirements

The main purpose of `jVQA` is to implement SSIM-based indexes for experimentation and comparison, supporting at the very least 720p HD content. Support for AviSynth input is also desirable, because this frameserver provides relevant pre-processing operations (resizing, cropping, frame rate adjustment, deinterlacing, denoising, deblocking and color correction), and it abstracts video decoding, including synchronizing the frames from decoded compressed video, as well as being a common input between the compared VQMT, and the `x264` encoder, especially within the MeGUI encoding suite[30] (WAGGONER, 2010). AviSynth allows to substitute the large raw video files by considerably smaller and more practical losslessly-compressed (by 1:5) video files. Decoding is performed through scripts such as in Listing 4.1, where Microsoft DirectShow decodes the file, the frame rate is explicitly synchronized, and the output is converted to "YV12" if needed. Alternatively, an FFmpeg compilation including support for AviSynth input may substitute explicit AviSynth support. AviSynth is primarily based on Microsoft Windows, but the AvxSynth port for Linux is also available online[31]. Both FFmpeg and AviSynth (through the operating system decoders) support virtually all common

30 <http://sourceforge.net/p/megui/wiki/Home>.
31 <http://www.avxsynth.org>.

video formats, old and new.

Listing 4.1 — AviSynth script example.

```
DirectShowSource("video1.mp4", fps=23.976, convertfps=true).AssumeFPS(24000,1001)
ConvertToYV12()
```

JVQA is implemented in the Java programming language as an attempt towards multi-platform availability, by means of the Java virtual machine that is available for most common operating systems[32]. Java implements the object-oriented (OO) programming paradigm, which promotes low code redundancy and high granularity, capable of reducing programming mistakes, as well as promoting reusability, extensibility, and component substitution (GAMMA, 1994). OO design patterns also facilitate dynamic VQA index composition, so the techniques described in Chapter 3 may be combined and compared in several configurations.

Video represents a considerable volume of data, and the relative complexities of several VQA techniques is of particular interest, so it is important that jVQA be computationally efficient, implementing common optimizations in its algorithms, reflecting consumer applications. These include convolving images with approximated integer filters instead of precise floating-point filters; decomposing the 2-D Gaussian filter by its 1-D components; employing the integral image; optimizing array operations in general; storing data in small primitive data types such as *short* and so on. Finally, jVQA should provide a command-line interface (CLI) that facilitates batch processing in experimental sessions, with fully configurable metric composition parameters. A graphical user interface is also desirable for discrete tests and demonstrations.

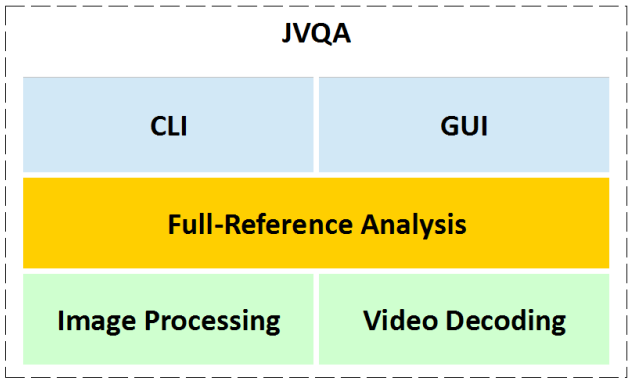
The consolidated requirements are the following:

1. customizable SSIM-based index composition;
2. design for comparative experimentation;
3. ability to process HD content;
4. AviSynth and/or FFmpeg decoding;
5. multi-platform deployment;

32 <http://docs.oracle.com/javase/8/docs>.

6. design for reusability, extensibility, and component substitution;
7. computation efficiency with common image processing optimizations;
8. command-line interface for batch processing;
9. graphical user interface for fast discrete tests and demonstrations.

Fig. 4.1 — Logical layers of the architecture for `jVQA`.



Source: the Author.

The architecture of `jVQA` may be generalized as logical components and layers as in Fig. 4.1. The following sections describe the specific designs of each component.

4.2. Native video decoding for Java applications

Both AviSynth and FFmpeg are available for several operating systems, but are distributed as native compiled code libraries. The Java Native Interface (JNI)[33] allows Java applications to call native code. There are also several open source libraries provided by the programming community for facilitating native access besides JNI, such as the Java Native Access (JNA) library[34]. The latter is especially convenient for providing a tool for automatically generating the Java code interface for accessing any particular library.

JNA thus allowed a standalone component for `jVQA`, the Java Native Access for AviSynth – `jNAvi`, to be developed (see Fig. 4.5). `JNAvi` is concerned solely with decoding video files through native AviSynth and providing convenience functions for retrieving byte streams for each plane in the Y'V12 colorspace, as well as some of

33 <http://docs.oracle.com/javase/8/docs/technotes/guides/jni>.
34 <http://github.com/twall/jna>.

the video file's interesting properties. `JNAvi` was developed with Eclipse IDE[35] and Git source control[36] provided by SourceForge, which publishes the project as open source at <http://sourceforge.net/projects/jnavi/>.

Although AviSynth decoding is available in common operating systems such as Windows, Linux and Mac OS, the FFmpeg (as well as its alternate version, Libav) decoder package is more widely available and more practical as a standalone library, since it usually does not depend on the decoders from the underlying system, independently supporting 225 media file formats and 170 video stream formats. The JavaCPP library[37] facilitates JNI access with no additional overhead and includes a preset implementing full FFmpeg access, along with the native binary dynamic-linked libraries for Android, Linux, Mac OS X, and Windows; these libraries are also integrated into `jVQA` (Fig. 4.5). FFmpeg also conveniently provides more metadata for the input files. However, it became evident during the development of `jVQA` that FFmpeg video decoding does not guarantee frame accuracy, due to variable frame rates or the nature of compressed video: P frames and B frames, for instance, are not necessarily encoded in the same order as they are intended to be played. `JVQA` has yet to treat such synchronization problems, so currently AviSynth remains the recommended input format (whether decoding is performed by `jNAvi` or by FFmpeg, which supports AviSynth input as well).

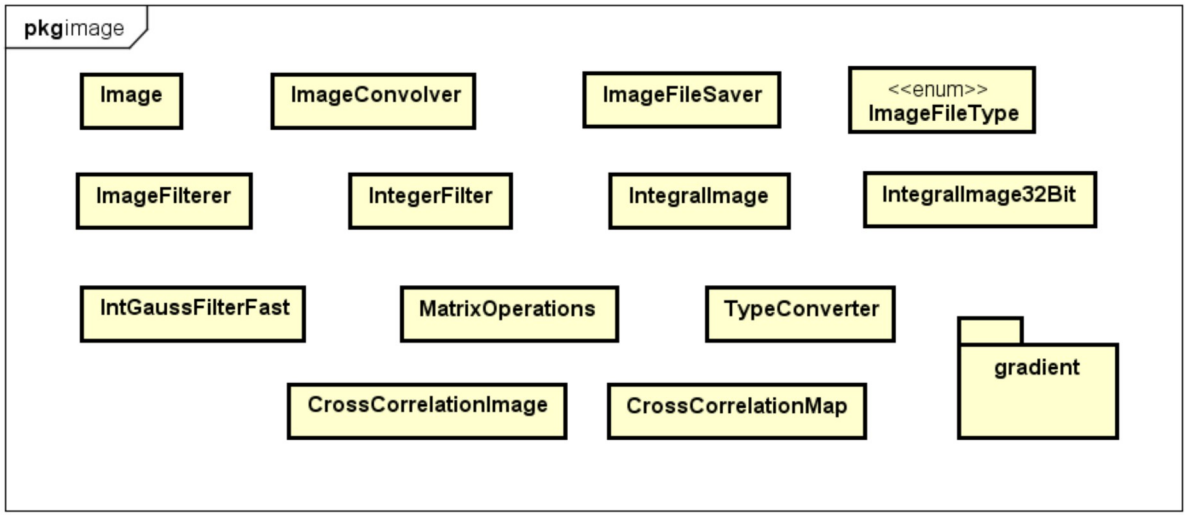
4.3. Image processing tools

`JVQA` requires many image processing tools, the most relevant of which having been discussed in Chapter 3. Although effective libraries such as OpenCV[38] and ImageJ[39] were available, the tools were written from scratch in order to avoid costly data structure conversions, additional calls to native code and unnecessary floating-point operations; this also allows stricter memory management, optimized integer

35 <http://www.eclipse.org>.
36 <http://git-scm.com>.
37 <http://github.com/bytedeco/javacpp-presets>.
38 <http://opencv.org>.
39 <http://imagej.net/Welcome>.

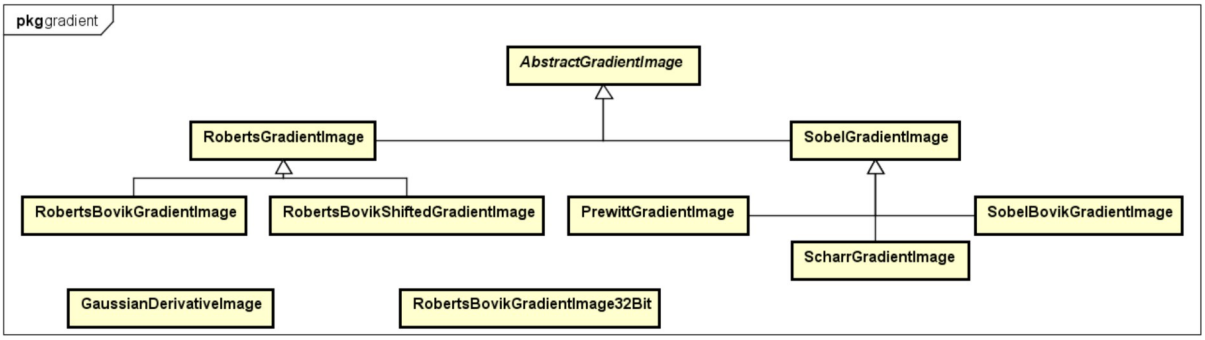
operations and application-specific operations such as correlation images with dynamic ranges of 256×256 (referred as 32-bit images in the design due to the 32-bit data type used for its pixels). Fig. 4.2. illustrates the image processing package in `jVQA`.

Fig. 4.2 — Image processing tools in the `image` package.



Source: the Author.

Fig. 4.3 — Gradient implementations among the image processing tools.



Source: the Author.

`JVQA` requires a wide array of gradient operators to implement SSIM-based metrics, as illustrated in Fig. 4.3 and described in Chapter 3. However, the experiments in Chapter 5 indicate that the differences in VQA results for each of these operators is negligible, with other factors such as feature pooling and similarity stabilization carrying substantially more weight. For this reason, the Roberts operator optimized by the integer approximation proposed by Bovik in Fast SSIM is the most

effective of these operators in the context of VQA for streaming. The Gaussian derivative is a tool for VQM (PINSON and WOLF, 2004), which has yet to be implemented in `jVQA`.

4.4. JVQA design specification

The application tool requires the user to input the two video files to compare, along with the specific settings for visual quality index (VQI):

- input scaling method: full-scale, half-scaled, downscaled to 256, four-scaled, or five-scaled;
- feature enhancement method: variance, gradient, shifted gradient, two-component variance, three-component variance, four-component variance, or gradient deviation;
- feature pooling method: global, 3σ Gaussian, 2σ Gaussian, Box, or downsampling Box;
- feature pooling window size: an integer value within the range 0-20;
- similarity index stabilization: logical (full dynamic range) or arithmetic (distorted by constants); and
- luma similarity: disregard, global, 3σ Gaussian, 2σ Gaussian, Box, or downsampling Box.

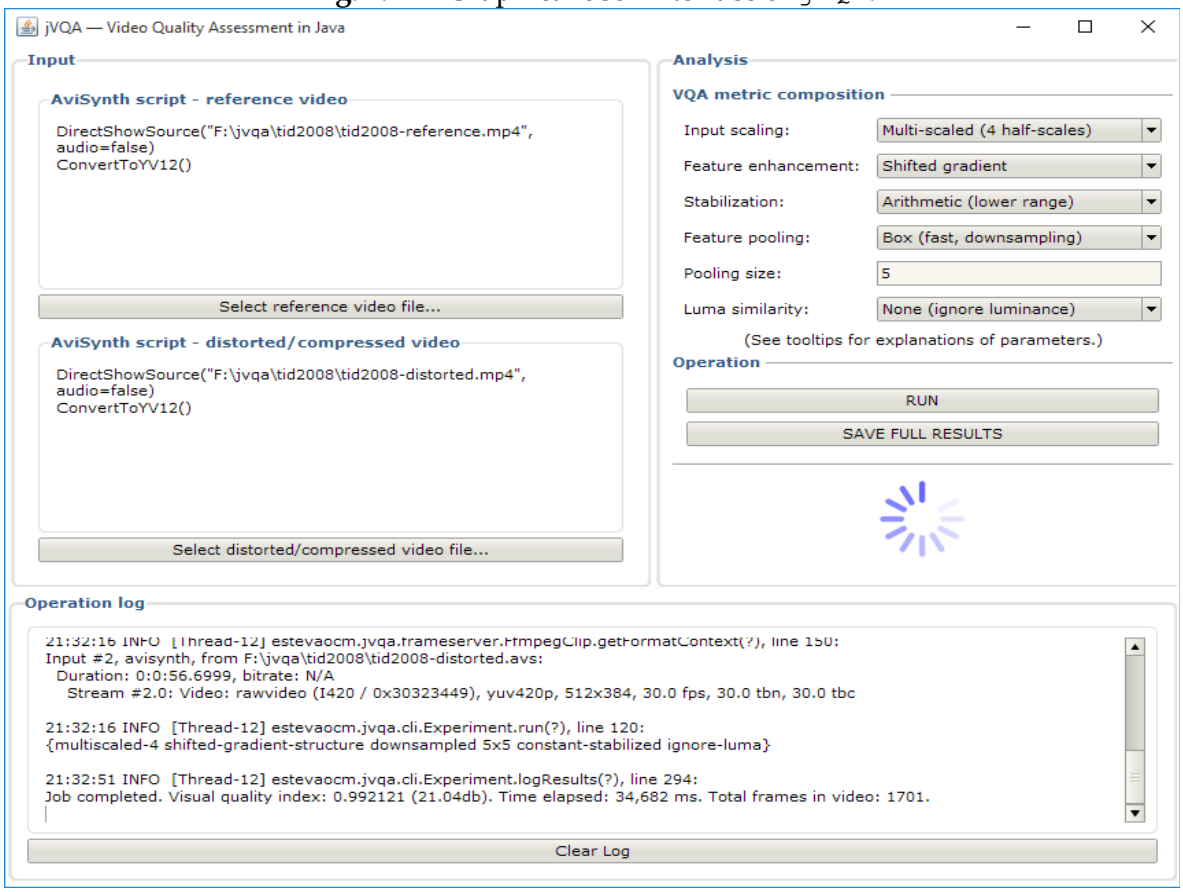
`JVQA` must compute the VQI comparing the two input files and output to the console the overall index and the computing time for the video analysis. These data, together with the VQ indexes for each individual frames, are to also be automatically recorded to text and/or CSV (comma-separated values) files. Finally, an optional function for outputting the quality map of individual frames in a lossless image format such as PNG is important for analyzing behavior and distortions (artifacts) of the indexes.

The application entry point is implemented in the `JVQACommandLine-Interface` (CLI). If no operation parameters are provided, the `JVQAGraphical-UserInterface` (GUI) based on Apache Pivot[40] is launched (Fig. 4.4). In either

40 <http://pivot.apache.org>.

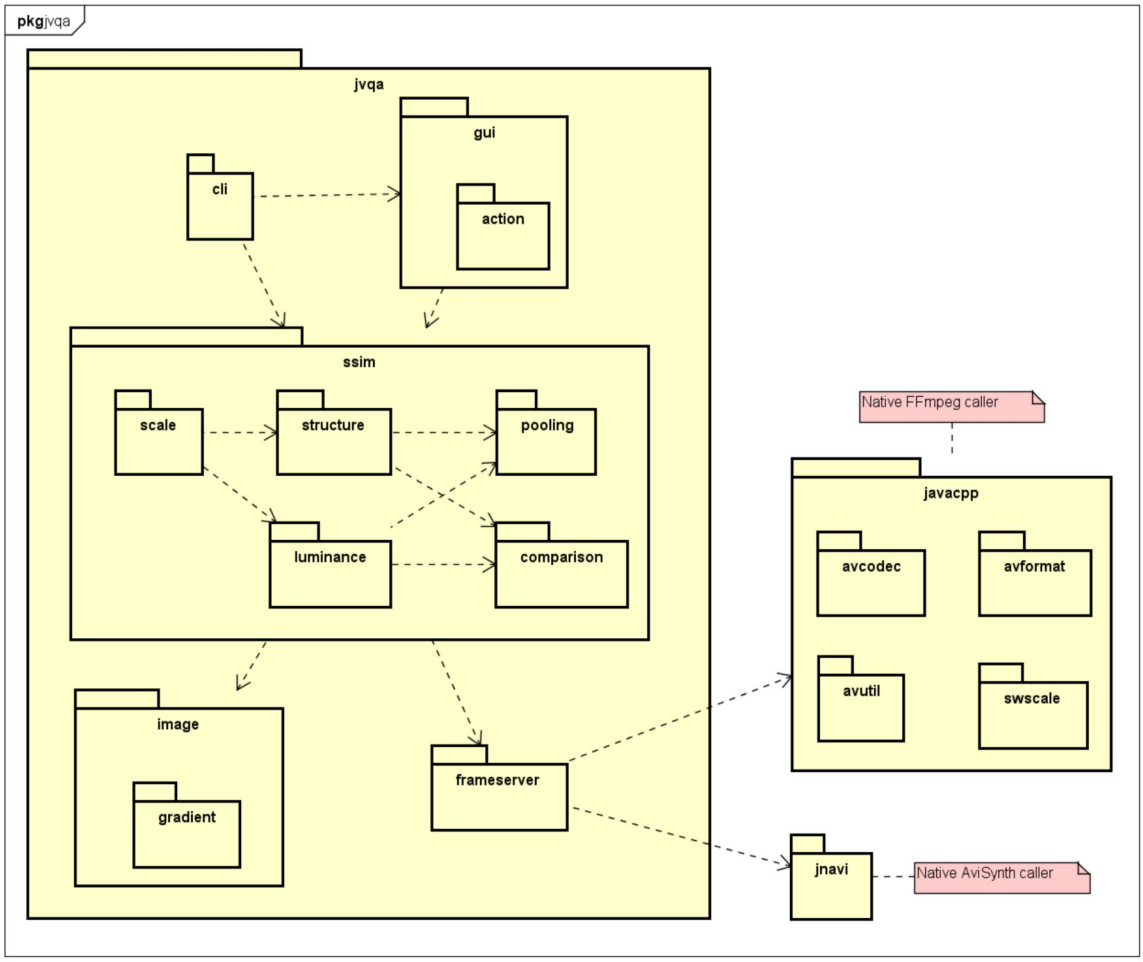
case, the user parameters are parsed into a `CustomIndex` object, modeled as a "bean". Next, an `Experiment` object receiving the `CustomIndex` and the video file paths is constructed, in turn instantiating a `ComparisonFrameServer` encapsulating either a `FfmpegComparisonFrameServer` or an `AVSComparison-FrameServer`. `Experiment` then instantiates a `StructuralSimilarityIndex`, that also depends on the `CustomIndex` and `ComparisonFrameServer` instances. Experiment is `jVQA`'s main operation controller and manages the workflow and output, whereas `StructuralSimilarityIndex` is responsible for the actual video analysis. The UML diagram in Fig. 4.5. illustrates the dependencies in this outermost layer of the software.

Fig. 4.4 — Graphical user interface of `jVQA`.



Source: the Author.

Fig. 4.5 — `JVQA` Architecture with package dependencies.



Source: the Author.

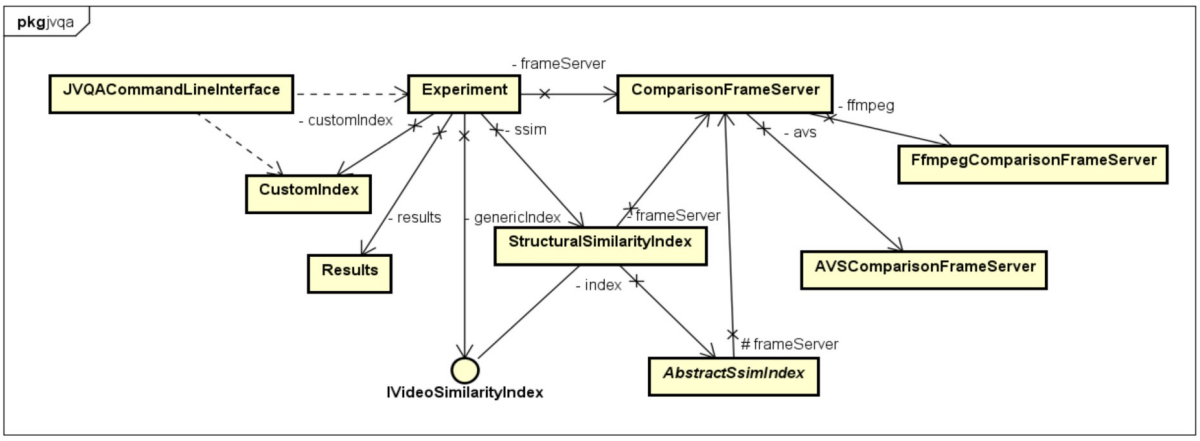
The `StructuralSimilarityIndex` class implements the "abstract factory" design pattern, by which the settings in the `CustomIndex` bean argument produce a composition of each respective metric component, returned as a dynamic specialization of `AbstractSsimIndex` defined during runtime (Figs. 4.6 and 4.7). The settings produce specialized handler objects in composition order:

- a specialized object implementing the `IFilteredPooling` interface (Fig. 4.8);
- a specialized object extending the `AbstractStructureComparator` class (Fig. 4.9), encapsulating the `IFilteredPooling` instance;
- a specialized implementation of either of the interfaces `ICorrelationIndex` or `ISimilarityIndex` (Figs. 4.10 and 4.11) as a member of the

`AbstractStructureComparator`;

- a specialized object extending the `AbstractLuminanceComparator` class, analogous to `AbstractStructureComparator` (Fig. 4.12);
- and, finally, a specialized object extending the `AbstractSsimIndex` class, which encapsulates the structure and luma comparators (Fig. 4.13).

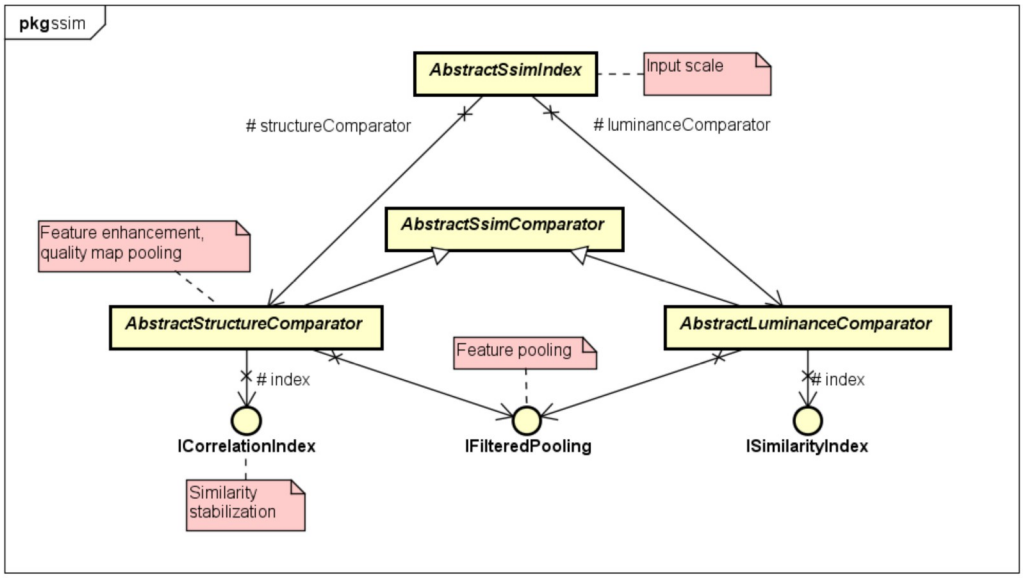
Fig. 4.6 — First layer of the workflow in `jVQA`.



Source: the Author.

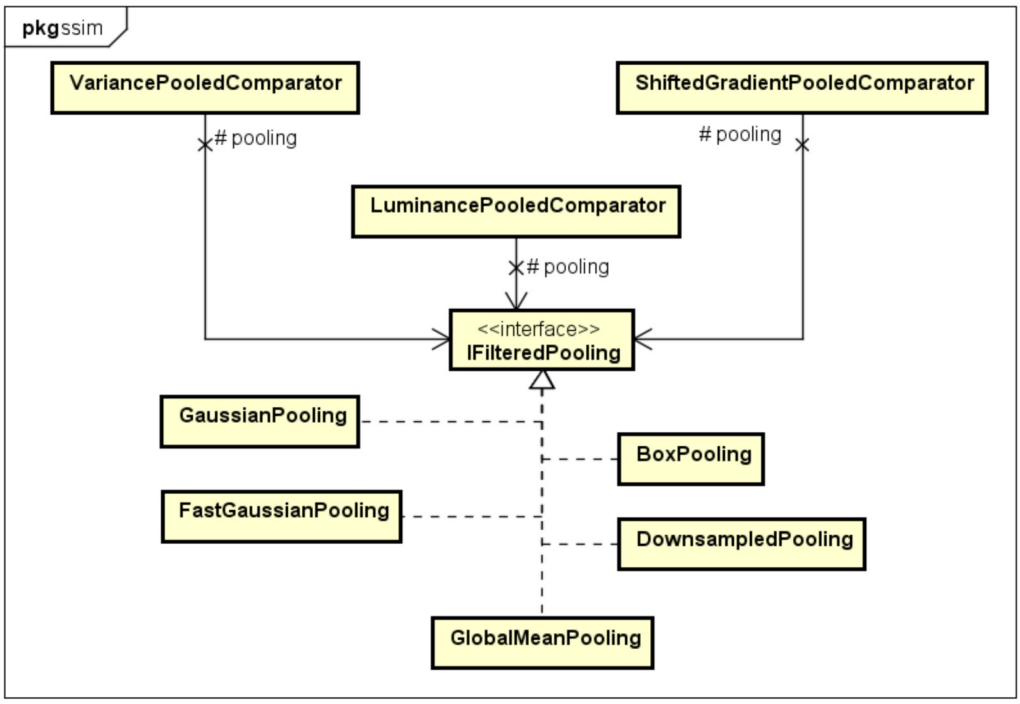
`AbstractSsimIndex`, in turn, provides an implementation of the "strategy" design pattern, by defining a family of algorithms, encapsulating them and allowing interchangeability among them.

Fig. 4.7 — Abstractions and interfaces that provide the structure of the `ssim` package.



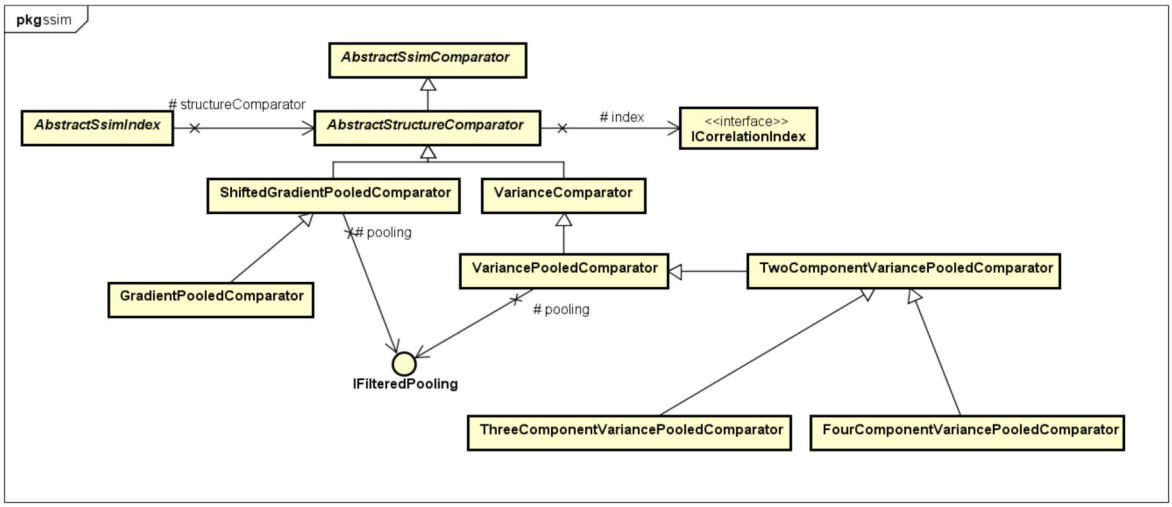
Source: the Author.

Fig. 4.8 — The `IFilteredPooling` interface, its implementations, and referencing classes.



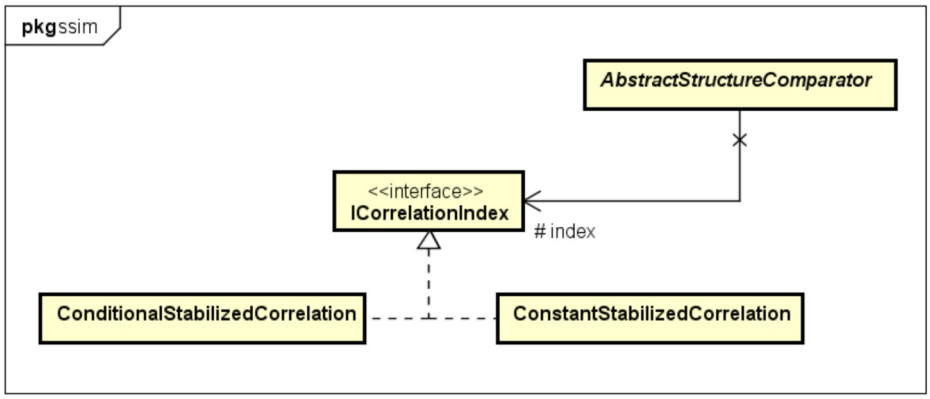
Source: the Author.

Fig. 4.9 — The `AbstractStructureComparator` class and its extending and referencing classes.



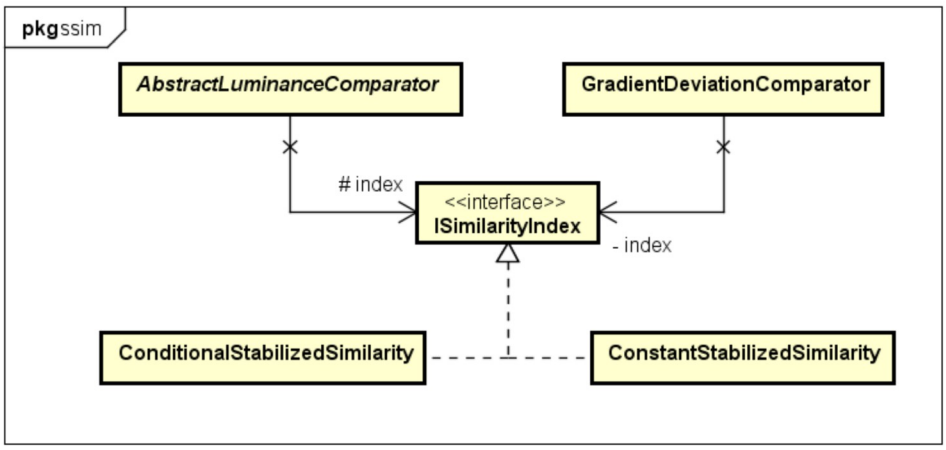
Source: the Author.

Fig. 4.10 — The `ICorrelationIndex` interface, its implementations, and referencing classes.



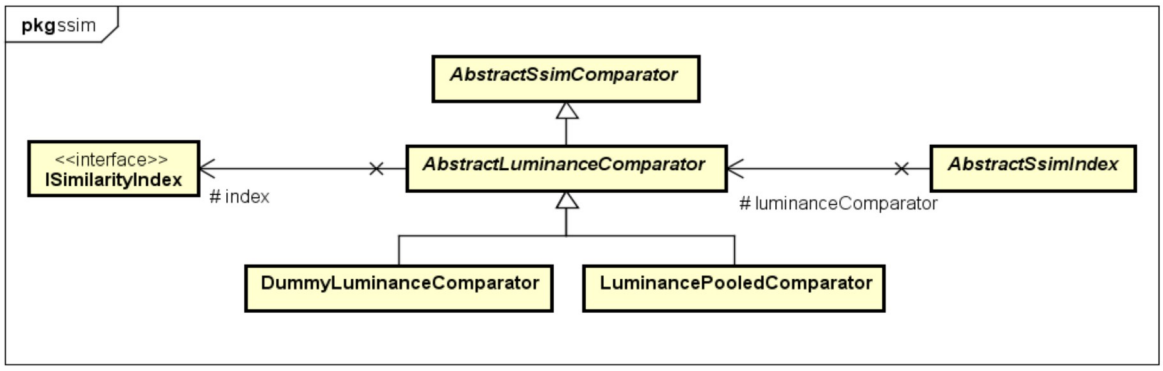
Source: the Author.

Fig. 4.11 — The `ISimilarityIndex` interface, its implementations, and referencing classes.



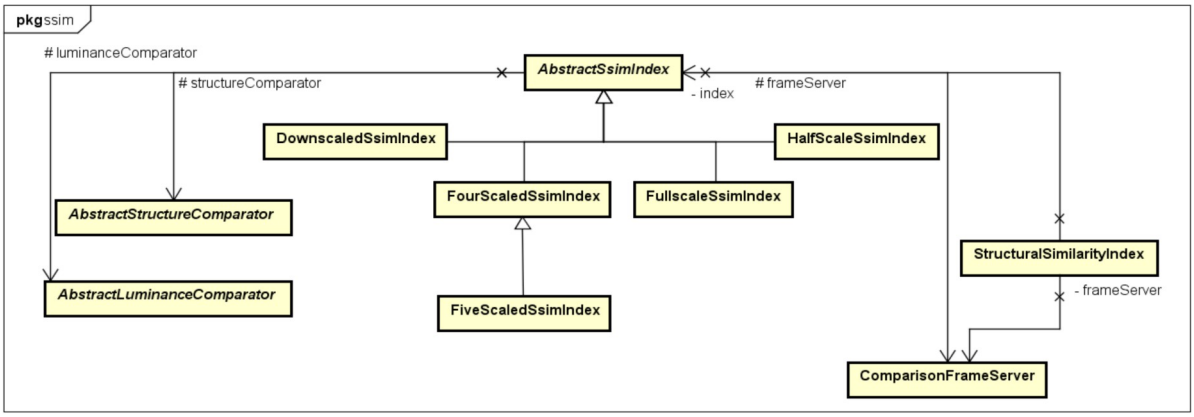
Source: the Author.

Fig. 4.12 — The `AbstractLuminanceComparator` class and its extending and referencing classes.



Source: the Author.

Fig. 4.13 — The `AbstractSsimindex` class and its extending and referencing classes.



Source: the Author.

4.5. JVQA computing performance

`JVQA` is not required to achieve a particular performance level. However, video processing is an especially demanding application, and should be implemented efficiently; particularly, VQA should not require periods of time disproportional to the duration of the content, and interchangeable VQA techniques should be proportionally efficient for comparisons. It is also important to understand `JVQA` computing and memory performance, for consistent experimentation.

`JVQA` memory use is, naturally, proportional to the resolution of the analyzed videos. The Java Virtual Machine (JVM) imposes a limit to dynamic memory allocation, the heap size, which defaults to 256 MiB[41]. This may be redefined when calling the application, by the parameter `Xmx`. For example, 720p HD content requires 384 MiB to be allocated to the JVM, therefore the parameter "-Xmx384m". Further, 1080p Full-HD content requires 640 MiB, and 4K Ultra-HD content requires 1024 MiB. Most operations in `jVQA` impose no known limits other than memory allocation; the known exception is the integral image, required for feature pooling based on Box and downsampling operators, which is limited to 8-bit 4K content due to data types employed (which may be adapted in the future for greater resolutions).

`JVQA`'s dependency on FFmpeg or AviSynth may constrain the intention for

41 <http://www.oracle.com/technetwork/java/javase/6u18-142093.html>.

multi-platform support at the binary level. JNAvi requires the AviSynth Windows DLL[42] (although it may be adapted for the respective libraries of other platforms), and FFmpeg must either be bundled with `jVQA` in the specific platform's binary compilation or provided by the system (as in Linux), which means the software package already must be specific to the target system. With such shortcomings, the software might as well have been written in C++, which would further allow CPU and GPU optimizations for improving performance. All the aforementioned limitations regardless, `jVQA`'s current computing performance meets the expectations and practical requirements for the comparative experiments of this work.

4.6. Closing remarks

This chapter detailed the Video Quality Assessment in Java software laboratory, which supports several SSIM-based indexes and is designed for extensibility and batch experimentation. `JVQA` is published as free and open-source software by the GNU Public License (GPL) version 3 at <http://sourceforge.net/projects/jvqa/>. This is in contrast with the similar MSU VQMT software, which is closed-source and requires a paid license to work with HD content.

`JVQA` allows SSIM customization with several options, and supports 4K Ultra-HD content. The importance of the AviSynth frame server for VQA experimentation is also elaborated. The following chapter will present the comparative experiments with the video quality assessment techniques from Chapter 3 that were implemented in `jVQA` and applied to typical content for video streaming.

42 Dynamic link library.

5. EXPERIMENTS

In this chapter, the visual quality assessment (VQA) techniques discussed in Chapter 3 are tested for subjective visual quality prediction performance, computing performance, and correlation with modern psychovisual encoding rate-distortion optimization (RDO). The techniques are performed by the `jVQA` software described in Chapter 4. Section 5.1 details the standard comparison methodology for the various VQA technique configurations and the experiments with a public VQA dataset. Section 5.2 offers SSIM-based index behavior considerations, including scalability concerns. Section 5.3 proposes an experiment for comparing encoder RDO modes with consumer video for improving content production.

5.1. Perceptual video quality model comparison methodology

The various visual quality indexes (VQI) referenced to and studied in this work are published with comparisons to the original SSIM under the methodology proposed by the video quality experts group (VQEG, 2003) of the International Telecommunication Union (ITU-T) for full-reference television (FRTV). This requires a *dataset of full reference distorted images*, i.e., the distorted images must be provided alongside the original image. Each distorted image must also be provided with its respective **differential mean opinion scores (DMOS)**, which the VQEG defines as a quantitative measure of the subjective quality of a video sequence as judged by a panel of human observers, with a continuous range of 0.0 (excellent quality) to 5.0 (worst quality). The VQEG datasets are, however, dated; more modern data sets are provided by the Laboratory for Image & Video Engineering (LIVE) of the University of Texas, which developed SSIM and many of its variations. This work employs specifically the **LIVE Mobile VQA Database** (MOORTHY et al., 2012).

The VQEG methodology evaluates the quality performance of VQA models by their ability to estimate subjective quality, i.e., their predictable and repeatable *correlation* to DMOS, according to three aspects:

- *prediction monotonicity*: the degree to which the model's predictions agree with the relative magnitudes of DMOS;
- *prediction accuracy*: the ability to predict DMOS with low error; and
- *prediction consistency*: the degree to which the model maintains prediction accuracy over the range of tested video sequences.

The VQEG recommends seven performance metrics for assessing quality performance, but most SSIM-based papers limit these to one per performance aspect, as follows:

- **Spearman rank correlation coefficient (RCC)** for correlation monotonicity;
- **Pearson linear correlation coefficient (LCC)** for correlation accuracy; and
- **root mean squared error (RMSE)** for correlation consistency.

In order to compute LCC and RMSE, it is first essential to fit the SSIM values into the DMOS range so the two data sets can be properly compared in the same analysis space. For example, if the DMOS for a sample ranges from 0.5 to 4.0, where lower is better, and SSIM ranges from 0.8 to 0.9999, where greater is better, the data must be adjusted in such a way that 0.9999 corresponds roughly to 0.5, and 0.8, to 4.0. Computing this **SSIM-predicted DMOS ($\mathbf{DMOS_P}$)** is accomplished by nonlinear regression by the logistic function (12).

$$DMOS_P(SSIM) = \frac{B_1}{1 + e^{-B_2(SSIM - B_3)}} \quad (12)$$

In all experiments in this work, LCC is consistent with RMSE, suggesting they may be redundant in this context; indeed, some papers give only RCC and RMSE, without LCC. RCC is mostly consistent with LCC and RMSE, but a few exceptions in our data break any intended notion of redundancy. This level of consistency among the three metrics is also observed in most references of this work. In general, most SSIM-based publications (SESHADRINATHAN; BOVIK, 2009; CHEN; BOVIK, 2011; XUE et al., 2014; among many) emphasize RCC as the most relevant metric for

quality of VQA models. Some recent papers (PONOMARENKO et al., 2009; WANG and LI, 2011; ZHANG et al., 2011; REHMAN; ZENG; WANG, 2015) also include the Kendall rank correlation coefficient, which is mostly consistent with Spearman RCC, but not 100%. The more recent VQEG evaluation for HD-TV (2010) employs only the LCC and the RMSE, and considers RMSE the most relevant. Depending on confidence intervals and statistical significance, the four methods may be considered consistent and redundant.

5.1.1. Testing with the LIVE Mobile Video Quality Database

The LIVE Mobile VQA Database consists of 10 raw Y'V12 reference videos and 200 distorted videos, each of 1280×720 resolution (HD, 720p), at a frame rate of 30 Hz, and 15 seconds in duration. Of the 200 distortions, 40 are compression-based, such that each reference video is compressed with progressively greater loss of information by four signal-to-noise ratio (SNR) levels ($R_1 < R_2 < R_3 < R_4$); this work is concerned only with this subset for encoding quality assessment. Such files are compressed in the H.264 Scalable Video Coding (SVC) format at data rates between 700 kbit/s and 6,000 kbit/s. Although the highest data rate is considerable for the resolution, especially compared to the data rates of Netflix indicated in Table 2.2 (see Chapter 2), SVC is less efficient than High Profile (WAGGONER, 2010), so relevant distortion remain in that version.

As commented in Chapter 4, scientific studies on VQA such as those from VQEG and LIVE are commonly based on raw Y'V12 video files (extension ".yuv"), particularly when working with Matlab. This is also the case for most video encoders, although these are typically extended by decoders such as AviSynth and FFmpeg, which allow the encoders to accept virtually any video file format as input. AviSynth requires a plugin such as RawSource[43] to open yuv files. Raw video files are considerably large, however, so in order to facilitate working with the content where AviSynth or FFmpeg decoding is available, such as in `jVQA`, the content was losslessly encoded to H.264 streams contained in MPEG-4 files (extension ".mp4") by

43 <http://github.com/chikuzen/RawSource_2.6x>.

the `x264` encoder. This was achieved by setting the constant quality parameter of the encoder to zero (`qp=0`), which disables lossy quantization altogether.

Table 5.1 — SSIM-based indexes tested on the Mobile VQA Database.

Visual quality index name	Feature enhancement	Feature pooling	Similarity stabilization	Map pooling	Scaling
SSIM	Covariance	7×7 Gaussian	Additions of constant	Mean	Original
MS-SSIM	Covariance	7×7 Gaussian	Additions of constant	Mean	Original + 4x downscaled
3-SSIM	Sobel gradients of covariance	7×7 Gaussian	Additions of constant	Mean	Original
Fast SSIM	Roberts gradients	7×7 Gaussian	Additions of constant	Mean	Original
GMSD	Prewitt gradients	Global (unfiltered)	Additions of constant	Standard deviation	Original
SG-Sim (Roberts)	Shifted Roberts gradients	7×7 Gaussian	Additions of constant	Mean	Original
SG-Sim (Roberts, logical)	Shifted Roberts gradients	7×7 Gaussian	Logical treatment	Mean	Original
SG-Sim	Shifted Prewitt gradients	7×7 Gaussian	Additions of constant	Mean	Original
SG-Sim (logical)	Shifted Prewitt gradients	7×7 Gaussian	Logical treatment	Mean	Original
Fast SG-Sim	Shifted Prewitt gradients	5×5 downsampling Box	Additions of constant	Mean	Original
5S-SG-Sim	Shifted Prewitt gradients	7×7 Gaussian	Additions of constant	Mean	Original + 4x downscaled
4S-SG-Sim	Shifted Prewitt gradients	7×7 Gaussian	Additions of constant	Mean	4x downscaled
Fast MS-SG-Sim	Shifted Prewitt gradients	5×5 downsampling Box	Additions of constant	Mean	4x downscaled

Along with perceptual quality prediction, this investigation is also interested in comparing the relative computing times of each SSIM version, thus determining not only the best subjective correlation but also the best processing performance, allowing the overall efficiency to be determined. All tests were conducted with 32-bit

`jVQA` with FFmpeg decoding and AviSynth frame synchronization. The execution environment was a single Java virtual machine processing thread on a 64-bit Windows 7 system and Intel Core i5-4690 CPU with 8 GB of RAM.

The SSIM techniques were arranged in 13 combinations in order to isolate the contribution of each technique, and given mnemonics, as described in Table 5.1. (Note the "Fast SSIM" in this table is improved by the 7×7 Gaussian integer approximation proposed in this work.) For these purposes, luma similarity is disconsidered.

5.1.2. Results and discussion

Table 5.2 gives the experimental results, sorted by RCC. High RCC and LCC values are best, whereas the best values for RMSE and time are the lowest. The emphasized time and efficiency results are the best results for all data. Efficiency is defined by (13). The results for each VQI and video file are given in Appendix B; Figs. 5.1 and 5.2 show the scatter plots for the predicted DMOS data predicted by the most representative of these indexes, produced by nonlinear regression by (12). The tests were performed on 32-bit Java and FFmpeg over Windows. Running on 64-bit and Linux improves performance by up to 30 times, but the values become less distinguishable for algorithm complexity comparison purposes.

$$Efficiency = \frac{RCC \times LCC}{RMSE \times Time} \quad (13)$$

The **shifted gradient (SG)** feature enhancement proposed by this work achieves the *greatest correlation to subjective quality* represented by DMOS among the tested dataset, as well as *most of the greatest computing speeds and efficiencies*. The full-scale SG-Sim bests the predicting ability of SSIM, Fast SSIM, 3-SSIM, and GMSD, and performs statistically equal to MS-SSIM, while computing 69% faster. **Multi-scale SG-Sim** (4S and 5S) achieves the *highest DMOS correlation of all metrics*, which is considerably higher than the next highest metric, MS-SSIM. Surprisingly, 4-scale SG-Sim outmatches 5-scale SG-Sim not only in speed, but also in quality, despite the loss

of detail from the full scale. Finally, **Fast MS-SG-Sim** is 439% faster than MS-SSIM and 11% faster than GMSD.

Gaussian pooling is confirmed to consistently improve the indexes relative to downsampling pooling, though decreasing speed and efficiency. **Downsampling pooling** fulfills its design, sacrificing a subtle degree of correlation to DMOS in exchange for a significant increase in speed: **103% faster** in full scale and 59% faster with 4 scales.

Table 5.2 — Quality and efficiency of SSIM-based indexes over the mobile VQA dataset.

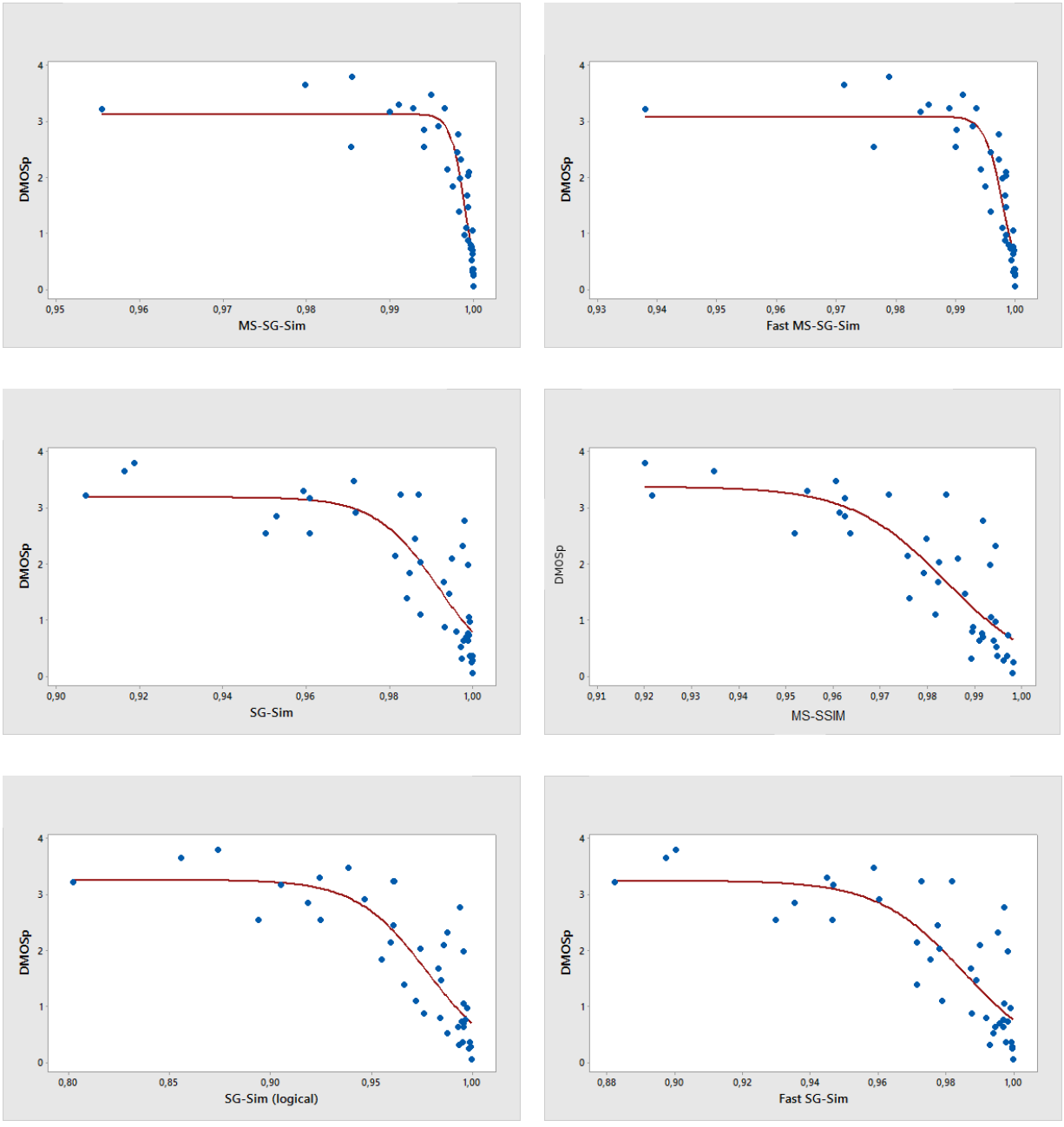
Visual quality index	RCC	LCC	RMSE	Time (s)	Time (normalized)	Efficiency
4S-SG-Sim	0.935	0.925	0.434	**62**	**0.41**	**4.82**
5S-SG-Sim	0.934	0.921	0.447	187	1.27	1.54
Fast MS-SG-Sim	0.929	0.915	0.461	**39**	**0.26**	**7.09**
SG-Sim	0.843	0.832	0.633	124	0.83	1.34
MS-SSIM	0.840	0.839	0.619	210	1.40	0.81
SG-Sim (Roberts)	0.838	0.813	0.663	103	0.69	1.46
SG-Sim (logical)	0.832	0.839	0.619	127	0.85	1.33
SG-Sim (Roberts, logical)	0.823	0.812	0.665	103	0.69	1.46
Fast SG-Sim	0.810	0.816	0.668	**61**	**0.41**	**2.47**
Fast SSIM	0.807	0.803	0.679	102	0.68	1.40
GMSD	0.782	0.804	0.678	**55**	**0.37**	**2.53**
3-SSIM	0.731	0.761	0.739	222	1.48	0.51
SSIM	0.708	0.743	0.763	150	1.00	0.69

Although the SG-Sim versions based on the Prewitt operator achieve the best quality, the Roberts operator achieves greater efficiency (20% faster) at marginally lower quality. As for similarity stabilization, the **logical treatment** achieves **equal speed and efficiency** to the addition of constants for SG-Sim (with both the Prewitt and Roberts operators), although the RCC, LCC and RMSE produced are different. **Arithmetic stabilization** achieves the highest RCC, but the product of RCC, LCC and RMSE between the two methods is equal, so the advantage may be considered

inconclusive.

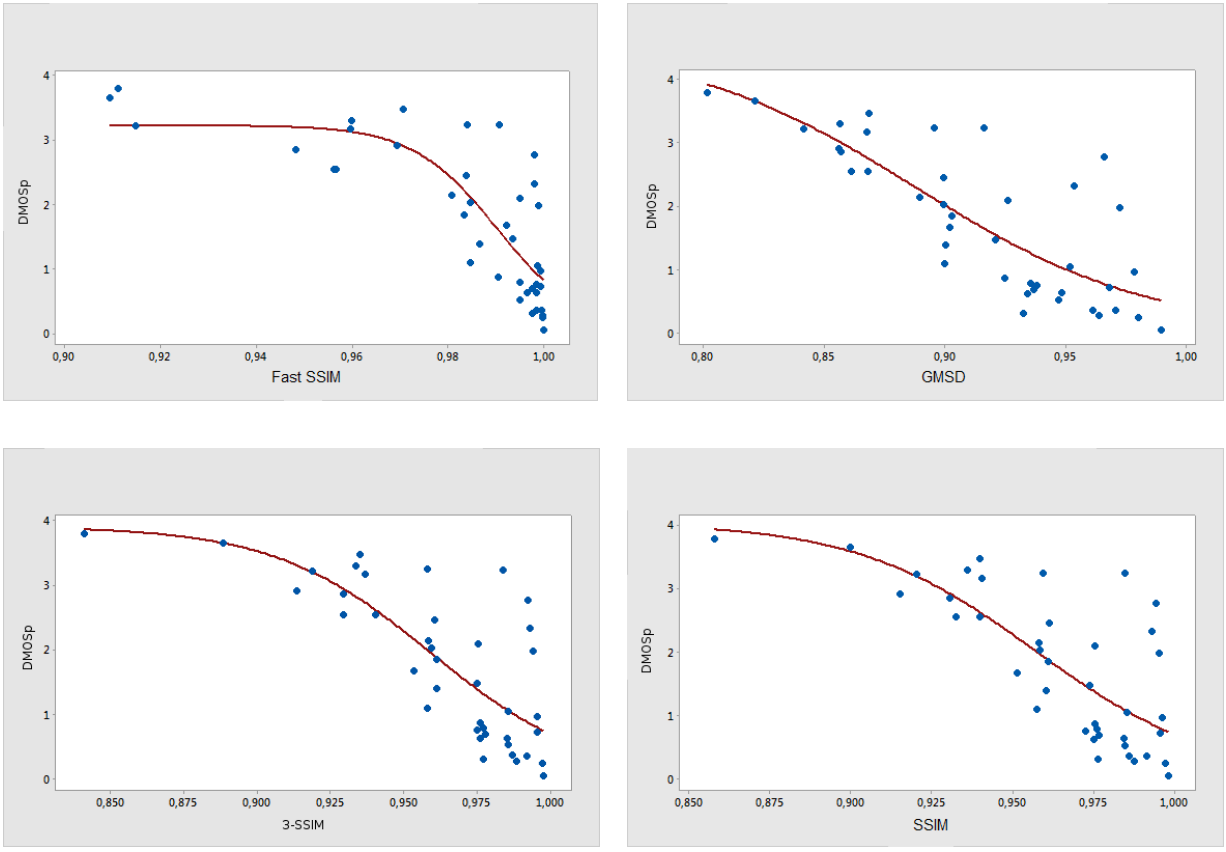
It is interesting to note that Xue et al. (2014) show that GMSD is a better general-purpose image quality index than MS-SSIM and 11 other metrics, which is *inconsistent* with the above results for VQA related to encoding. This suggests that the nature of the testing materials of that paper is significantly different and merit investigation.

Fig. 5.1 — Scatter plots of SSIM-based indexes fitted to predicted DMOS.



Source: the Author.

Fig. 5.2 — Scatter plots of SSIM-based indexes fitted to predicted DMOS.



Source: the Author.

Although the SG-Sim versions based on the Prewitt operator achieve the best quality, the Roberts operator achieves greater efficiency (20% faster) at marginally lower quality. As for similarity stabilization, the **logical treatment** achieves **equal speed and efficiency** to the addition of constants for SG-Sim (with both the Prewitt and Roberts operators), although the RCC, LCC and RMSE produced are different. **Arithmetic stabilization** achieves the highest RCC, but the product of RCC, LCC and RMSE between the two methods is equal, so the advantage may be considered inconclusive.

It is interesting to note that Xue et al. (2014) show that GMSD is a better general-purpose image quality index than MS-SSIM and 11 other metrics, which is *inconsistent* with the above results for VQA related to encoding. This suggests that the nature of the testing materials of that paper is significantly different and merit investigation.

5.1.3. Complementary image quality assessment experiments

It is interesting to determine SG-Sim's effectiveness as a *general-purpose image quality metric*, beyond simply a video encoding quality metric. Since SG-Sim does not account for temporal information, it is actually an image quality metric before a video quality metric; however, it has been developed for evaluating encoding quality, not random distortions and noise, so it must be specifically tested for image quality assessment (IQA).

Xue et al. (2014) present **GMSD** as an IQA metric that achieves higher quality than *12 other metrics* (the best of which being the Information-Weighted SSIM and the Feature Similarity); however, **SG-Sim** has been shown to achieve *higher quality* than GMSD on the *LIVE Mobile VQA database*. This suggests SG-Sim may outperform all those other metrics as well. Nevertheless, the experiments are conducted in different datasets, so this hypothesis requires specific verification.

GMSD was tested (Xue et al., 2014) against the *LIVE IQA database* (779 distortions) (SHEIKH; SABIR; BOVIK, 2006), the *Categorical Subjective Image Quality (CSIQ) database* (886 distortions) (LARSON; CHANDLER, 2010), and the *Tampere Image Database (TID2008)* (1,700 distortions) (PONOMARENKO et al., 2009). The present results against the LIVE Mobile VQA database are, in fact, *inconsistent* with Xue et al. (2014), since the former rank MS-SSIM higher than GMSD, while the latter rank GMSD highest. Thus, it is necessary to test SG-Sim against those additional datasets in order to transitively verify its potential quality advantage over the other metrics. Also noteworthy is that SSIM-related papers, including GMSD, use *different logistic functions* for nonlinear regression than (12) from VQEG (2003), for reasons undisclosed, which may or may not account for inconsistencies in results.

TID2008 is the largest recent public database for IQA, with 25 reference images, 17 types of distortions for each reference image, and 4 levels of each type of distortion. This amounts to a larger dataset than the LIVE IQA and CSIQ databases combined, so in the results of Xue et al. (2014), it heavily biases the overall weighted results for the three databases, which are consistent with their results for TID2008 in 10 out of 13 metrics, including the three best and the four worst. For this reason,

testing SG-Sim against the LIVE IQA and CSIQ databases in addition to TID2008 would probably be redundant, and TID2008 is taken as sufficient testing data. As recommended by Ponomarenko et al. (2009), the Spearman and Kendall rank correlation coefficients are computed for the visual quality metrics of interest, and given in Table 5.3.

Table 5.3 — Rank correlation coefficients between representative SSIM-based metrics and TID2008.

Visual quality index	Spearman RCC	Kendall RCC	Time (s)	Time (relative)	Efficiency
5S-SG-Sim	**0.853**	**0.648**	205	1.10	0.50
GMSD (shifted)	**0.845**	**0.660**	118	0.63	**0.88**
GMSD	**0.842**	0.642	**95**	**0.51**	**1.06**
Fast MS-SG-Sim	0.827	0.621	**80**	**0.43**	**1.19**
4S-SG-Sim	0.824	0.618	**102**	**0.55**	**0.93**
MS-SSIM	0.815	0.624	224	1.20	0.47
SG-Sim	0.812	0.604	153	0.82	0.60
SG-Sim (Euclidean)	0.806	0.537	168	0.90	0.53
SG-Sim (Sobel)	0.805	0.597	150	0.81	0.60
Fast SG-Sim	0.799	0.591	**98**	**0.53**	**0.90**
SG-Sim (logical)	0.766	0.560	152	0.82	0.48
SG-Sim (unfiltered)	0.703	**0.652**	**90**	**0.48**	0.75
3-SSIM	0.696	0.500	191	1.03	0.34
SG-Sim (Roberts)	0.695	0.502	144	0.77	0.45
SSIM (3σ)	0.682	0.491	196	1.05	0.32
Fast SSIM	0.678	0.486	131	0.70	0.47
SSIM (2σ)	0.678	0.488	186	1.00	0.33
SSIM (unfiltered)	0.656	0.464	**57**	**0.31**	**0.99**

The VQI rankings for TID2008 are **largely consistent** with those for the Mobile VQA Database, with a few exceptions: 5-scale SG-Sim ranks as the highest quality, instead of 4-scale, and GMSD is the second highest; also, MS-SSIM ranks higher than

SG-Sim. Surprisingly, Bovik's *gradient magnitude approximation* (Chen and Bovik, 2010) ranks higher than the precise Euclidean magnitude; the **Prewitt gradient operator** ranks higher than the Sobel operator; and the Roberts operator ranks significantly lower than the other two. The **2σ Gaussian pooling** is *confirmed* statistically equivalent to the 3σ Gaussian pooling. **Unfiltered SSIM** achieves 96.8% similarity to Gaussian-filtered SSIM, and is 3 times faster and one of the most efficient indexes for this dataset. Unfiltered SG-Sim, on the other hand, is *insignificantly* faster than **Fast SG-Sim** and achieves significantly lower DMOS correlation and efficiency. Finally, the **luma similarity** component is found *detrimental* to overall DMOS prediction, as evidenced by the results in Table 5.4.

Table 5.4 — Rank correlation coefficients between representative SSIM-based metrics with luma similarity and TID2008.

Visual quality index	RCC without luma	RCC with luma
5S-SG-Sim	0.853	0.772
MS-SSIM	0.815	0.798
SG-Sim	0.812	0.657
SG-Sim (logical)	0.766	0.645
3-SSIM	0.696	0.673
SSIM	0.682	0.653
Fast SSIM	0.678	0.646

Concluding the MOS/DMOS correlation tests, the **Multi-scale Shifted Gradient Similarity** index based on the Prewitt operator and arithmetic stabilization is once more *confirmed* the most effective visual quality index tested in this work, both for video and static images, and with downsampling pooling is also the most efficient. As 5-Scale SG-Sim outperforms both GMSD and MS-SSIM, it may be transitively predicted to also outperform the other 11 metrics bested by GMSD in Xue et al. (2014). The produced index data sets are considerably long texts, and are made available in the package at http://estevao.altervista.org/ufpe/jvqa-tid2008.zip.

5.2. Similarity index behavior and scalability

The numeric behavior of the similarity index must be adequately understood to draw significant comparisons. Encoded video versions have a tendency to be in the 0.8 – 1.0 range even when badly distorted by excessively low data rates. Clearly, the index is **not** to be considered as **a linear percentage**, but probably close to a logarithmic function. For example, the worse SSIM index of the 40 distorted videos in the Mobile VQA dataset is 0.8578, whereas the best is 0.9981 and the respective DMOS are 3.7938 and 0.0500 (see Appendix B). So the index values are not linearly proportional to the quality loss, and their meanings are not *immediately intuitive*.

Table 5.5 — Statistics for representative SSIM-based indexes for the Mobile VQA dataset.

VQA Index	Lowest	Lowest dB	Highest	Highest dB	Mean	Standard deviation	Amplitude
SG-Sim, unfiltered	0.6731	4.86	0.9864	18.66	0.8457	0.0748	0.3133
GMSD	0.8016	7.02	0.9895	19.79	0.9151	0.0465	0.1879
SG-Sim (logical)	0.8321	7.75	0.9999	40.00	0.9689	0.0397	0.1677
3-SSIM	0.8408	7.98	0.9978	26.58	0.9630	0.0327	0.1570
SSIM	0.8578	8.47	0.9981	27.21	0.9639	0.0301	0.1403
MS-SSIM	0.9201	10.97	0.9983	27.70	0.9790	0.0200	0.0782

Converting a SSIM-based index to decibels (DAEDE; MOFFITT, 2015) by (14) as in the `x264` encoder[44] (an adaptation from the decibel form of MSE, which is the PSNR metric) may be convenient to better distinguish between values that are very close and to understand variations in quality, because an SSIM precision to the fourth decimal corresponds to the first decimal in decibels. The decibels derived from the indexes are included in the output of `jVQA`.

$$dB = -10\log_{10}(1 - SSIM) \quad (14)$$

Different SSIM versions exhibit different index amplitudes, which correlate with their standard deviations, as shown in Table 5.5. While this property has no

44 <http://mailman.videolan.org/pipermail/x264-devel/2010-June/007391.html>.

relation with DMOS, it is convenient when comparing very similar video versions.

SSIM-based indexes are *not consistent* across different *resolutions*, not even adjusting the feature pooling window size proportionally. This behavior presents a **scalability** problem. Table 5.6 gives statistics for compressed scene from Blu-Ray, a typical high-action scene with film grain noise that has been encoded from 25 Mbit/s to 2 Mbit/s and down-scaled to common device display resolutions. This dataset illustrates the index *flattening* at increasingly lower resolutions, as feature coherency rises. The indexes may be normalized relative to a reference resolution by exponentiation, as in MS-SSIM. For example, 0.993785 to the power of 6.44 equals 0.960646, a reasonable approximation to 0.960667. Such should be a function of the ratios between the root of the total sampling pixels for each resolution, the **pixel ratio root (PRR)**. However, the various content investigated produce a wide variation of exponents, with no consistent rule yet identified.

Table 5.6 — SG-Sim index relations across resolutions.

Resolution	SG-Sim	Exponent	PRR	$PRR^{5.018}$
1920×1080	0.960667	1.00	1.00	1.00
1280×720	0.993785	6.44	1.50	7.65
720×404	0.999273	55.20	2.67	138.10
320×180	0.999995	8,030.00	6.00	8,030.88

5.3. SSIM and SG-Sim correlation to rate-distortion optimization of encoders

Most digital video is distributed in lossy compressed formats, such as H.264. Video encoders are highly efficient in compressing visual data and selecting the least relevant visual data for disposing at low data rates, using **rate-distortion optimization (RDO)** techniques. Historically, most RDO decisions were based on the *peak signal-to-noise ratio (PSNR)*, but recent encoders such as `x264` (MERRITT, 2006) and `libvpx`[45] acknowledge SSIM as a more effective quality metric. State-of-the-art encoders, such as `x264` and `x265`, take a step further by implementing techniques

45 <http://www.webmproject.org/about/faq>.

empirically found to improve subjective quality beyond that produced by SSIM-based decisions. Such techniques in those two particular encoders are called **Psychovisual RDO (Psy-RDO)** (SHIKARI, 2008a; SHIKARI, 2009a; WAGGONER, 2010; PATTERSON, 2012; AVIDEMUX, 2012; MEWIKI, 2012). Thus, the encoders offer three *RDO modes: PSNR*, *SSIM* and *Psy*. After encoding, they output the final PSNR and SSIM indexes achieved for the content, in which case Psy-encoded content will naturally score lower by lack of an adequate VQA metric to evaluate Psy-RDO. Considering **SG-Sim** has already been proven to better correlate with DMOS than many other SSIM-based VQA metrics, it is expected to also better correlate with innovative RDO modes such as Psy than with the dated PSNR and SSIM modes, as well as provide a valuable tool for RDO implementations.

Video content transmitted to consumers by popular services such as Netflix and Youtube commonly falls within one of the following types: clean natural scenes, natural scenes with film grain noise, or classic animation (note that computer-generated animation may be classified as clean natural scenes for VQA purposes). Particularly, classic animation presents an encoding challenge due to the predominance of "hard" outlines that are especially difficult for DCT-based encoding. Additionally, content with high motion, i.e., high changes in temporal information, which includes "hard" scene cuts, generally require more bits to encode within a certain quality level than low motion scenes, particularly the so-called "talking head" scenes with out-of-focus background. Therefore, representative content classes may be proposed as:

- a high-motion natural scene with film grain ("**cinema**");
- a high-motion clean animation scene ("**anime**");
- a low-motion natural scene with out-of-focus background ("**interview**").

The LIVE Mobile VQA dataset is constituted entirely of the clean natural scenes type, so other content sources must be considered. Excerpts from Blu-ray commercial titles in 1080p H.264 format provide typical samples of each content class. For each class, **five excerpts** 14 to 18 seconds in duration, according to the respective scene intervals, were extracted and encoded in `x264` with configurations appropriate for adaptive streaming, as detailed below.

According to StatCounter (2015), 71% of global Web browsers in 2015 had displays of 1280×720 **(HD 720p)** and higher, while displays of 1920×1080 and higher accounted for only 13%. 720p is, in fact, a broadly supported resolution for computers, video game consoles, television sets, tablets and smartphones. Although the availability of 1920×1080 (1080p) and 3840×2160 (4K) displays is rising, consideration of mobile devices suggests that 720p is currently the most representative of Web display resolutions. This is convenient for testing content, because the resolution is not as high as to require an inordinate amount of time to compute, while not as small as to lose significant detail for visual quality.

5.3.1. Essential encoder configuration for experimentation

In adaptive streaming systems, content is segmented to periods of 2 to 4 seconds to enable fast quality adaptation (ZAMBELLI, 2009; LEVKOV, 2010, 2011; STOCKHAMMER, 2011; APPLE, 2014). For experimental purposes, this is equivalent to fixing the group of pictures (GOP) to such period. The tested Blu-ray content has 23.976 frames per second, so the period is fixed at 96 frames by the parameter **`keyint`**, which sets the maximum interval for key frames (intra-coded or I-frames). In the specific case of H.264, predicted and bi-predictive frames (P- and B-frames) may refer to frames outside their own I-frame intervals, but not outside their instantaneous decoder refresh (IDR) frames interval, so in adaptive streaming all I-frames must be IDR-frames by setting the parameter **`min-keyint`** = `1` (AVIDEMUX, 2012).

Typical Web streaming services allow a delay of 1.5 second for video buffer verification (VBV). This facilitates taking advantage of variable data rate encoding (VBR mode), which generally improves quality as bits may be saved from frames of low visual complexity for improving frames of higher complexity. The maximum data rate must be restricted as to not cause a buffer underrun. A maximum data rate **`(vbv-maxrate)`** of twice the average data rate is recommended, with a buffer size **`(vbv-bufsize)`** of thrice the average (PATTERSON, 2012). A slightly-below-average data rate for 720p in the Web is 1,000 kbit/s, with VBR limited to 2,000

kbit/s.

As mentioned above, x264 provides a **`tune`** setting that calibrates the encoding to appropriate content types ("film", "grain" and "animation") or to maximize VQA scores ("SSIM" and "PSNR") (SHIKARI, 2009a, 2009b, 2010b). This investigation is especially interested in how SSIM and SG-Sim evaluate content encoded with each of these settings.

The "SSIM" and "PSNR" modes both disable the psychovisual optimizations; the difference between these modes is that "PSNR" mode disables **adaptive quantization (AQ)**, `aq-mode`, a technique for avoiding blocking and banding artifacts in low detail regions of the video frames (SHIKARI, 2008b; AVIDEMUX, 2012). The default setting for `aq-mode`, also used in Psy RDO, is a basic variance-based AQ (VAQ), also called a complexity mask, for each individual frame; with `tune` = SSIM, however, `aq-mode` is set so that VAQ is automatically biased frame-by-frame, which improves the mean SSIM index for the video sequence.

Psychovisual optimizations in `x264` are enabled by default (unless overridden by SSIM or PSNR modes) and are adjusted with the `tune` values "film", "grain" and "animation" (along with other adjusted settings) by the `psy-rd` setting, which defaults to "1.0:0.0". The first part of the value of `psy-rd` is the strength for psychovisual RDO, which biases encoding quantization, *i.e.*, the bit allocation across coding blocks of the video frame, towards preservation of visual energy (high-frequency information) at each block, adjusting for less energy (thus, blurring) in high-movement frames (AVIDEMUX, 2012; PATTERSON, 2012), and defaults to 1.0; `tune` = "animation" lowers this value to 0.4, because classic animation includes less high frequency information than natural video. The second value in `psy-rd` is the strength for trellis RDO, a technique that optimizes quantization by accounting for the actual final size of the bit stream after entropy coding (AVIDEMUX, 2012; MERRITT, 2005; PATTERSON, 2012) and defaults to 0.0 (disabled); `tune` = "film" raises this to 0.15 and "grain" raises it to 0.25. Other relevant settings affected by Psy RDO are the strength of adaptive quantization, which is reduced with "animation" and "grain", and the strength of the in-loop deblocking filter, reduced with "film" and "grain", but increased with "animation". Many other settings are also affected

by these tunings, beyond the scope of this discussion.

5.3.2. Encoder configuration for compatibility with Web decoders

An essential concern to Web streaming is broad client compatibility. This requirement imposes restrictions related to decoder performance and standard support levels. Because the tested content is in 720p resolution, which require medium to high-end decoders, it is generally safe to support the High Profile for H.264 (WAGGONER, 2010). This format standard also allows calibration of several techniques that improve quality while demanding more system resources, which may result in decoding performance loss unless properly configured.

The most relevant compatibility setting is the number of frames stored in memory for referencing by predictive frames. More reference frames results in more efficient compression, thus higher quality, but more system resources are required and decoding performance eventually suffers. Waggoner (2010), Patterson (2010), Levkov (2011) and x264 developers propose to limit to **`ref`** = 4, also accounting for diminishing returns; this is also consistent with most H.264 support levels. Because this and other settings not only affect performance but also quality, it is important to consider when producing content for testing so the experiments are conducted under realistic conditions.

Another restriction for decoder compatibility is the number of consecutive B-frames allowed during coding. B-frames are the most efficient type of coded pictures for compression, but also increase error propagation, and increasing this parameter may cause decoding errors in some devices. Waggoner (2010), Levkov (2011), Ozer and Youtube (Patterson, 2012) recommend limiting **`bframes`** = 2. The default value is 3 and diminishing returns become considerable when greater than 4.

The final decoder compatibility parameter to observe is **`b-pyramid`**, which allows B-frames to be referenced by other frames, forming pyramid-like references. Many decoders do not support this option of the H.264 specification because the Blu-ray specification does not require it, so broad compatibility requires disabling it (PATTERSON, 2012).

5.3.3. Encoder configuration for improved quality at low data rates

MPEG video formats such as H.264 are asymmetric regarding the coding and decoding algorithms, because coding involves more effort than decoding (WAGGONER, 2010). For instance, motion estimation requires the encoder to perform wide searches in both spatial and temporal dimensions, the results of which are relatively simple to decode. Due to the significantly low data rates in Web streaming, when encoding video on demand content, it is advisable to invest greater encoding effort in order to produce better quality. There are many parameters in `x264` to increase encoding quality at the expense of encoding performance but no additional expense in decoding performance (PATTERSON, 2012; AVIDEMUX, 2012):

- multipass rate control **`(pass)`** may be employed for two encoding passes so the entire video is analyzed before the final encoding, allowing informed encoding decisions; an additional benefit is that all other parameters below only apply to either the first or the last pass, further increasing efficiency;
- direct motion vector prediction mode **`(direct)`** – 1st pass only – may be increased from "spatial" to "auto" (adaptive selection between "spatial" and "temporal");
- adaptive B-frame decision method **`(b-adapt)`** – 1st pass only – may be increased from 1 ("fast") to 2 ("optimal"), with low cost with only 2 consecutive B-frames;
- number of frames for frame type lookahead **`(rc-lookahead)`** – 1st pass only – may be increased from 40 to 60 (this setting is disabled in "zero latency" mode);
- subpixel motion estimation and mode decision **`(subme)`** – final pass only – may be increased from 7 (RDO enabled for all frames) to 10 (quarter-pixel RDO refinement for all frames);
- trellis rate-distortion quantization **`(trellis)`** – final pass only – may be increased from 1 (enabled only on the final encoding of a macroblock) to 2 (enabled on all mode decisions: motion vector refinement, macroblock

partitioning and quantization);

- integer pixel motion estimation method **`(me)`** – final pass only – may be increased from "hex" (hexagonal search of radius 2) to "umh" (uneven multi-hexagon search); and
- maximum motion vector search range **`(merange)`** – final pass only – may may be increased from 16 to 24.

The proposed improved encoding parameters are similar to the x264 speed `preset` "veryslow", the second slowest and highest quality, though the decoding constraints mitigate some of the decrease in encoding performance for the full preset, and the `partitions` setting remains at the default value, which is more efficient.

5.3.4. Experimental methodology

Five excerpts from 1080p Blu-ray titles for each content class, "cinema", "anime" and "interview", were encoded with x264 at 720p resolution to three RDO versions, Psy-RDO, SSIM and PSNR, for a total of 45 distorted videos for testing correlation with SG-Sim and other SSIM-based indexes. For each content class, the Psy-RDO version is encoded with the specific tuning appropriate for maximizing perceptual quality for the content: "grain" for "cinema", "animation" for "anime" and "film" for "interview". The average data rates for "cinema" and "film" are 1,000 kbit/s, whereas "animation" was set to 2,000 kbit/s because perceptual quality for 1,000 kbit/s was significantly worse compared to the other two content classes.

The primary objective of the test is to determine the **ranking** of the three RDO versions for each visual quality index (VQI) of interest and each video sequence. Most SSIM-based indexes, particularly the versions based in covariance, are expected to rank the SSIM version as highest. The Psy-RDO implementations in x264 have been tested and calibrated by video coding communities in the Web and are regarded as producing higher perceptual quality than classic SSIM or MSE-based RDO (WAGGONER, 2010). Given that SG-Sim shows higher correlation with perceptual quality represented by DMOS than SSIM, if the Psy versions are ranked highest by SG-Sim, this may be considered additional proof of the effectiveness of the

Psy-RDO in `x264`, and SG-Sim would be an ideal VQA metric for the content produced by this encoder, as well as potentially any other video encoder. In this sense, correlation between Psy-RDO and SG-Sim would reinforce the relevance of Psy-RDO regarding perceptual quality, and of SG-Sim regarding applicability to state-of-the-art video encoding. In this experiment, SG-Sim is computed with the Roberts operator and logical stabilization, instead of with the Prewitt operator and arithmetic stabilization.

5.3.5. Results and discussion

Tables 5.7 to 5.9 give results for VQI and their respective decibel representations, as well as the amplitude ΔVQI and the computing time in seconds (only in the first table, as the others would be redundant for comparison of metrics), as performed by `jVQA`. As discussed in Section 5.2, converting SSIM-based indexes to decibels facilitates comparisons of very close values. The best indexes, decibel values and amplitudes are emphasized in the tables, with values within 0.1 decibels considered statistically equal in this dataset.

The results are mostly consistent with expectations: 28 SSIM results of 30 favor SSIM RDO over Psy RDO; and **all 30 SG-Sim results favor Psy RDO**. This *confirms* the correlation between SG-Sim and the Psy RDO of `x264`, as well as the allegation of this encoder's developers and users that better perceptual quality is produced by that mode, since SG-Sim has the best DMOS correlation among the tested methods. It is interesting to note that the `jVQA` decibel results indicate statistical equality between the SSIM and PSNR RDO modes in `x264`; differences appear only at the second decimal, which is negligible for most practical purposes.

The SSIM behavior for the "anime" sequence is also notable, giving statistical equivalence between the three RDO modes for most excerpts, or all of them if the equivalence criteria is relaxed to 0.5 points, which would also indicate equivalence of SSIM for all RDO modes for the "interview" sequence. However, because of the logarithmic nature of decibels and the tendency of SG-Sim to produce higher values than SSIM for this dataset, such equivalence may be simply a matter of the higher

scale of values resulting for that content. Perhaps, for this dataset, a stricter rule for equivalency would be more mathematically appropriate. In any case, 20 SSIM results of 30 favor SSIM over PSNR, and PSNR over Psy, whereas the other 10 results favor PSNR over SSIM, and SSIM over Psy, which are consistent with the expectation that SSIM ranks Psy RDO worse than the SSIM and PSNR modes.

Table 5.7 — Visual quality index (VQI) results and computing times for video sequence "cinema".

	Psy (Grain) RDO		**SSIM RDO**		**PSNR RDO**			
VQA method	**VQI**	**dB**	**VQI**	**dB**	**VQI**	**dB**	**ΔVQI**	**Time (s)**
Cinema excerpt #1								
SSIM	0.993492	21.9	0.993742	**22.0**	**0.993755**	**22.0**	0.00026	142
MS-SSIM (4 scales)	0.997354	25.8	**0.997694**	**26.4**	0.997690	**26.4**	0.00034	67
MS-SG-Sim	**0.999045**	**30.2**	0.996478	24.5	0.996655	24.8	0.00257	52
Fast MS-SG-Sim	**0.995492**	**23.5**	0.988830	19.5	0.989303	19.7	**0.00666**	30
Cinema excerpt #2								
SSIM	0.991529	20.7	**0.992677**	**21.4**	0.992598	**21.3**	0.00115	139
MS-SSIM (4 scales)	0.996175	24.2	**0.996861**	**25.0**	0.996829	**25.0**	0.00069	65
MS-SG-Sim	**0.999509**	**33.1**	0.998396	28.0	0.998486	28.2	0.00111	50
Fast MS-SG-Sim	**0.997420**	**25.9**	0.994046	22.3	0.994331	22.5	**0.00337**	31
Cinema excerpt #3								
SSIM	0.994183	22.4	**0.994600**	**22.7**	0.994582	**22.7**	0.00042	125
MS-SSIM (4 scales)	0.997552	26.1	**0.997925**	**26.8**	0.997923	**26.8**	0.00037	59
MS-SG-Sim	**0.999305**	**31.6**	0.997406	25.9	0.997504	26.0	0.00190	45
Fast MS-SG-Sim	**0.997326**	**25.7**	0.992646	21.3	0.992870	21.5	**0.00468**	28
Cinema excerpt #4								
SSIM	0.991789	20.9	**0.992860**	**21.5**	0.992843	**21.5**	0.00107	138
MS-SSIM (4 scales)	0.996857	25.0	0.997432	**25.9**	**0.997438**	**25.9**	0.00058	65
MS-SG-Sim	**0.999875**	**39.0**	0.999505	33.1	0.999554	33.5	0.00037	49
Fast MS-SG-Sim	**0.998088**	**27.2**	0.996273	24.3	0.996440	24.5	**0.00182**	31
Cinema excerpt #5								
SSIM	0.991379	20.6	0.992478	**21.2**	**0.992479**	**21.2**	0.00110	167
MS-SSIM (4 scales)	0.995921	23.9	0.996626	**24.7**	**0.996668**	**24.8**	0.00075	78
MS-SG-Sim	**0.999624**	**34.2**	0.998860	29.4	0.998949	29.8	0.00076	59
Fast MS-SG-Sim	**0.996988**	**25.2**	0.994576	22.7	0.994823	22.9	**0.00241**	37

The *amplitudes* are clearly directly affected by different VQA techniques: *multi-scaling compresses* the indexes, reducing amplitudes, whereas feature pooling by

downsampling expands the indexes, increasing amplitudes. DMOS correlation analysis in 5.1.2 indicates that multi-scaling increases quality performance by a wider margin than downsampling pooling decreases it, and those two techniques combined with SG-Sim produce the greatest amplitudes for the RDO dataset, which is convenient for comparisons of content of high similarity.

Table 5.8 — Visual quality index (VQI) results for video sequence "interview".

VQA method	**Psy (Film) RDO**		**SSIM RDO**		**PSNR RDO**		
	VQI	**dB**	**VQI**	**dB**	**VQI**	**dB**	**ΔVQI**
Interview excerpt #1							
SSIM	0.998240	27.5	**0.998420**	**28.0**	0.998417	**28.0**	0.00018
MS-SSIM (4 scales)	0.999345	31.8	**0.999386**	**32.1**	**0.999386**	**32.1**	0.00004
MS-SG-Sim	**0.999999**	**63.1**	0.999995	53.0	0.999997	55.1	0.00000
Fast MS-SG-Sim	**0.999991**	**50.5**	0.999938	42.1	0.999959	43.9	**0.00005**
Interview excerpt #2							
SSIM	0.998712	28.9	**0.998842**	**29.4**	0.998830	**29.3**	0.00013
MS-SSIM (4 scales)	0.999488	32.9	**0.999516**	**33.2**	0.999515	**33.2**	0.00003
MS-SG-Sim	**0.999999**	**66.9**	0.999998	57.3	0.999999	59.4	0.00000
Fast MS-SG-Sim	**0.999994**	**52.2**	0.999973	45.6	0.999980	46.9	**0.00002**
Interview excerpt #3							
SSIM	0.997242	25.6	0.997557	**26.1**	**0.997565**	**26.1**	0.00032
MS-SSIM (4 scales)	0.999146	30.7	0.999221	**31.1**	**0.999226**	**31.1**	0.00008
MS-SG-Sim	**0.999995**	**52.7**	0.999895	39.8	0.999927	41.3	0.00010
Fast MS-SG-Sim	**0.999935**	**41.9**	0.999367	32.0	0.999499	33.0	**0.00057**
Interview excerpt #4							
SSIM	0.997394	25.8	**0.997699**	**26.4**	**0.997699**	**26.4**	0.00031
MS-SSIM (4 scales)	0.999173	30.8	0.999247	**31.2**	**0.999251**	**31.3**	0.00008
MS-SG-Sim	**0.999996**	**54.5**	0.999884	39.4	0.999932	41.7	0.00011
Fast MS-SG-Sim	**0.999953**	**43.3**	0.999493	33.0	0.999628	34.3	**0.00046**
Interview excerpt #5							
SSIM	0.998168	27.4	**0.998323**	**27.8**	0.998307	**27.7**	0.00015
MS-SSIM (4 scales)	0.999187	30.9	**0.999238**	**31.2**	0.999235	**31.2**	0.00005
MS-SG-Sim	**0.999999**	**59.2**	0.999992	50.8	0.999994	52.0	0.00001
Fast MS-SG-Sim	**0.999948**	**42.8**	0.999667	34.8	0.999734	35.8	**0.00028**

Table 5.9 — Visual quality index (VQI) results for video sequence "anime".

VQA method	Psy (Film) RDO		SSIM RDO		PSNR RDO		
	VQI	dB	VQI	dB	VQI	dB	ΔVQI
Anime excerpt #1							
SSIM	**0.991601**	**20.8**	0.986704	18.8	0.986639	18.7	0.00496
MS-SSIM (4 scales)	**0.995996**	**24.0**	0.992221	21.1	0.992382	21.2	0.00377
MS-SG-Sim	**0.999493**	**33.0**	0.997213	25.6	0.997125	25.4	0.00237
Fast MS-SG-Sim	**0.997300**	**25.7**	0.991881	20.9	0.991830	20.9	**0.00547**
Anime excerpt #2							
SSIM	0.994887	**22.9**	**0.995012**	**23.0**	0.995008	**23.0**	0.00013
MS-SSIM (4 scales)	0.997553	26.1	0.997675	**26.3**	**0.997682**	**26.4**	0.00013
MS-SG-Sim	**0.999729**	**35.7**	0.999553	33.5	0.999650	34.6	0.00018
Fast MS-SG-Sim	**0.997853**	**26.7**	0.996993	25.2	0.997412	25.9	**0.00086**
Anime excerpt #3							
SSIM	0.996653	**24.8**	0.996732	**24.9**	**0.996760**	**24.9**	**0.00011**
MS-SSIM (4 scales)	0.998332	**27.8**	0.998383	**27.9**	**0.998393**	**27.9**	0.00006
MS-SG-Sim	**0.999985**	**48.2**	0.999978	46.7	0.999983	47.7	0.00001
Fast MS-SG-Sim	**0.999771**	**36.4**	0.999658	34.7	0.999746	35.9	**0.00011**
Anime excerpt #4							
SSIM	0.996733	**24.9**	**0.996842**	**25.0**	0.996820	**25.0**	0.00011
MS-SSIM (4 scales)	0.998458	28.1	**0.998544**	**28.4**	0.998536	**28.3**	0.00009
MS-SG-Sim	**0.999980**	**47.0**	0.999934	41.8	0.999954	43.4	0.00005
Fast MS-SG-Sim	**0.999723**	**35.6**	0.999412	32.3	0.999524	33.2	**0.00031**
Anime excerpt #5							
SSIM	0.996565	**24.6**	**0.996632**	**24.7**	0.996581	**24.7**	0.00007
MS-SSIM (4 scales)	0.998308	27.7	**0.998394**	**27.9**	0.998369	**27.9**	0.00009
MS-SG-Sim	**0.999896**	**39.8**	0.999765	36.3	0.999810	37.2	0.00013
Fast MS-SG-Sim	**0.999029**	**30.1**	0.998309	27.7	0.998536	28.3	**0.00072**

5.4. Closing remarks

In this chapter, the behavior of SSIM-based visual quality indexes was studied in detail following the relevant *differential mean opinion score correlation* methodology of ITU-VQEG, by which 12 index configurations are compared in order to identify those techniques most consistent with subjective quality, based on the public LIVE Mobile VQA Database. Index *complexity*, in the form of computing time, was also investigated for such data, and an index *efficiency* score was proposed, weighting DMOS correlation with complexity. 14 such index configurations were further compared against the TID2008 database, with consistent results. *Scalability* concerns were raised and explained, but remain open. Finally, a methodology for *comparison of encoder RDO modes* was proposed and applied to the `x264` encoder, also considering index amplitudes. The "multi-scaled shifted Prewitt gradient similarity index with downsampled pooling and arithmetic stabilization", dubbed **Fast MS-SG-Sim**, proposed by this work, is found the best visual quality index among those investigated in both DMOS prediction efficiency and RDO mode comparison and amplitude specifically regarding compression distortions.

6. CONCLUSIONS

This work has tackled the decades-old problem of objective perceptual video quality assessment (VQA), in the specific context of adaptive streaming in the Web, as described in depth in Chapter 2. A perfect mathematical model of human VQA, however, would be of yet undefined complexity, so practical solutions in actual applications must seek a balance between **quality prediction accuracy** and **computing efficiency**, especially in the case of encoding and streaming video. The Structural Similarity (SSIM) and related indexes applied to VQA, as well as the psychovisual rate-distortion optimization (Psy-RDO) implementation in the `x264` encoder, are interesting cases in which such balance has been achieved with notable effectiveness, yet still offer opportunities for improvement, including such a VQA metric that may adequately qualify the existing improvements of Psy-RDO.

6.1. Contributions

Several SSIM-based VQA metrics, especially Multi-Scale Fast SSIM and Gradient Magnitude Similarity Deviation (GMSD), were analyzed, decomposed and reassembled in order to measure and understand the contribution of each component technique to quality prediction and computing efficiency. The techniques were detailed in Chapter 3, and this approach allowed to develop a mathematical adjustment to the image feature enhancement techniques that improves subjective quality prediction, thereby consolidated into the "**Shifted Gradient Similarity" (SG-Sim)** index. Numerous publications have employed the gradient feature to improve the original SSIM's covariance feature, and the "shifted" gradient adjusts the feature in order to avoid loss of important visual information and improve the proportions between the compared magnitudes.

Chapter 3 further proposed more efficient spatial pooling filters for SSIM-based indexes: the decomposed *1-D Gaussian filter pair* limited to *two standard deviations,* and the *downsampling Box filter* based on the *integral image*, which retain, respectively, 99% and 98% response equivalence, and achieve speed gains of 68% and 382%,

respectively. The downsampling filter also enables broader scalability, particularly for Ultra High Definition content, and is consolidated into the **"Fast SG-Sim"** index version.

In order to conduct large batches of tests over large video files in a practical fashion, and to produce comparable data for both quality and efficiency, the many variations of SSIM were implemented in the **"Video Quality Assessment in Java" (jVQA)** software, introduced in Chapter 4. `JVQA` allows to customize SSIM indexes by assembling from the many implemented component techniques, and was designed for reuse and extensibility through object-oriented design patterns. Furthermore, `jVQA` accepts virtually any type of video file as input by decoding through the AviSynth and FFmpeg *free and open-source (FOSS)* platforms, supports 4K Ultra-HD content, and is itself FOSS under the GNU Public License. Thus, a secondary product of this work is a practical and free alternative to proprietary VQA software such as Matlab, Video Quality Measurement Tool, and SSIMWave Video QoE Monitor[46].

The VQA metrics of interest were assembled and run through `jVQA` on the LIVE Mobile VQA and TID2008 databases, which include several relevant distortions with full references, as well as the respective mean opinion scores (MOS) for *subjective quality verification*. The latter was accomplished by computing the **correlation coefficients** recommended by the Video Quality Experts Group (VQEG), as described in Chapter 5. By this methodology, SG-Sim, Fast SG-Sim and Fast Multi-Scale SG-Sim were found to achieve *notable subjective quality prediction and efficiency*. SG-Sim was shown to surpass the quality of all other metrics for the LIVE database, although it was defeated by GMSD and MS-SSIM for the TID2008 database. Fast SG-Sim was defeated only by MS-SSIM for LIVE, and ranked next to SG-Sim for TID2008. Finally, **Multi-Scale SG-Sim** was the best-performing metric for subjective quality prediction for all data, whereas **Fast MS-SG-Sim** was the second best for LIVE and the third best for TID2008, behind GMSD. For all data, *Fast MS-SG-Sim was the most efficient metric*, followed alternatively by 4-Scale SG-Sim and GMSD, and Fast SG-Sim at the fourth rank.

The shifted gradient feature enhancement was found to perform better by the

46 <http://www.ssimwave.com>.

Prewitt operator than Sobel's and Roberts', and coupled with *arithmetic stabilization* rather than the proposed *logical stabilization*. As for multi-scaling, *4-scaling* was found considerably more efficient than *5-scaling*, and depending on content, may either improve (as in the LIVE Mobile VQA database) or damage (as in the TID2008 database) subjective quality prediction. Finally, not pooling the feature-enhanced image before the similarity term (*unfiltered* SSIM) proved significantly efficient, but resulted in the worse quality prediction. Furthermore, Chapter 5 also provided considerations about the *behavior* of SSIM-based indexes, such as *amplitudes*, *decibel representation*, and *scalability* concerns.

SG-Sim was also confirmed to correlate higher with the Psy-RDO of `x264` than with its other RDO modes, SSIM and PSNR-based. As explained, the former implementation is considered to produce greater perceptual visual quality than the two latter traditional ones. This was verified by applying SG-Sim, SSIM and other metrics to evaluate an original video database consisting of excerpts from commercial Blu-ray titles of three classifications, called "cinema", "anime" and "interview", each compressed by `x264`, at low data rates, to three versions, one for each of the aforementioned RDO modes. Thus, not only may SG-Sim contribute to RDO in general due to its higher predictive ability of subjective quality than SSIM and the mean squared error, it is immediately applicable to more accurately comparing the quality of content encoded by different RDO modes and encoders, such as `x264`, `x265`, and `libvpx`.

In conclusion, this work has produced valuable insights, techniques and implementations of video quality assessment. As a final contribution, Appendix A offers an extensive list of additional recent reading materials regarding VQA metrics, video coding recommendations (especially for H.264 and other formats available with HTML5), MPEG-DASH research, and usage statistics for Web media.

6.2. Limitations and future work

The VQA metrics selected for comparison in this work were limited in complexity in order to *minimize encoding latency* and *maximize efficiency*. Thus, none of these metrics process the *temporal dimension*, which would account for brusque data rate fluctuations that may hurt perceptual quality considerably more than "smooth" variations, contributing to the overall effectiveness of a VQA metric. It should be noted, however, that it is common practice to compare video coding implementations by efficient spatial-only metrics such as PSNR, SSIM, and Fast SSIM (VATOLIN, 2012; DAEDE and MOFFITT, 2015; http://arewecompressedyet.com).

Motion estimation was first coupled with SSIM by Wang, Lu and Bovik in 2004; other metrics of interest that include the temporal dimension are Video Quality Model (VQM) (PINSON; WOLF, 2004), Spatio-Temporal Video SSIM (MOORTHY; BOVIK, 2009a), Motion-based Video Integrity Evaluation (MOVIE) (SESHADRINATHAN; BOVIK, 2009), Motion-Compensated SSIM (MOORTHY; BOVIK, 2010), and SSIMplus (REHMAN; ZENG; WANG, 2015). Further high-performing metrics which do not process temporal information but compute more complex feature enhancements include Complex Wavelet SSIM (WANG; SIMONCELLI, 2005), PSNR-HVS (EGIAZARIAN et al., 2006; PONOMARENKO et al., 2009), Fixation SSIM (MOORTHY; BOVIK, 2009b), Information-Weighted SSIM (WANG; LI, 2011), and Feature Similarity (ZHANG et al., 2011). All these metrics are relevant candidates for implementation in `jVQA` and further comparative testing with SG-Sim.

This work investigated existing RDO implementations and proposed they may be improved by using SG-Sim as a decision metric, instead of SSIM, mean squared error or other metrics. However, such implementation adjustments have not been performed or verified. Also on the topic of RDO, the RDO mode comparison could benefit from more statistical information such as the standard deviation to pair with the mean of the VQA results for each frame, and RDO from different encoders, such as `x265`, `libvpx` and `Daala`[47] (DAEDE; MOFFITT, 2015), could be investigated as

47 <http://wiki.xiph.org/Daala>.

well. However, H.264 remains the most relevant case study as the most ubiquitous digital video format.

The experiments in this work were limited to content in 720p (HD) resolution, and could be expanded to 1080p (Full-HD) and 4K (Ultra-HD). This may allow to better understand the relations between the responses of SSIM-based metrics through different resolutions. Because downsampling is an operation based on a low-pass filter, comparing index responses for different resolutions is equal to comparing content of the same resolution with one of the versions blurred by such a filter, compromising a meaningful comparison between different resolutions unless the higher-resolution version is first appropriately blurred. Then, a universal inter-resolution VQA metric, which produces responses that are coherent and consistent between different resolutions, requires a mathematical model for adjusting the index responses to a fixed reference resolution, perhaps as performed by the SSIMplus index.

The comparative testing of VQA metrics conducted on the LIVE Mobile Video Quality Database was limited to the samples with distortions only due to compression, but the database also includes samples for several other types of distortions: wireless channel packet-loss, frame freezing, rate adaptation, and temporal dynamics. Testing these additional types of distortions may widen the scope of relevance of SG-Sim, although the lack of a temporal component in SG-Sim will certainly limit its effectiveness, except for packet-loss. Another relevant and recent VQA database is the LIVE Video Quality Database (SESHADRINATHAN et al., 2010), based on samples from the Technical University of Munich, although the samples were downsampled from HD to 768×432 in order to reduce the necessary computing resources. It bears notice, as well, that the two databases share three of the authors of each, so some redundancy may be expected, although the source materials are distinct. There are also two relevant databases from VQEG, the dated Standard Definition TV database (2000) and the more recent HD-TV database (2010), both available at <http://www.its.bldrdoc.gov/vqeg/downloads.aspx>; as well as the recent TID2013 (PONOMARENKO et al., 2015).

Lastly, future `jVQA` functionality extensions bear mention, as this software may

become a valuable asset for research on VQA. Improved output and statistics are of primary interest. The frame-by-frame similarity data may be used to produce a full quality graph for the analyzed video sequence; this would be useful to identify distortion spikes and particular "bad" frames, which may also be specifically exported for verification. Further, the quality map output that is currently restricted to individual frames exported as PNG files may be extended to exporting the full sequence of quality maps for the video sequences analyzed. `JVQA` also currently includes implementations of the Spearman rank correlation coefficient, the Pearson linear correlation coefficient, and the root mean squared error, which have yet to be made available in the application's interfaces, and may be extended with a non-linear regression program as well. There also remain frame-accuracy problems with the FFmpeg decoder to solve, and performance optimizations to explore.

REFERENCES

AARON, Anne et al. **Per-title encode optimization**. The Netflix Tech Blog (on-line), Dec. 14, 2015. Available in: <http://techblog.netflix.com/2015/12/per-title-encode-optimization.html>. Accessed on: Dec. 21, 2015.

APPLE. **Best practices for creating and deploying HTTP Live Streaming media for the iPhone and iPad** (Technical Note TN2224). IOS Developer Library (on-line), Feb. 28, 2014. Available in: <http://developer.apple.com/library/ios/technotes/tn2224/_index.html>. Accessed on: June 1, 2014.

AVIDEMUX. **H.264 encoding guide**. Avidemux Wiki Documentation (on-line), Nov. 11, 2012. Available in: <http://www.avidemux.org/admWiki/doku.php?id=tutorial: h.264>. Accessed on: June 1, 2013.

CHEN, G.; YANG, C.; XIE, S. **Gradient-based structural similarity for image quality assessment**. In: ICIP – IEEE International Conference on Image Processing, Oct. 2006, Atlanta. *Proceedings…* p. 2929-2932.

CHEN, M.; BOVIK, A. C. **Fast structural similarity index algorithm**. *Journal of Real-Time Image Processing*, v. 6, Springer, p. 281-287, Dec. 2011.

CISCO. **Cisco Visual Networking Index:** Forecast and Methodology, 2014-2019. Cisco (on-line), May 27, 2015. Available in: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf>. Accessed on: Dec. 21, 2015.

DAEDE, T.; MOFFITT, J. **Video codec testing and quality neasurement**: draft-daede-netvc-testing-02. Mozilla, IETF (on-line), Oct. 19, 2015. Available in: <http://tools.ietf. org/html/draft-daede-netvc-testing-02>. Accessed: Feb. 5, 2016.

EGIAZARIAN, K. et al. **New full-reference quality metrics based on HVS**. In: II INTERNATIONAL WORKSHOP ON VIDEO PROCESSING AND QUALITY METRICS, Scottsdale, USA, 2006. *Proceedings…* 4 p.

GAMMA, E. et al. **Design patterns**: elements of reusable object-oriented software. [S.l]: Addison-Wesley, 1994.

GONZALEZ, R. C.; WOODS, R. E. **Digital image processing**. 3rd edition. [S.l]: Pearson Education, 2007. 976 p.

JOBS, Steven. **Thoughts on Flash**. Apple (on-line), April, 2010. Available in: <http://www.apple.com/hotnews/thoughts-on-flash/>. Accessed on: Dec. 21, 2015.

LARSON, E. C.; CHANDLER, D. M. **Most apparent distortion**: Full-reference image quality assessment and the role of strategy. *Journal of Electronic Imaging*, Bellingham, SPIE, v. 19, n. 1, p. 011006-1–011006-21, Jan. 2010.

LAW, W. **DASHing into an era of convergence**. Akamai.com (on-line), Apr. 17, 2012. Available in: <http://blogs.akamai.com/2012/04/san-francisco-has-a-largely.html>. Accessed on: June 1, 2013.

LEDERER, S. et al. **A seamless Web integration of adaptive HTTP streaming**. In: EUSIPCO – European Signal Processing Conference 20, 2012, Bucharest. *Proceedings...*

LEDERER, S. et al. **Libdash**: An Open Source Software Library for the MPEG-DASH Standard. In: IEEE INTERNATIONAL CONFERENCE ON MULTIMEDIA AND EXPO, San José, 2013. *Proceedings...*

LEVKOV, M. **Video encoding and transcoding recommendations for HTTP Dynamic Streaming on the Flash Platform**: preliminary recommendations for video on demand. Adobe Systems (on-line), Oct. 19, 2010. Available in: <http://download.macromedia.com/flashmediaserver/http_encoding_recommendations.pdf>. Accessed on: Dec. 21, 2015.

LEVKOV, M. **Video encoding cookbook and profile guidelines for the Adobe Flash Plataform**. Adobe Systems (on-line), Nov. 30, 2011. Available in: <http://www.adobe.com/content/dam/Adobe/en/devnet/adobe-media-server/pdfs/video-encoding-cookbook.pdf>. Accessed on: Dec. 21, 2015.

LI, C.; BOVIK, A. C. **Content-partitioned structural similarity index for image quality assessment**. *Signal Processing:* Image Communication 25, Amsterdam, Elsevier, p. 517-526, 2010 (b).

LI, C.; BOVIK, A. C. **Content-weighted video quality assessment using a three-component image model**. *Journal of Electronic Imaging*, Bellingham, SPIE, v. 19, Jan.-Mar. 2010 (a).

LIN, W.; KUO, C. J. **Perceptual visual quality metrics: a survey**. *Journal of Visual Communication and Image Representation*, Amsterdam, Elsevier, 2011.

MARQUES, A.; BETTENCOURT, R.; FALCÃO, J.. **Internet Live Streaming**. Lisboa: Instituto Superior Técnico, May 2012.

MERRITT, L. et al. **X264**: a high performance.264/AVC encoder. Seattle: Washington University, 2006.

MERRITT, L.. **Notes on the implementation of trellis quantization in H.264**. Akuvian.org (on-line), Nov. 3, 2005. Available in: <http://akuvian.org/src/x264/trellis.txt>. Accessed on: June 1, 2014.

MEWIKI. **X264 settings**. MeWiki (on-line), Dec. 9, 2012. Available in: <http://en.wikibooks.org/wiki/MeGUI/x264_Settings>. Accessed on: June 1, 2013.

MONTEIRO, E. C. et al. **Perceptual video quality assessment in adaptive streaming encoding**. In: VCIP – IEEE International Conference on Visual Communications and Image Processing, Dec. 13-16, 2015, Singapore. *Proceedings...*

MOORTHY, A. K. et al. **Video Quality Assessment on Mobile Devices**: Subjective, Behavioral and Objective Studies. *IEEE Journal of Selected Topics in Signal Processing*, Piscataway, v. 6, n. 6, p. 652-671, Oct. 2012.

MOORTHY, A. K.; BOVIK, A. C. **Efficient Motion Weighted Spatio-Temporal Video SSIM Index**. Austin: Texas University, Nov. 2009 (a).

MOORTHY, A. K.; BOVIK, A. C. **Efficient video quality assessment along temporal trajectories**. *IEEE Transactions on Circuits and Systems for Video Technology*, Chicago, v. 20, n. 11, p. 1653-1658, 2010.

MOORTHY, A. K.; BOVIK, A. C. **Visual importance pooling for image quality assessment**. *IEEE Journal of Selected Topics in Signal Processing*, Piscataway , v. 3, n. 2, p. 193–201, Apr. 2009 (b).

MULTICOREWARE. **Command line options**. X265 documentation (on-line), 2015. Available in: <http://x265.readthedocs.org/en/stable/cli.html>. Accessed on: Dec. 21, 2015.

PARK, A.; WATSON, M. **HTML5 video at Netflix**. *The Netflix Tech Blog*. Netflix (on-line), Apr. 15, 2013. Available in: <http://techblog.netflix.com/2013/04/html5-video-at-netflix.html>. Accessed on: June 1, 2013.

PATTERSON, J. R. C. **Video encoding settings for H.264 excellence**. Lighterra (on-line), Apr. 2012. Available in: <http://www.lighterra.com/papers/videoencodingh264>. Accessed on: June 1, 2013.

PINSON, M. H.; WOLF, S. **A new standardized method for objectively measuring video quality.** *IEEE Transactions on Broadcasting*, Piscataway, v. 50, n. 3, p. 312-313, Sep. 2004.

PONOMARENKO, N. et al. **Image database TID2013:** Peculiarities, results and perspectives. *Signal Processing: Image Communication*, Elsevier, v. 30, 2015, p. 57-77.

PONOMARENKO, N. et al. **TID2008—**A database for evaluation of full-reference visual quality assessment metrics. *Advances of Modern Radio Electronics*, v. 10, n. 4, p. 30–45, 2009.

REHMAN, A.; ZENG, K.; WANG, Z. **Display device-adapted video quality-of-experience assessment**. In: IS&T-SPIE ELECTRONIC IMAGING, Human Vision and Electronic Imaging XX, Burlingame, Feb. 2015. *Proceedings…*

ROUSE, D.; HEMAMI, S. **Analyzing the role of visual structure in the recognition of natural image content with multi-scale SSIM**. In: SPIE HUMAN VISION AND ELECTRONIC IMAGING XIII, Feb. 2008 (a), San José. *Proceedings…*

ROUSE, D.; HEMAMI, S. **Understanding and simplifying the structural similarity metric**. In: ICIP – IEEE International Conference on Image Processing, Oct. 2008 (b), San Diego. *Proceedings…* p. 1188-1191.

SANDVINE. **Over 70% of North American traffic is now streaming video and audio**. Sandvine (on-line), Waterloo, Dec. 7, 2015. Available in: <http://www.sandvine.com/pr/2015/12/7/sandvine-over-70-of-north-american-traffic-is-now-streaming-video-and-audio.html>. Accessed on: Jan. 26, 2016.

SESHADRINATHAN, K. et al. **Study of subjective and objective quality assessment of video**. *IEEE Transactions on Image Processing*, Charlottesville, v. 19, n. 6, p. 1427–1441, 2010.

SESHADRINATHAN, K.; BOVIK, A. C. **Motion-based perceptual quality assessment of video**. In: IS&T/SPIE ELECTRONIC IMAGING, Feb. 2009, San José. *Proceedings…*

SHEIKH, H. R.; SABIR, M. F.; BOVIK, A. C. **A statistical evaluation of recent full reference image quality assessment algorithms**. *IEEE Transactions on Image Processing,* v. 15, n. 11, p. 3440-3451, Nov. 2006.

SHIKARI, D. (Jason Garret-Glaser) et al. **VC-1 over MPEG-4 AVC on Blu-Ray**?, message #3. Doom9's Forum (on-line), Dec. 1, 2008 (c). Available in: <http://forum.doom9.org/showthread.php?t=143196>. Accessed on: June 1, 2013.

SHIKARI, D. (Jason Garret-Glaser). **Encoding animation**. Diary Of An x264 Developer (on-line), Aug. 9, 2009 (c). Available in: <http://web.archive.org/web/20141124052059/http://x264dev.multimedia.cx/archives/102>. Accessed on: Dec. 21, 2015.

SHIKARI, D. (Jason Garret-Glaser). **Film grain optimization**. Diary Of An x264 Developer (on-line), May 3, 2009 (b). Available in: <http://web.archive.org/web/20150501153035/http://x264dev.multimedia.cx/archives/25>. Accessed on: Dec. 21, 2015.

SHIKARI, D. (Jason Garret-Glaser). **Flash, Google, VP8, and the future of internet video**. Diary Of An x264 Developer (on-line), Feb. 22, 2010 (a). Available in: <http://web.archive.org/web/20150323044656/http://x264dev.multimedia.cx/archives/292>. Accessed on: Dec. 21, 2015.

SHIKARI, D. (Jason Garret-Glaser). **How to cheat on video encoder comparisons**. Diary Of An x264 Developer (on-line), June 21, 2010 (b). Available in: <http://web.archive.org/web/20141103202912/http://x264dev.multimedia.cx/archives/472>. Accessed on: Dec. 21, 2015.

SHIKARI, D. (Jason Garret-Glaser). **Psy RDO**. Diary Of An x264 Developer (on-line), July 17, 2008 (a). Available in: <http://web.archive.org/web/20150419071059/http://x264dev.multimedia.cx/archives/37>. Accessed on: Dec. 21, 2015.

SHIKARI, D. (Jason Garret-Glaser). **Why so many H.264 encoders are bad**. Diary Of An x264 Developer (on-line), Oct. 4, 2009 (a). Available in: <http://web.archive.org/web/20150419053534/http://x264dev.multimedia.cx/archives/164>. Accessed on: Dec. 21, 2015.

SHIKARI, D. (Jason Garret-Glaser). **X264 adaptive quantization and you**. AnimeSuki Forums (on-line), Mar. 31, 2008 (b). Available in: <http://forums.animesuki.com/showthread.php?t=64485>. Accessed on: June 1, 2013.

SODAGAR, I. **MPEG-DASH**: The Standard for Multimedia Streaming Over Internet. In: IEEE MULTIMEDIA, v. 18, n. 4, Oct.-Dec. 2011, Trier, p. 62-67.

STATCOUNTER. **StatCounter global stats**: screen resolution. StatCounter (on-line), Oct. 2015. Available in: <http://gs.statcounter.com/#all-resolution-ww-monthly-201410-201510>. Accessed on: Dec. 21, 2015.

STOCKHAMMER, T.. **Dynamic Adaptive Streaming over HTTP**: Design Principles and Standards. In: W3C WEB AND TV WORKSHOP, 2, 2011, Berlin. *Proceedings…*

VATOLIN, D. et al. **MPEG-4 AVC/H.264 video codecs comparison**. Graphics & Media Lab Video Group (on-line), Moscow State University, May, 2012. Available in: <http://compression.ru/video/codec_comparison/h264_2012/mpeg4_avc_h264_video_codecs_comparison.pdf>. Accessed on: Dec. 21, 2015.

VIDEO QUALITY EXPERTS GROUP. **Final report from the Video Quality Experts Group on the validation of objective models of video quality assessment**, phase II. Boulder: Institute for Telecommunication Sciences, Aug. 25, 2003.

VIDEO QUALITY EXPERTS GROUP. **Report on the validation of video quality models for high definition video content**, version 2.0. Boulder: Institute for Telecommunication Sciences, June 30, 2010.

WAGGONER, B. **Compression for Great Video and Audio**: Master Tips and Common Sense. 2nd edition. [S.l]: Focal Press, 2010. 624 p.

WANG, S. et al. **Perceptual video coding based on SSIM-inspired divisive normalization**. *IEEE Transactions on Image Processing*, v. 22, n. 4, p. 1418–1429, 2013.

WANG, S. et al. **SSIM-motivated rate-distortion optimization for video coding**. *IEEE Transactions on Circuits and Systems for Video Technology*, Chicago, v. 22, n. 4, p. 516–529, 2012.

WANG, Z.; LI, Q. **Information content weighting for perceptual image quality assessment**. *IEEE Transactions on Image Processing*, Charlottesville, v. 20, n. 5, p. 1185–1198, 2011.

WANG, Z.; LI, Q. **Video quality assessment using a statistical model of human visual speed perception**. *Journal of the Optical Society of America A*, v. 24, p. B61–B69, 2007.

WANG, Z.; LU, L.; BOVIK, A. C. **Video quality assessment based on structural distortion measurement**. *Signal Processing: Image Communication*, v. 19, n. 2, p. 121–132, 2004.

WANG, Z.; SHANG, X. **Spatial pooling strategies for perceptual image quality assessment**. In: ICIP – IEEE International Conference on Image Processing, 2006, Atlanta. *Proceedings…*

WANG, Z.; SIMONCELLI, E. P. **Translation insensitive image similarity in complex wavelet domain**. ICASSP – IEEE International Conference on Acoustics, Speech and Signal Processing, v. 2, p. 573–576, Mar. 2005, Philadelphia. *Proceedings…*

WANG, Z. et al. **Image quality assessment**: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, v. 13, n. 4, Apr. 2004, p. 600-612.

WANG, Z. et al. **The SSIM index for image quality assessment**. University of Waterloo (on-line), 2011. Available in: <http://ece.uwaterloo.ca/~z70wang/research/ssim/index.html>. Accessed on: Dec. 21, 2015.

WANG, Z.; BOVIK, A. C. **A universal image quality index.** *IEEE Signal Processing Letters*, v. 9, p. 81-84, Mar. 2002.

WANG, Z.; BOVIK, A. C. **Mean squared error**: love it or leave it? *IEEE Signal Processing Magazine*. Jan. 2009.

WANG, Z.; SIMONCELLI, E. P.; BOVIK, A. C. **Multi-scale structural similarity for image quality assessment**. In: IEEE ASILOMAR CONFERENCE ON SIGNALS, SYSTEMS AND COMPUTERS, Nov. 2003, Pacific Grove. *Proceedings...*

XUE, W. et al. **Gradient Magnitude Similarity Deviation**: A Highly Efficient Perceptual Image Quality Index. *IEEE Transactions on Image Processing*, Charlottesville, v. 23, p. 684-695, Feb. 2014.

ZAMBELLI, Alex. **IIS Smooth Streaming technical overview**. [S.l.]: Microsoft, 2009.

ZHANG, L. et al. **FSIM: A Feature Similarity Index for Image Quality Assessment**. *IEEE Transactions on Image Processing*, v. 20, p. 2378-2386, Aug. 2011.

APPENDIX A: FURTHER READING

This work involved extensive research in practical encoding for Web streaming, producing dozens of interesting references, not all of which were necessary for citation in the main text. They are presented here as recommendations for the interested reader, organized in topics: Web video statistics, video coding in general, encoding for HTML5, H.264 encoding, MPEG-DASH, and image and video quality assessment.

A.1. Web video statistics

AKAMAI TECHNOLOGIES. **Akamai releases second quarter 2015 'State of the Internet' report**. Akamai Technologies (on-line), Sep. 23, 2015. Available in: <https://www.akamai.com/us/en/about/news/press/2015-press/akamai-releases-second-quarter-2015-state-of-the-internet-report.jsp>. Accessed on: Dec. 21, 2015.

LI, Mingzhe et al. **Characteristics of streaming media stored on the Web**. *Computer Science Technical Report Series*. Worcester: Worcester Polytechnic Institute, May, 2003.

TAGIAROLI, Guilherme. **Média de velocidade do 3G no Brasil fica abaixo do que operadoras prometem, diz pesquisa**. São Paulo: UOL Notícias (on-line), Feb. 6, 2013. Available in: <http://tecnologia.uol.com.br/noticias/redacao/2013/02/06/media-de-velocidade-do-3g-no-brasil-fica-abaixo-do-que-operadoras-prometem-diz-pesquisa.htm>. Accessed on: June 1, 2013.

TELECO. **INTERNET no Brasil** – Estatísticas. Teleco (on-line), Sep. 16, 2015. Available in: <http://www.teleco.com.br/internet.asp>. Accessed on: Dec. 21, 2015.

A.2. Video coding in general

ADHIKARI, Vijay Kumar et al. **Unreeling Netflix**: understanding and improving multi-CDN movie delivery. In: IEEE INFOCOM, Orlando, Mar. 2012. *Proceedings...*

APPLE. **Encoding video materials for DVD**. DVD Studio Pro 4 User Manual. Apple (on-line), nov. 2011. Available in: <http://documentation.apple.com/en/dvdstudiopro/usermanual/index.html#chapter=4%26section=6>. Accessed on: June 1, 2013.

BENES. **Blu-ray movie bitrates here**. Blu-ray Forum (on-line), Nov. 8, 2006. Available in: <http://forum.blu-ray.com/showthread.php?t=3338>. Accessed on: June 1, 2013.

BODE, Karl. **Netflix quietly helps capped U.S. broadband users with new video quality settings that first appeared in Canada**. DSLReports.com (on-line), June 22, 2011. Available in: <http://www.dslreports.com/shownews/Netflix-Quietly-Helps-Capped-US-Broadband-Users-114834>. Accessed on: Dec. 21, 2015.

DAILYMOTION. **Upload guidelines**. Dailymotion (on-line). Available in: <http://www.dailymotion.com/upload/faq>. Accessed on: June 1, 2013.

DOOM9. **Aspect ratios**. Doom9.org (on-line), July 30, 2004. Available in: <http://www.doom9.org/aspectratios.htm>. Accessed on: June 1, 2013.

ENCODING.com. **What is Vid.ly Lite and how do I use it**? Vid.ly (on-line). Available in: <http://www.encoding.com/what_is_vid.ly_lite_and_how_do_i_use_it>. Accessed on: Feb. 1, 2014.

HASSLER, Bjoern. **Youtube bitrates**. B's Blog (on-line), June 24, 2013. Available in: <http://www.sciencemedianetwork.org/Blog/20130624_YouTube_bitrates>. Accessed on: Feb. 1, 2014.

HOLLAND, David. **Netflix and Youtube dominate downstream bandwidth, fixed and mobile**. ReelSEO.com (on-line), Feb. 25, 2014. Available in: <http://www.reelseo.com/netflix-youtube-bandwidth>. Accessed on: June 1, 2013.

KALTURA, Inc. **Best practices for multi-device transcoding**. Kaltura (on-line). Available in: <http://knowledge.kaltura.com/node/217>. Accessed on: Dec. 21, 2015.

KALTURA. **Recommended video source formats and specifications**. Kaltura (on-line). Available in: <http://knowledge.kaltura.com/node/837>. Accessed on: June 1, 2013.

LAUFER, Matt. **Best practices for high quality transcoding**. Encoding.com (on-line), Oct. 28, 2013. Available in: <http://features.encoding.com/blog/2013/10/28/best-practices-ensuring-high-quality-transcoding/>. Accessed on: Dec. 1, 2013.

LEVKOV, Maxim; NGUYEN, Tom. **Simple mobile video encoding recommendations for Flash Player and AIR**. *Adobe Developer Connection*. Adobe (on-line), Aug. 22, 2011. Available in: <http://www.adobe.com/devnet/devices/articles/mobile_video_encoding.html>. Accessed on: Dec. 21, 2015.

MCFARLAND, Patrick. **Approximate Youtube bitrates**. Ad Terras per Aspera (on-line), May 24, 2010. Available in: <http://adterrasperaspera.com/blog/2010/05/24/approximate-youtube-bitrates>. Accessed on: June 1, 2013.

MIKEYTS. **Netflix upgrades 1080p encoding for more detail at lower bitrates**, message nº 8. High-Def Digest Forums (on-line), Dec. 11, 2012. Available in: <http://forums.highdefdigest.com/hd-digital-downloads-new/128658-netflix-upgrades-1080p-encoding-more-detail-lower-bitrates.html#post2360960>. Accessed on: June 1, 2013.

HYRAL (Estêvão C. Monteiro); WAGGONER, Ben. **Resolutions for adaptive web content**. Doom9's Forum (on-line). Doom9.org, Oct. 15, 2013. Available in: <http://forum.doom9.org/showthread.php?p=1648048>. Accessed on: Oct. 15, 2013.

AARON, Anne; RONCA, David. **High quality video encoding at scale**. The Netflix Tech Blog (on-line), Dec. 9, 2015. Available in: <http://techblog.netflix.com/2015/12/high-quality-video-encoding-at-scale.html>. Accessed on: Dec. 21, 2015.

GOVIND, Nirmal; BALACHANDRAN, Athula.. **Optimizing content quality control at Netflix with predictive modeling**. The Netflix Tech Blog (on-line), Dec. 10, 2015. Available in: <http://techblog.netflix.com/2015/12/optimizing-content-quality-control-at-netflix-predictive-modeling.html>. Accessed on: Dec. 21, 2015.

OZER, Jan. **Video compression for Flash, Apple devices and HTML5**. USA: Doceo Publishing, May 2, 2011. 272 p.

RICK, Christophor. **Vid.ly**: Upload Video, Grab URL, Play on Almost Every Connected Device. ReelSEO.com (on-line), Jan. 24, 2011. Available in: <http://www.reelseo.com/vidly>. Accessed on: June 1, 2013.

SAM. **Unbox video quality**. *Amazon Instant Video Blog*. Typepad (on-line), Apr. 25, 2007. Available in: <http://unbox.typepad.com/amazon_unbox/2007/04/unbox_video_qua.html>. Accessed on: June 1, 2013.

SCOTT, Michael. **Netflix streaming quality**, message nº 3102. AVS Forum (on-line), Oct. 2013. Available in: <http://www.avsforum.com/t/1089285/netflix-streaming-quality/3090#post_23803189>. Accessed on: Dec. 1, 2013.

SCOTT, Michael. **Odd Netflix issue – X-High/HD no longer available**, message nº 154. AVS Forum (on-line), Dec. 13, 2012. Available in: <http://www.avsforum.com/forum/184-video-download-services-hardware/1440503-odd-netflix-issue-x-high-hd-no-longer-available-6.html#post22699826>. Accessed on: june 1, 2013.

SOUZA, D. **Odd Netflix issue – X-High/HD no longer available**, message nº 131. AVS Forum (on-line), Dec. 8, 2012. Available in: <http://www.avsforum.com/forum/184-video-download-services-hardware/1440503-odd-netflix-issue-x-high-hd-no-longer-available-5.html>. Accessed on: June 1, 2013.

TAYLOR, Jim. **DVD frequently asked questions** (and answers). DVD Demystified (on-line), June 27, 2013. Available in: <http://www.dvddemystified.com/dvdfaq.html#3.4>. Accessed on: June 1, 2013.

VIDEOHELP.com. **What is Blu-ray Disc, AVCHD and HD DVD**? VideoHelp.com (on-line), 2006. Available in: <http://www.videohelp.com/hd>. Accessed on: June 1, 2013.

VIDEOHELP.com. **What is DVD?** VideoHelp.com (on-line), 2004. Available in: <http://www.videohelp.com/dvd>. Accessed on: June 1, 2013.

VIMEO. **Compression guidelines on Vimeo**. Vimeo (on-line). Available in: <http://vimeo.com/help/compression>. Accessed on: June 1, 2013.

WIKIPEDIA.org. **List of displays by pixel density**. Wikipedia (on-line), 2013. Available in: <http://en.wikipedia.org/wiki/List_of_displays_by_pixel_density>. Accessed on: June 1, 2013.

ZAMBELLI, Alex. **An inside look at NBC Olympics video player**. Alex Zambelli's Streaming Media Blog, Aug. 21, 2008. Available in: <http://alexzambelli.com/blog/2008/08/21/an-inside-look-at-nbc-olympics-video-player>. Accessed on: June 1, 2013.

ZAMBELLI, Alex. **Baptism of fire in the olympic cauldron**. Alex Zambelli's Streaming Media Blog (on-line), Feb. 19, 2014. Available in: <http://alexzambelli.com/blog/2014/02/19/baptism-of-fire-in-the-olympic-cauldron>. Accessed on: Mar. 1, 2013.

ZAMBELLI, Alex. **H.265/HEVC ratification and 4K video streaming**. Alex Zambelli's Streaming Media Blog (on-line), Jan. 28, 2013. Available in: <http://alexzambelli.com/blog/2013/01/28/h-265hevc-ratification-and-4k-video-streaming>. Accessed on: June 1, 2013.

ZAMBELLI, Alex. **Smooth Streaming multi-bitrate calculator**. Windows Media Video Tools (on-line), Jan. 24, 2013. Available in: <http://alexzambelli.com/WMV/MBRCalc.html>. Accessed on: June 1, 2013.

ZENCODER. **IOS/mobile encoding**. Zencoder (on-line), 2012. Available in: <http://ap.zencoder.com/docs/guides/encoding-settings/ios-and-mobile>. Accessed on: June 1, 2013.

A.3. MPEG Dynamic Adaptive Streaming over HTTP

AKHSHABI, Saamer; BEGEN, Ali; DOVROLIS, Constantine. **An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP**. In: MMSYS – ACM Multimedia Systems Conference, 2, 2011, New York. *Proceedings...*

DASH INDUSTRY FORUM. **For promotion of MPEG-DASH**. DASH Industry Forum (on-line), Jun. 2013. Available in: <http://dashif.org>. Accessed on: Sep. 1, 2013.

DE CICCO, Luca; MASCOLO, Saverio. **An experimental investigation of the Akamai adaptive video streaming**. In: USAB – Usability Symposium, 2010, Klagenfurt. *Proceedings...*

GPAC MULTIMEDIA OPEN SOURCE PROJECT. **MP4Box**. Telecom ParisTech (on-line). Available in: <http://gpac.wp.mines-telecom.fr/mp4box>. Accessed on: June 1, 2013.

ISO; IEC. **ISO/IEC 23009-1**:2012: Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats. *ISO Standards Catalogue*. [S.l]: Apr. 3, 2012.

LEDERER, Stefan et al. **An evaluation of Dynamic Adaptive Streaming over HTTP in vehicular environments**. In: MMSYS – ACM Multimedia Systems Conference, 4, 2012, Chapel hill. *Proceedings...*

MPEG-DASH INDUSTRY FORUM. **Overview of MPEG-DASH standard**. MPEG-DASH Industry Forum (on-line). Available in: <http://web.archive.org/web/20121224080930/http://dashpg.com/mpeg-dash>. Accessed on: Dec. 21, 2015.

MÜLLER, Christopher; TIMMERER, Christian. **A VLC media player plugin enabling Dynamic Adaptive Streaming over HTTP**. In: MMSYS – ACM Multimedia Systems Conference, Nov. 28, 2011, Scottsdale. *Proceedings...*

TIMMERER, Christian. **HTTP streaming of MPEG media**. Multimedia Communication (on-line), Apr. 26, 2012. Available in: <http://multimedia communication.blogspot.com/2010/05/http-streaming-of-mpeg-media.html>. Accessed on: Oct. 19, 2012.

TIWARI, Rajeev. **MPEG-DASH support in Youtube**. Streaming Media and RTOS (on-line), Jan. 3, 2013. Available in: <http://streamingcodecs.blogspot.hu/2013/01/mpeg-dash-support-in-youtube.html>. Accessed on: June 1, 2013.

A.4. Encoding for HTML5

AKUPENGUIN (Loren Merritt). **Any benefit in mod8 over mod4**? Doom9's Forums (on-line), Apr. 5, 2009. Available in: <http://forum.doom9.org/showthread.php?t=146148&highlight=mod16>. Accessed on: June 1, 2013.

ARTHUR, Charles. **Google's WebM v H.264**: who wins and loses in the video codec wars? *Technology Blog*. The Guardian (on-line), Jan. 17, 2011. Available in: <http://www.theguardian.com/technology/blog/2011/jan/17/google-webm-vp8-video-html5-h264-winners-losers>. Accessed on: Dec. 21, 2015.

GROIS, Dan et al. **Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders**. In: 30th PICTURE CODING SYMPOSIUM 2013, Dec. 8-11, 2013, San José. *Proceedings...*

NETFLIX. **NfWebCrypto**. GitHub (on-line). Available in: <http://github.com/Netflix/ nfwebcrypto>. Accessed on: Dec. 1, 2013.

OZER, Jan. **The secret to encoding high quality web video**: Tutorial. ReelSEO.com (on-line), June 7, 2011. Available in: <http://www.reelseo.com/secret-encoding-web-video>. Accessed on: June 1, 2013.

SHIKARI, Dark (Jason Garret-Glaser); AKUPENGUIN (Loren Merritt). **Mod 16 x264**. Doom9's Forums (on-line), Mar. 5, 2008. Available in: <http://forum.doom9.org/showthread.php?p=1108947>. Accessed on: June 1, 2013.

SHIKARI, Dark (Jason Garret-Glaser). **H.264 and VP8 for still image coding**: WebP? Diary Of An x264 Developer (on-line), Sep. 30, 2010. Available in: <http://web.archive.org/web/20150419071902/http://x264dev.multimedia.cx/archives/541>. Accessed on: Dec. 21, 2015.

SHIKARI, Dark (Jason Garret-Glaser). **Stop doing this in your encoder comparisons**. Diary Of An x264 Developer (on-line), June 14, 2010. Available in: <http://web.archive.org/web/20150223215544/http://x264dev.multimedia.cx/archives/458>. Accessed on: Dec. 21, 2015.

SHIKARI, Dark (Jason Garret-Glaser). **The first in-depth technical analysis of VP8**. Diary Of An x264 Developer (on-line), May 19, 2010. Available in: <http://web.archive.org/web/20150411070012/http://x264dev.multimedia.cx/archives/377>. Accessed on: Dec. 21, 2015.

SHIKARI, Dark (Jason Garret-Glaser). **The problems with wavelets**. Diary Of An x264 Developer (on-line), Feb. 26, 2010. Available in: <http://web.archive.org/web/20150416142524/http://x264dev.multimedia.cx/archives/317>. Accessed on: Dec. 21, 2015.

SHIKARI, Dark (Jason Garret-Glaser). **VP8: a retrospective**. Diary Of An x264 Developer (on-line), July 13, 2010. Available in: <http://web.archive.org/web/20150301015756/http://x264dev.multimedia.cx/archives/486>. Accessed on: Dec. 21, 2015.

VATOLIN, Dmitriy *et al*. **HEVC/H.265 video codecs comparison**. Moscow: Graphics & Media Lab Video Group, Moscow State University, Oct. 15, 2015.

WAGGONER, Ben. **Updated Rule of ^3/4 for H.264 high profile?** Doom9's Forums (on-line), May. 8, 2013. Available in: <http://forum.doom9.org/showthread.php?t=167816>. Accessed on: June 1, 2013.

A.5. H.264 encoding

MEWIKI. **X264 encoding suggestions**. MeWiki (on-line), Dec. 13, 2012. Available in: <http://web.archive.org/web/20150208111101/http://mewiki.project357.com/wiki/X264_Encoding_Suggestions>. Accessed on: June 1, 2013.

MULDER. **Video quality metric**, message n. 4. Doom9's Forums (on-line), Apr. 2009. Available in: <http://forum.doom9.org/showthread.php?p=1270886>. Accessed on: June 1, 2013.

RICHARDSON, Iain. **H.264 and MPEG-4 video compression**: video coding for next-generation multimedia. Aberdeen: Robert Gordon University, 2003.

SHIKARI, Dark (Jason Garret-Glaser). **Psy RDO**: Official testing thread. Doom9 Forum (on-line), May 31, 2008. Available in: <http://forum.doom9.org/showthread.php?t=138293>. Accessed on: June 1, 2013.

SHIKARI, Dark (Jason Garret-Glaser). **The spec-violation hall of shame**. Diary Of An x264 Developer (on-line), Nov. 15, 2009. Available in: <http://web.archive.org/web/20150426024711/http://x264dev.multimedia.cx/archives/212>. Accessed on: Dec. 21, 2015.

SHIKARI, Dark (Jason Garret-Glaser). **X264: the best low-latency video streaming platform in the world**. Diary Of An x264 Developer (on-line), Jan. 13, 2010. Available in: <http://web.archive.org/web/20150507012544/http://x264dev.multimedia.cx/archives/249>. Accessed on: Dec. 21, 2015.

VATOLIN, Dmitriy et al. **MSU subjective comparison of modern video codecs**. Graphics & Media Lab Video Group (on-line), Moscow State University, Jan. 2006. Available in: <http://www.compression.ru/video/codec_comparison/subjective_codecs_comparison_en.html>. Accessed on: Dec. 21, 2015.

A.6. Image and video quality assessment

BOVIK, Alan C.; WANG, Zhou. **Modern image quality assessment**. [S.l.]: Morgan & Claypool Publishers, 2006. 146 p.

REHMAN, A.; ZENG, K.; WANG, Z. **Display device-adapted video quality-of-experience assessment**. In: IS&T-SPIE ELECTRONIC IMAGING, Human Vision and Electronic Imaging XX, Feb. 2015, San Francisco. *Proceedings…*

SESHADRINATHAN, K. et al. **A subjective study to evaluate video quality assessment algorithms**. *IS&T/SPIE Electronic Imaging*, v. 7527, 2010.
SOUNDARARAJAN, R.; BOVIK, A. C. **Video quality assessment by reduced reference spatio-temporal entropic differencing**. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 23, n. 4, p. 684–694, 2013.

ZHANG, L. et al. **A comprehensive evaluation of full reference image quality assessment algorithms**. ICIP – International Conference on Image Processing, p. 1477–1480, 2012. *Proceedings…*

APPENDIX B: VISUAL QUALITY INDEXES FOR THE LIVE MOBILE VIDEO QUALITY DATABASE

Table B.1 — Visual quality indexes for SSIM-based metrics.

Distorted video version	DMOS	SSIM	MS-SSIM	3-SSIM	GMSD	Fast SSIM
bf_r1	3.2438	0.9594	0.9717	0.9580	0.8956	0.9839
bf_r2	2.0938	0.9753	0.9865	0.9754	0.9260	0.9949
bf_r3	1.0438	0.9853	0.9935	0.9858	0.9516	0.9987
bf_r4	0.3563	0.9915	0.9969	0.9919	0.9705	0.9996
dv_r1	3.1688	0.9404	0.9625	0.9369	0.8675	0.9597
dv_r2	1.8438	0.9610	0.9793	0.9612	0.9028	0.9834
dv_r3	0.7875	0.9760	0.9895	0.9771	0.9356	0.9949
dv_r4	0.3625	0.9860	0.9949	0.9870	0.9613	0.9984
fc_r1	2.7750	0.9942	0.9917	0.9925	0.9661	0.9980
fc_r2	1.9813	0.9954	0.9932	0.9943	0.9725	0.9988
fc_r3	0.9688	0.9964	0.9944	0.9956	0.9784	0.9994
fc_r4	0.0500	0.9981	0.9981	0.9978	0.9895	0.9999
hc_r1	2.8563	0.9307	0.9625	0.9295	0.8571	0.9481
hc_r2	2.0313	0.9584	0.9824	0.9595	0.8992	0.9846
hc_r3	0.6875	0.9767	0.9918	0.9779	0.9368	0.9976
hc_r4	0.2750	0.9875	0.9962	0.9883	0.9640	0.9997
la_r1	3.2375	0.9847	0.9840	0.9837	0.9161	0.9907
la_r2	2.3250	0.9932	0.9944	0.9931	0.9534	0.9981
la_r3	0.7250	0.9956	0.9970	0.9956	0.9682	0.9993
la_r4	0.2438	0.9973	0.9983	0.9972	0.9803	0.9997
po_r1	3.7938	0.8578	0.9201	0.8408	0.8016	0.9112
po_r2	2.9125	0.9153	0.9614	0.9134	0.8562	0.9694
po_r3	1.6750	0.9514	0.9822	0.9536	0.9020	0.9922
po_r4	0.7563	0.9726	0.9916	0.9750	0.9379	0.9985
rb_r1	3.4750	0.9397	0.9606	0.9350	0.8686	0.9707
rb_r2	2.4563	0.9613	0.9798	0.9605	0.8992	0.9838
rb_r3	0.8688	0.9755	0.9896	0.9760	0.9248	0.9904
rb_r4	0.5250	0.9848	0.9946	0.9855	0.9470	0.9949
sd_r1	3.2250	0.9202	0.9215	0.9186	0.8415	0.9148
sd_r2	2.5500	0.9399	0.9518	0.9404	0.8681	0.9566
sd_r3	1.3938	0.9602	0.9762	0.9612	0.9004	0.9865
sd_r4	0.3063	0.9763	0.9892	0.9772	0.9326	0.9975
ss_r1	3.3000	0.9361	0.9545	0.9338	0.8566	0.9599
ss_r2	2.1438	0.9580	0.9757	0.9585	0.8895	0.9808
ss_r3	1.4750	0.9739	0.9879	0.9749	0.9208	0.9935
ss_r4	0.6313	0.9844	0.9940	0.9853	0.9482	0.9984
tk_r1	3.6563	0.8999	0.9346	0.8885	0.8214	0.9093
tk_r2	2.5500	0.9325	0.9636	0.9295	0.8611	0.9562
tk_r3	1.0938	0.9576	0.9816	0.9581	0.8997	0.9846
tk_r4	0.6250	0.9750	0.9910	0.9762	0.9342	0.9964
Mean	**1.7617**	**0.9639**	**0.9790**	**0.9630**	**0.9151**	**0.9818**

Table B.2— Visual quality indexes for SG-Sim-based metrics.

Distorted video version	DMOS	SG-Sim	Fast SG-Sim	SG-Sim (logical)	SG-Sim (Roberts, logical)	5S-SG-Sim	4S-SG-Sim	Fast MS-SG-Sim
bf_r1	3.2438	0.9826	0.9728	0.9609	0.9719	0.9957	0.9965	0.9934
bf_r2	2.0938	0.9950	0.9899	0.9859	0.9909	0.9992	0.9994	0.9985
bf_r3	1.0438	0.9990	0.9972	0.9958	0.9976	0.9999	0.9999	0.9997
bf_r4	0.3563	0.9999	0.9994	0.9989	0.9993	1.0000	1.0000	0.9999
dv_r1	3.1688	0.9609	0.9467	0.9050	0.9139	0.9882	0.9900	0.9841
dv_r2	1.8438	0.9848	0.9754	0.9549	0.9610	0.9968	0.9975	0.9950
dv_r3	0.7875	0.9961	0.9917	0.9839	0.9864	0.9994	0.9996	0.9989
dv_r4	0.3625	0.9993	0.9978	0.9950	0.9952	0.9999	0.9999	0.9998
fc_r1	2.7750	0.9980	0.9970	0.9937	0.9956	0.9980	0.9981	0.9973
fc_r2	1.9813	0.9988	0.9981	0.9956	0.9972	0.9984	0.9984	0.9978
fc_r3	0.9688	0.9994	0.9989	0.9973	0.9985	0.9989	0.9989	0.9985
fc_r4	0.0500	1.0000	0.9999	0.9998	0.9999	1.0000	1.0000	1.0000
hc_r1	2.8563	0.9529	0.9354	0.9187	0.9273	0.9919	0.9941	0.9901
hc_r2	2.0313	0.9874	0.9781	0.9744	0.9780	0.9988	0.9993	0.9984
hc_r3	0.6875	0.9985	0.9958	0.9955	0.9964	0.9999	0.9999	0.9998
hc_r4	0.2750	0.9999	0.9995	0.9994	0.9995	1.0000	1.0000	1.0000
la_r1	3.2375	0.9871	0.9816	0.9612	0.9763	0.9922	0.9928	0.9889
la_r2	2.3250	0.9975	0.9954	0.9878	0.9939	0.9984	0.9985	0.9973
la_r3	0.7250	0.9992	0.9981	0.9949	0.9977	0.9997	0.9997	0.9992
la_r4	0.2438	0.9998	0.9994	0.9985	0.9992	1.0000	1.0000	0.9999
po_r1	3.7938	0.9187	0.9003	0.8739	0.8807	0.9817	0.9854	0.9789
po_r2	2.9125	0.9718	0.9602	0.9464	0.9539	0.9945	0.9958	0.9929
po_r3	1.6750	0.9930	0.9873	0.9833	0.9876	0.9989	0.9992	0.9983
po_r4	0.7563	0.9989	0.9970	0.9963	0.9977	0.9998	0.9998	0.9996
rb_r1	3.4750	0.9714	0.9586	0.9384	0.9453	0.9936	0.9949	0.9912
rb_r2	2.4563	0.9862	0.9775	0.9609	0.9638	0.9974	0.9980	0.9959
rb_r3	0.8688	0.9933	0.9877	0.9761	0.9769	0.9990	0.9993	0.9983
rb_r4	0.5250	0.9972	0.9938	0.9875	0.9877	0.9997	0.9998	0.9994
sd_r1	3.2250	0.9069	0.8821	0.8020	0.8321	0.9513	0.9555	0.9379
sd_r2	2.5500	0.9503	0.9298	0.8942	0.9219	0.9831	0.9853	0.9762
sd_r3	1.3938	0.9842	0.9715	0.9664	0.9799	0.9975	0.9983	0.9959
sd_r4	0.3063	0.9974	0.9928	0.9936	0.9968	0.9998	0.9999	0.9997
ss_r1	3.3000	0.9593	0.9449	0.9244	0.9375	0.9891	0.9910	0.9855
ss_r2	2.1438	0.9813	0.9713	0.9595	0.9682	0.9961	0.9969	0.9942
ss_r3	1.4750	0.9943	0.9889	0.9845	0.9886	0.9991	0.9993	0.9984
ss_r4	0.6313	0.9989	0.9970	0.9954	0.9970	0.9998	0.9999	0.9997
tk_r1	3.6563	0.9164	0.8974	0.8555	0.8630	0.9760	0.9798	0.9713
tk_r2	2.5500	0.9608	0.9463	0.9246	0.9313	0.9923	0.9941	0.9900
tk_r3	1.0938	0.9874	0.9789	0.9720	0.9755	0.9985	0.9991	0.9978
tk_r4	0.6250	0.9977	0.9944	0.9931	0.9941	0.9998	0.9999	0.9997
Mean	**1.7617**	**0.9825**	**0.9752**	**0.9631**	**0.9689**	**0.9951**	**0.9958**	**0.9934**