University of Brasília

Institute of Exact Sciences
Department of Computer Science

Using Texture Measures for Visual Quality Assessment

Pedro Garcia Freitas

Thesis submitted in partial satisfaction of the requirements for the degree of
Doctor of Philosophy in Informatics

Doctoral Advisor
Prof. Dr. Mylene Christine Queiroz de Farias

Brasília
2017

Ficha Catalográfica de Teses e Dissertações

Está página existe apenas para indicar onde a ficha catalográfica gerada para dissertações de mestrado e teses de doutorado defendidas na UnB. A Biblioteca Central é responsável pela ficha, mais informações nos sítios:

http://www.bce.unb.br
http://www.bce.unb.br/elaboracao-de-fichas-catalograficas-de-teses-e-dissertacoes

Esta página não deve ser inclusa na versão final do texto.

University of Brasília

Institute of Exact Sciences
Department of Computer Science

Using Texture Measures for Visual Quality Assessment

Pedro Garcia Freitas

Thesis submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Informatics

Prof. Dr. Mylene Christine Queiroz de Farias (Doctoral Advisor)
CIC/UnB

Prof. Dr. Bruno L. M. Espinoza
CIC/UnB

Prof. Dr. Alexandre Ricardo Soares Romariz
ENE/UnB

Prof. Dr. Alexandre de Almeida Prado Pohl
UTFPR

Prof. Dr. Bruno Luiggi Macchiavello
Coordinator of Postgraduate Programs in Informatics

Brasília, 27 of September of 2017

To all Computer Science researchers, especially the great ones.

Acknowledgements

First and foremost, I would like to thank my family for giving me the support and encouragement to undertake this research. Without its blessings, this achievement would not have been possible.

I am very grateful to my advisor, Prof. Mylene C.Q. Farias. In my journey towards this work, she was a friend, an inspiration and a role model. Without her heartfelt support and guidance, this thesis would not have been possible. I shall enormously be grateful for her assistance.

I would like to thank my colleagues of the Digital Signal Processing Group (GPDS): Prof. Marcelo M. de Carvalho, Alessandro R. Silva, Helard B. Martinez, Max E. V. Melgar, Dário D. R. Morais, Gustavo L. Sandri, and Alexandre F. Silva. Their support, encouragement and ideas have been great contribution in the completion of this thesis. I am specially grateful to Welington Y. L. Akamine, who has helped me in several points of my research.

I thank my previous colleagues of Delft University of Technology (TU Delft): Raynor Vliegendhart, Alessio Bazzica, Ernestasia Siahaan, Yi Zhu, Ruud de Jong, Junchao Xu and Reyhan Aydogan. I am specially indebted with Dra. Judith A. Redi, my supervisor during my time in Netherlands. They have all made my life and work abroad a pleasing experience.

I would be glad to express gratitude to my friends Bernardo M. David, Felipe G. Lacerda, Érico M. P. Fonseca, Wagner M. Silva, and Hugo T. M. Kussaba. Their encouragement was fundamental to the successful completion of this work.

I would like to acknowledge the financial support for this work provided by the fellowship from the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Cientfico e Tecnológico (CNPq), and University of Brasília (UnB).

Resumo

Na última década, diversas aplicações multimídia tem gerado e distribuído conteúdos de imagens e vídeos digitais. Serviços de multimídia que tem ganhado um vasto interesse incluem televisão digital, jogos de vídeo e aplicações em tempo real operando sobre a Internet. De acordo com predições da Cisco^{TM}, a percentagem do tráfego de dados de vídeo sobre a Internet era de 53% em 2014 e superará os 67% em 2018. Devido à esse aumento na demanda de conteúdo de dados visuais, a necessidade de métodos e ferramentas que estimem a qualidade da experiência (QoE) do consumidor é enorme. Entre os aspectos que contribuem para a QoE, a qualidade dos estímulos visuais é uma das maiores propriedades, pois pode ser alterada em diversos estágios da cadeia de comunicação, tal como na captura, na transmissão, ou na reprodução do conteúdo.

Considerando que os avaliadores naturais da qualidade visual são seres humanos, a estratégia básica para medir a qualidade visual consiste na realização de experimentos subjetivos. Esses experimentos são geralmente realizados com participantes humanos em laboratórios preparados com um ambiente controlado. Esses participantes avaliam a qualidade de um dado estimulo visual (imagem ou vídeo) e atribuem a eles um valor numérico associado à qualidade. Para avaliar a qualidade, os participantes seguem um conjunto de passos experimentais. Geralmente, esses passos são padronizados para favorecer a reprodutibilidade experimental. Os padrões de experimentos incluem metodologias de avaliação, tais como condições de visualização, escala de avaliação, materiais, etc. Após um conjunto de participantes avaliarem individualmente a qualidade de um dado estímulo, a média dos valores é calculada para gerar o valor médio das opiniões subjetivas (MOS). O MOS é frequentemente utilizado para representar a qualidade geral de um dado estímulo visual.

Como a coleta dos MOS é realizada a partir de experimentos com seres humanos, esse processo é demorado, cansativo, caro, e laborioso. Devido ao custo dos experimentos subjetivos, um grande esforço tem sido dedicado ao desenvolvimento de técnicas objetivas para a avaliação de estímulos visuais. Essas técnicas objetivas consistem em predizer o MOS automaticamente por meio de algoritmos computacionais. Tal automação torna possível a implementação de procedimentos computacionais rápidos e baratos para monitorar

e controlar a qualidade de estímulos visuais.

As técnicas objetivas para a avaliação de estímulos visuais podem ser classificadas em três tipos, dependendo da quantidade de informação necessária pelo método. Se todo o estímulo de referência (original) é requerido para a estimação da qualidade do estímulo testado, então essa técnica é classificada como sendo de referência completa. Quando somente alguma informação parcial da referência é necessária, a técnica é classificada como sendo de referência reduzida. Por outro lado, quando nenhuma informação sobre o estímulo de referência é necessária, a técnica é dita como sendo sem referência. Uma vez que a exigência de uma referência completa ou parcial é um obstáculo no desenvolvimento de diversas aplicações multimídia, as técnicas de sem referência são as mais convenientes na maioria dos casos.

Diversas técnicas objetivas para avaliação de qualidade visual têm sido propostas, embora ainda existam algumas questões em aberto no seu desenvolvimento. No caso de técnicas de avaliação de imagens, diversas técnicas de referência completa têm sido produzidas com uma excelente performance. Por outro lado, técnicas que não utilizam referências ainda apresentam limitações quando múltiplas distorções estão presentes. Além disso, as técnicas sem referência para imagens mais eficientes ainda apresentam modelos computacionalmente custosos, o que limita a utilização desses métodos em várias aplicações multimídia.

No caso de vídeos, o atual estado da arte ainda possui performance na predição dos MOS pior do que os métodos de imagens. Quando consideramos a acurácia da predição, os métodos objetivos para vídeos possuem uma correlação entre valores preditos e MOS ainda pequena se comparada com a correlação observada em métodos para imagens. Além disso, a complexidade computacional é ainda mais crítica no caso de vídeos, uma vez que a quantidade de informação processada é muito maior do que aquela presente em imagens.

O desenvolvimento de uma técnica objetiva de avaliação de qualidade visual requer resolver três grandes problemas. O primeiro problema é determinar um conjunto de características que sejam relevantes na descrição da qualidade visual. Essas características, geralmente, referem-se às medidas de estímulos físicos, tais como quantificação da nitidez de borda, estatísticas de cenas naturais, estatísticas no domínio de curvlets, filtros de Prewitt, etc. Além disso, múltiplos tipos de características podem ser combinados para gerar um vetor de características que descrevem melhor a qualidade de um dado estímulo. O segundo problema é estabelecer uma estratégia de agrupamento das características de forma que os valores numéricos sejam descritivos dentro de um modelo. Esse agrupamento se refere a uma combinação de medidas através de um subespaço de medidas para representar o estímulo analisado. Finalmente, o terceiro problema é a criação de um modelo que mapeie as características agrupadas de forma que se correlacione com os dados

preditos com os subjetivos.

Neste trabalho, nós apresentamos uma investigação de métodos de avaliação de qualidade visual baseada na medição de texturas. A pressuposição é que degradações visuais alteram as texturas e as estatísticas dessas texturas em imagens e vídeos. Essas medidas são executadas em termos das estatísticas extraídas do operador de padrões binários locais (LBP) e suas extensões. Este operador foi escolhido porque ele unifica outros modelos de análise de texturas mais tradicionais, tais como o espectro de textura, o nível de cinza de comprimento (GLRLM) e as matrizes de co-ocorrência de níveis de cinza (GLCM). O operador LBP, sendo um algoritmo simples e que favorece implementações rápidas, possui propriedades muito úteis para sistemas de processamento em tempo real de imagens e vídeos.

Devido às vantagens supracitadas, nós analisamos o operador LBP e algumas das suas extensões no estado da arte com o objetivo de investigar sua adequabilidade para o problema de avaliação de qualidade de imagens. Para isso, neste trabalho nós apresentamos uma extensa revisão do estado da arte dos operadores. Entre os operadores no estado da arte, podemos mencionar os padrões ternários locais (LTP), a quantização de fase local (LPQ), as estatísticas binarizadas de características de imagem (BSIF), os padrões locais binários rotacionados (RLBP), os padrões binários locais completos (CLBP), os padrões de configuração locais (LCP), entre outros. Ademais, nós também propomos novas extensões que melhoram a predição de qualidade. Entre as extensões propostas para a medida de características de qualidade, estão os padrões binários locais de múltipla escala (MLBP), os padrões ternários locais de múltipla escala (MLTP), os padrões de variância local (LVP), os padrões de planos ortogonais de cores (OCPP), os padrões binários locais salientes (SLBP) e os padrões binários locais salientes de múltipla escala (MSLBP).

Para testar a adequabilidade dos operadores de texturas supracitados, propomos um arcabouço para utilizar esses operadores na produção de novas métricas de qualidade de imagens. Dessa forma, muitas métricas sem referência podem ser geradas a partir da estratégia proposta. Utilizando as métricas geradas a partir do arcabouço proposto, uma extensa análise comparativa é apresentada neste trabalho. Essa análise foi feita com três das mais populares bases de dados de qualidade imagens disponíveis, sendo elas a LIVE, CSIQ e TID 2013. Os resultados gerados a partir dos testes nessas bases demonstram que os operadores no estado da arte mais adequados para mensurar a qualidade de imagens são o BSIF, o LPQ e o CLBP. Todavia, os resultados também indicaram que os operadores propostos atingiram resultados ainda mais promissores, com as abordagens baseadas em múltiplas escalas apresentando os melhores desempenhos entre todas variações testadas.

Inspirado nos resultados experimentais das métricas de imagens geradas, nós escolhemos um operador de textura conveniente para implementar uma métrica de avaliação de qualidade de vídeos. Além de incorporar informações de textura, nós também incorporamos informações de atividade espacial e informação temporal. Os resultados experimentais obtidos indicam que a métrica proposta tem uma performance consideravelmente superior quando testada em diversas bases de dados de vídeo de referência e supera os atuais modelos de qualidade vídeo.

Palavras-chave: Qualidade visual, métricas objetivas, avaliação da qualidade de imagens sem referência, avaliação da qualidade de vídeos

Abstract

In the last decade, many visual quality models have been proposed. However, there are some open questions involving the assessment of image and video quality. In the case of images, most of the proposed methods are very complex and require a reference content to estimate the quality, limiting their use in several multimedia application. For videos, the current state-of-the-art methods still perform worse than images in terms of prediction accuracy.

In this work, we present an investigation of visual quality assessment methods based on texture measurements. The premise is that visual impairments alter image and video textures and their statistics. These measurements are performed regarding the statistics of the local binary pattern (LBP) operator and its extensions. We chosen LBP because it unifies traditional texture analysis models. In addition, LBP is a simple but effective algorithm that performs only fundamental operations, which favors fast and simple implementations, which is very useful for real-time image and video processing systems.

Because of the abovementioned advantages, we analyzed the LBP operator and some of its state-of-the-art extensions addressing the problem of assessing image quality. Furthermore, we also propose new quality-aware LBP extensions to improve the prediction of quality. Then, we propose a framework for using these operators in order to produce new image quality metrics. Therefore, many no-reference image quality metrics can be generated from the proposed strategy.

Inspired by experimental results of generated no-reference image quality metrics, we chosen a convenient texture operator to implement a full-reference video quality metric. In addition to the texture information, we also incorporate features including spatial activity, and temporal information. Experimental results indicated that our metric presents a superior performance when tested on several benchmark video quality databases, outperforming current state-of-the-art full-reference video quality metrics.

Keywords: Visual quality, objective metrics, no-reference image quality assessment, video quality assessment

Contents

List of Figures

List of Tables

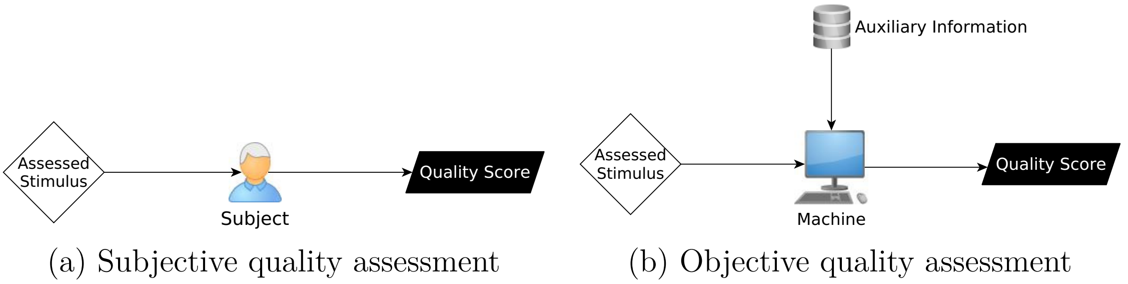
Chapter 1

Introduction

1.1 Overview

In the last years, there has been a huge progress in the usage of images and videos for an increasing number of applications. Multimedia services that have gained wide interest include digital television broadcasts, video games, and real-time video services over the Internet. According to predictions made by Cisco$^{\mathrm{TM}}$, the video portion on mobile data traffic was 53% in early 2014 and is expected to surpass 67% in 2018 [1]. With this increase in the demand of video contents, the need for tools for predicting quality of experience (QoE) is typically expected. Among the aspects that contribute to QoE, the quality of visual stimuli is one of the most important properties and can be altered in any stage of the multimedia communication chain, such as capturing, compression, transmission, reproduction, and displaying.

Since the genuine judges of visual quality are humans as end users, the essential strategy to measure visual quality consists of subjective experiments. Subjective experiments are typically performed in controlled laboratory environments and comprise a panel of human subjects which are usually non-experts. Subjects assess the quality of a given test stimuli, such as a sequence of images (or videos), as depicted in Fig. 1.1-(a). To perform the quality assessment, the subjects follow a set of planned steps. These steps are generally standardized to enable experimental reproducibility. For example, Recommendation ITU-R BT.500 [2] presents detailed information about several experimental parameters such as assessment methodologies, viewing conditions, grading scale, and timing. Experimental methodologies described in this recommendation include single and double stimulus methods. In single stimulus (SS), variants of the test videos are shown to subjects and no reference for comparison is provided. On the other hand, in double stimulus (DS), both reference and test stimuli are shown for comparison.



(a) Subjective quality assessment (b) Objective quality assessment

Figure 1.1: Visual quality assessment approaches: (a) using human beings or (b) using machines.

In subjective quality assessment, individual scores are given by the human subjects. After each subject assigns a score for the assessed stimulus, the scores are averaged to generate a mean opinion score (MOS), as illustrated in Fig. 1.2. The MOS is often a measure used to represent the overall quality of a visual stimuli. The Telecommunication Standardization Sector of International Telecommunication Union (ITU-T) defines several ways of referring to a MOS in Recommendation P.800.1 [3], depending on the type of the stimulus.

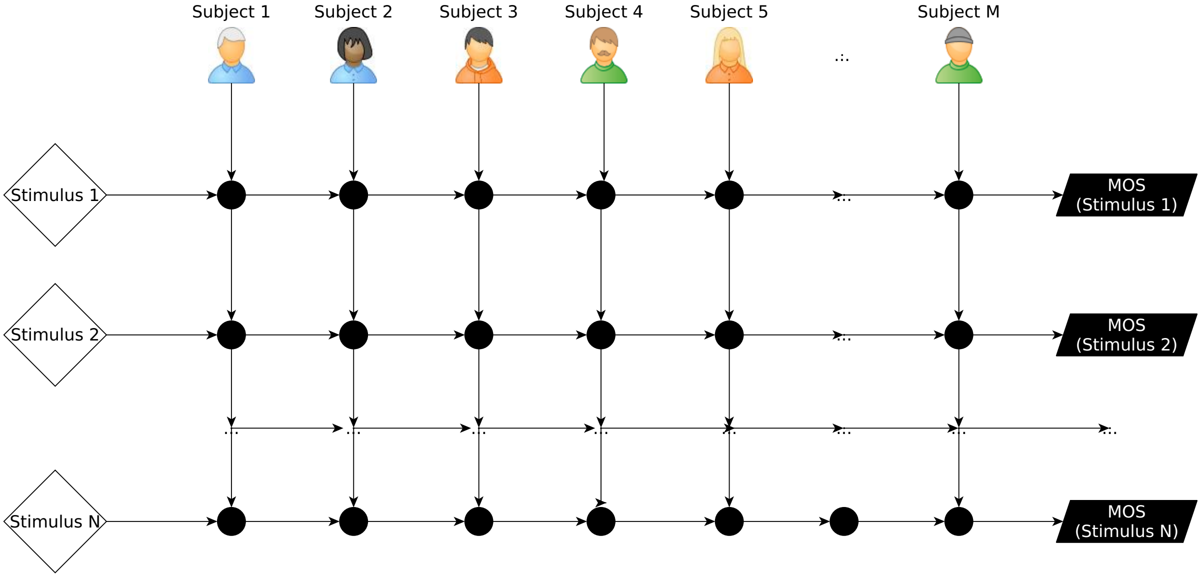


Figure 1.2: A common approach for subjective quality assessment.

Collecting MOS ratings is time-consuming, laborious, and expensive, since it requires the recruitment of human subjects. Due to the cost of subjective experiments, large efforts have been made to develop objective quality metrics [4, 5]. Objective quality metrics predict MOS automatically using a computational algorithm. Such automation makes it possible to implement fast and cheap procedures for monitoring and controlling the quality of visual stimuli, as illustrated in Fig. 1.1-(b).

Quality metrics are generally categorized into three types, depending on the amount of reference information required by the method [6, 7]. If the whole reference (original stimulus) is required to estimate the quality, the metric is classified as a full-reference (FR) method. When only partial information about the reference is used, the metric is classified as reduced-reference (RR) method. Since requiring even partial reference information can be a hindrance for several multimedia applications, in some cases the most convenient solution is to use no-reference (NR) methods, which blindly assess the quality of visual stimuli without needing any information about its reference [8].

Fig. 1.3 shows a block diagram depicting the steps of the FR, RR, and NR quality assessment methods. In the FR approach, the algorithm compares the assessed stimuli, probably distorted, with the original to produce a score that predicts the MOS of the assessed video. In the RR case, it is not necessary to access all the content of the original video. Instead, an intermediate step extracts significant visual features (e.g. texture statistics or other suitable characteristics of the reference). The quality assessment is performed by comparing the reduced reference information with the information of the assessed video. Finally, no-reference methods do not require access the original video, as shown in the block diagram.

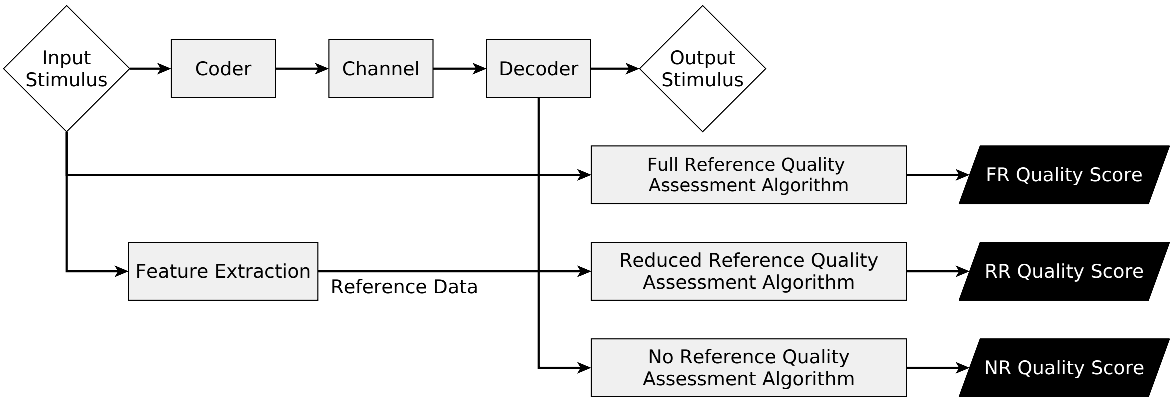

Input Stimulus
Coder
Channel
Decoder
Output Stimulus
Full Reference Quality Assessment Algorithm
FR Quality Score
Feature Extraction
Reference Data
Reduced Reference Quality Assessment Algorithm
RR Quality Score
No Reference Quality Assessment Algorithm
NR Quality Score

Figure 1.3: Types of objective quality assessment metrics: full-reference, reduced-reference, and no-reference (blind).

FR methods are among the most traditional quality metrics, including measures such as signal-to-noise ratio (SNR), peak-signal-to-noise ratio (PSNR), and mean squared error (MSE) [6]. These methods, although computationally simple, do not take into account the characteristics of human visual system (HVS) [9]. To overcome this limitation, more sophisticated FR methods incorporate HVS models [10, 11, 12, 13]. Several of these metrics are based on error sensitivity, attempting to analyze and quantify the error signal in a way that simulates the human quality judgment [14, 15, 13, 16, 17, 18]. Other FR methods incorporate HVS characteristics using *feature extraction* approaches [19, 20, 21, 22]. On the other hand, Lu *et al.* [24] showed that structural distortion-based FR methods for video quality assessment provide very promising results.

Webster *et al.* [25] proposed one of the first RR metrics. Their metric makes use of spatial and temporal activity features to assess the quality of videos. RR methods that use other types of features were proposed by Wang and Bovik [26], Bhateja *et al.* [27], Ma *et al.* [28], Wolf and Pinson [29], Farias [8], among others [30, 31, 32, 33]. These methods are generally less accurate than FR methods, but they are less complex, what makes them more suitable for real applications.

For applications which the reference video or even a small portion of it becomes a bottleneck, it becomes crucial to establish ways of estimating the quality of a visual stimuli using a NR method. Although NR metrics have been attracting a great attention in last decade [34, 35, 36, 37, 38, 39, 40, 41, 42, 43], their design is a challenge [44, 45]. Most proposed NR methods are based on *artifact measurements*, an approach that consists of analyzing the assessed video and estimating the amount visual degradations. This approach is often limited to specific artifacts and rarely can be used for general quality assessment.

1.2 Problem Statement

Designing a visual quality metric requires solving three major problems. The first problem is to determine a set of features that are relevant to visual quality. The second is to establish a pooling strategy for assessing visual quality over space and time. The third problem is how to create a model for mapping the pooled data into estimates of the subjective quality scores. Hemani and Reibman [46] have named these three problems as *measuring*, *pooling*, and *mapping*, respectively.

Measuring refers to the computation of the stimuli physical quantities. Examples of measured physical quantities include identification of artifacts to estimate the visual thresholds of those artifacts [35, 38, 42], quantification of edge sharpness [47], Prewitt filters [48], natural scene statistics [49, 50, 51, 52, 53, 54], statistics on curvlet domain [55], spatial and spectral entropies [56], subband statistics in the wavelet-packet domain [57], among other features [58, 59, 60, 61, 62, 63]. It is worth mentioning that multiple measurements can be used to generate a feature vector.

Pooling refers to the combination of the measurements, over a suitable subspace, to represent the quality of the stimuli. For example, for images, the spatial pooling over frequency and orientation produce a spatial map of responses. In videos, the pooling can be implemented over pixel space, frequency space, orientation, and time. Temporal pooling combines multiple frames into a single temporal score. For both image and video stimuli, a Minkowski summation is often used as a pooling strategy [42, 64, 65, 66, 67], given the low-level vision additivity property. Therefore, pooling strategies are often used to reduce the number of dimensions of measurements [68, 69, 70, 71, 72, 73]. Naturally, the chosen pooling strategy should consider how the HVS properties affect the physical measurements and their relationships among themselves.

The last problem to solve, when designing a quality metric, is the process of mapping the measurements to a quality score. This process generally uses a model to map the result of the pooling into an estimate of MOS. If the result of the pooling strategy is already linearly correlated with the subjective scores, the mapping is not necessary. The mapping model can be a pre-defined function, as adopted in Structural Similarity (SSIM) [74] and Gradient Magnitude Similarity Deviation (GMSD) [75], or automatically learned from the polled features, as adopted in most machine-learning (ML) based methods [60, 61, 62, 73].

Wang and Bovik [26] presented a concise review of methods of the image quality assessment (IQA) methods that are based on the quantification of distortions on natural scene statistics (NSS). Lin and Kuo performed a brief survey of IQA methods that use a categorization of features and of artifacts detection [76]. Similarly, reviews of objective methods of video quality assessment (VQA), which include a classification of these methods, were performed by Farias *et al.* [8, 77]. Vranjes *et al.* [78] classified the objective methods as data metrics, picture metrics, and bitstream metrics. Soundararajan and Bovik [79] performed a review of quality metrics based on information theory and concluded that this approach is rather limited for developing objective metrics.

Based on the work of Reibman *et al.* [80], the aforementioned methods were compiled into a categorization of visual quality assessment by Shahid *et al.* [4]. Fig. 1.4 outlines these three categories. Metrics are classified as either derived from pixel-based methods (PBM) or computed directly from the coded bit-stream (BBM). These two classes are not fully independent and some methods include aspects of both classes (hybrid methods).

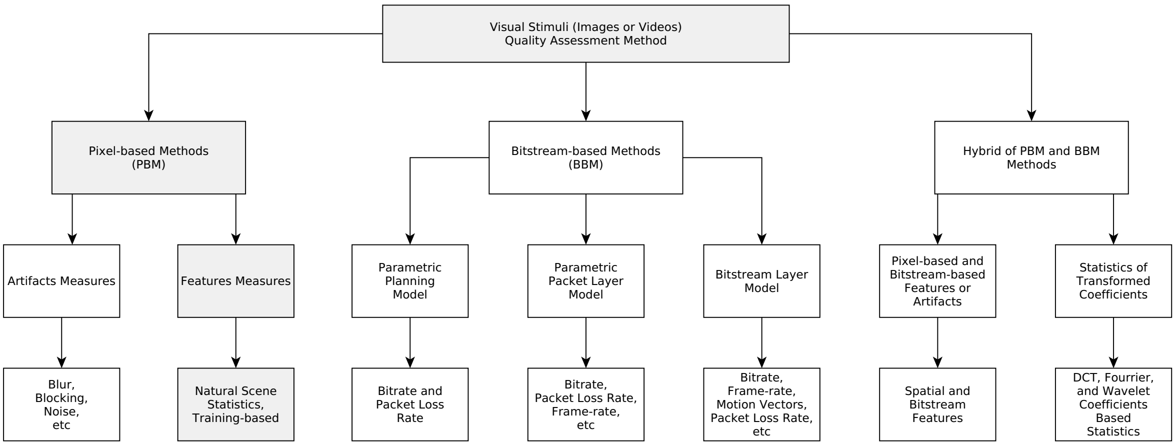

Visual Stimuli (Images or Videos)
Quality Assessment Method
Pixel-based Methods (PBM)
Bitstream-based Methods (BBM)
Hybrid of PBM and BBM Methods
Artifacts Measures
Features Measures
Parametric Planning Model
Parametric Packet Layer Model
Bitstream Layer Model
Pixel-based and Bitstream-based Features or Artifacts
Statistics of Transformed Coefficients
Blur, Blocking, Noise, etc
Natural Scene Statistics, Training-based
Bitrate and Packet Loss Rate
Bitrate, Packet Loss Rate, Frame-rate, etc
Bitrate, Frame-rate, Motion Vectors, Packet Loss Rate, etc
Spatial and Bitstream Features
DCT, Fourrier, and Wavelet Coefficients Based Statistics

Figure 1.4: Overview of visual quality assessment methods. The bottom row of boxes gives examples of data used as input for the approaches described in third row. The second row divides into three major categories proposed by Shahid *el al.* [4]

In the case of BBM, the quality can be computed in the absence of a decoder. In other words, the quality is computed without the need of decoding the compressed video. The BBM presents the advantage of simplicity, although they have a limited scope of application. They are usually designed for a specific coding standard or bitstream format, such as H.264/AVC or HEVC [4]. Such methods are based on either the information acquired from the bitstream or the packet header information (or both). They are adequate for network video applications, such as IPTV or video conferencing.

PBM methods are more complex than BBM methods because they take into consideration the visual content, by extracting features directly from pixels. As depicted in Fig. 1.4, PBM methods can use these features to predict the presence and strength of common artifacts or to analyze the impact that distortions have on NSS [4]. Quantification of artifacts has been used as a measure for several quality assessment methods [81, 8, 64, 82]. In this case, the quality values depend on a single artifact or a combination of artifacts. Artifact-based methods often use a degradation model obtained via simulations, which are not always comparable to degradations present in real scenarios [4]. Furthermore, artifact-based methods may be unable to assess the overall quality in the presence of other artifacts. Given these limitations of artifact-based methods, researchers have been working on PBM methods that do not make assumptions about specific artifacts [60, 61, 83, 84]. These methods generally perform an analysis of the statistical characteristics commonly found in natural (undistorted) stimuli.

It is worth pointing out that BBM methods can be used, instead of PBM methods, to reduce the overall computational complexity. The performance of BBM methods can be improved by adding measures obtained from PBM method. This hybrid approach inherits the computational simplicity of BBM and the flexibility and robustness of PBM.

1.3 Proposed Approach

In this work, we investigate the use of texture measurements for assessing the quality of images and videos. Considering the preceding discussion, the scope of the proposed metrics is highlighted in Fig. 1.4. We chose to adopt a feature-based approach because this type of approach does not require assumptions about the type of artifacts or content [4]. While artifact-based approaches use a prior knowledge to estimate quality, the chosen approach is more general and more practical in real-world applications. Feature-based approaches can be divided into (1) Natural scene statistics (NSS) [85, 51, 86, 87] and (2) Machine Learning-based (ML) approaches [61, 60, 88, 89]. NSS-based approaches are based on the hypothesis that the statistical properties of natural scenes are affected by distortions or artifacts [87]. Machine learning approaches, on the other hand, relies on a large number of features that are designed to capture relevant factors affecting visual quality. These features are not easily interpreted and choosing them is one of the main challenges of this approach.

Our approach follows the ML trend. We adopt a ML-based approach for VQA because (1) ML performance compares favorably with state-of-the-art quality metrics and (2) machine learning provides flexible and effective tools to support our final application [90]. Moreover, we based our methods on state-of-the-art IQA methods and generalize them to work for video quality. Our hypothesis is that if ML methods based on spatial textures provide a good performance for IQA, then using spatiotemporal texture-based learning methods should provide a good performance for VQA.

1.4 Summary of the Contributions

In summary, the main contributions of this work are:

- Development of new quality-aware texture operators;
- Development of a model for estimating the overall image quality without reference using the variants of LBP. Several image quality metrics can be generated from this model;
- Development of a full-reference metric for assessing video quality using variants of LBP in combination with other spatial and spatio-temporal features.

1.5 Organization of this Dissertation

This document is divided into five chapters. Chapter 2 describes the measurement of textures in order to generate image quality features. Chapter 3 presents IQA methods that use texture measurements to blindly estimate image quality. Chapter 4 presents a proved video quality assessment method that is also based on texture measurements. Finally, Chapter 5 summarizes the contributions of this work and discusses future works.

Chapter 2

Texture Measurements Based on Variants of Local Binary Patterns

2.1 Overview

Texture is a fundamental attribute of images. Although it is a ubiquitous concept, there is no a general consensus concerning an specific definition of texture. The etymological origin of the word ‘texture’, *texere*, means ‘to weave’ and indicates that it is associated with the combination of essential features to form a complex whole. Petrou and Garcia-Sevilla, for instance, define texture as a variation of the visual stimuli at scales smaller than the scale of interest [91]. Davies calls texture a pattern with both randomness and regularity [92]. In the context of this work, texture is the characteristic of an area, which is perceived as the combination of some basic patterns. These basic patterns present a certain regularity that appears in the statistical measures of the visual stimuli.

Texture analysis consists mainly of the process of feature extraction that uses texture information. To characterize a texture, the method identifies and selects a set of distinguishing and relevant features. Several texture analysis methods have been proposed in the past several decades, using a variety of texture feature extraction approaches [91, 92]. These approaches include gray level run-length (GLRLM) [93], gray level co-occurrence matrices (GLCM) [94], texture spectrum [95], textons [96], etc.

Among the popular texture descriptors used for feature extraction is the local binary patterns (LBP) [97]. This descriptor describes the local textures of an image by performing operations in each image pixel. The textures are labeled according to the relationships between each pixel and its neighbors. Because of its simplicity, this method had a big impact in the several applications of texture analysis, such as face recognition [98, 99], gender classification [100], among others.

One of the advantages of LBP descriptor is that it unifies traditional texture analysis models. Fig. 2.1 shows how the LBP is related with other common texture descriptors. In this figure, the arrows represent the connection between descriptors, while the texts over the arrows sums up the essential differences between these descriptors. As stated by Ahonen and Pietikäinen [101], LBP can be viewed as combination of local derivative filters whose outputs are quantized by binary thresholding.

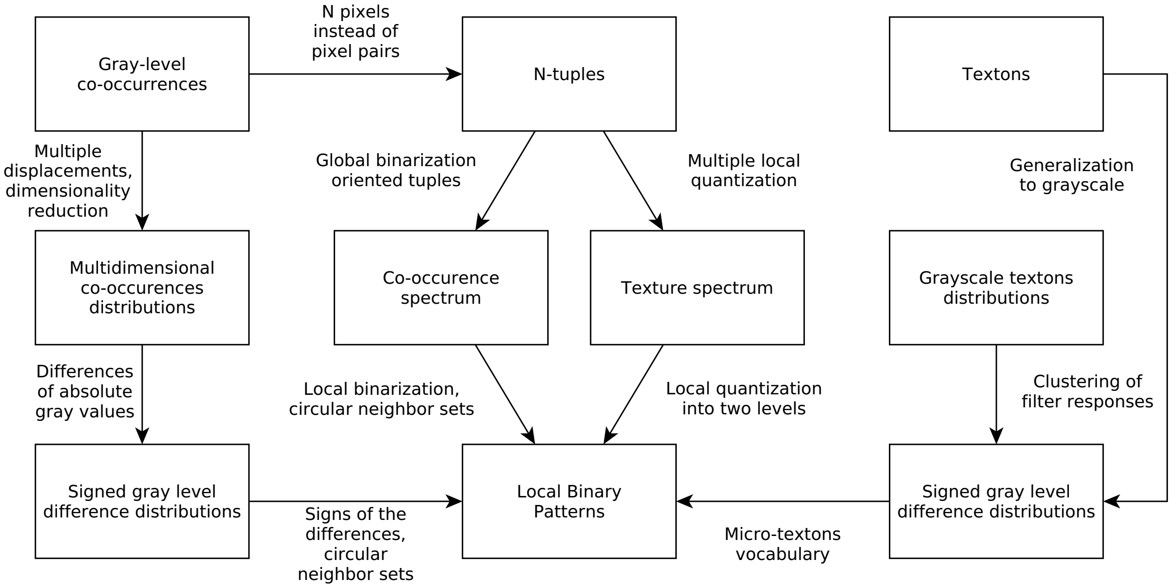
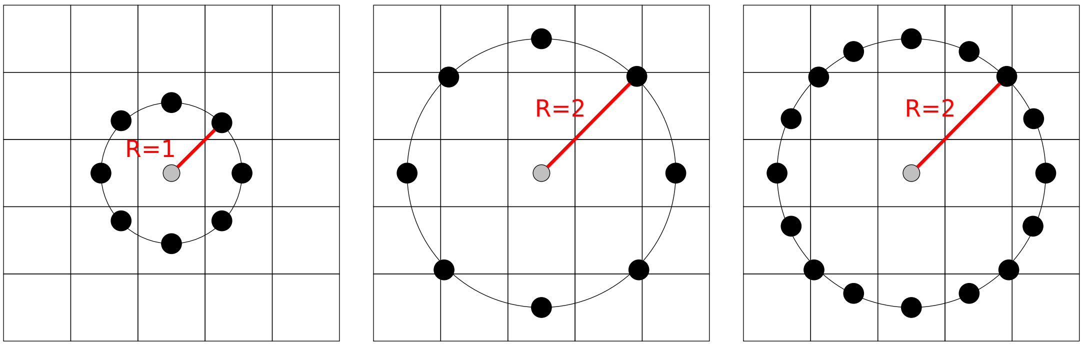

Gray-level co-occurrences
N pixels instead of pixel pairs
N-tuples
Textons
Multiple displacements, dimensionality reduction
Global binarization oriented tuples
Multiple local quantization
Generalization to grayscale
Multidimensional co-occurences distributions
Co-occurence spectrum
Texture spectrum
Grayscale textons distributions
Differences of absolute gray values
Local binarization, circular neighbor sets
Local quantization into two levels
Clustering of filter responses
Signed gray level difference distributions
Local Binary Patterns
Signed gray level difference distributions
Signs of the differences, circular neighbor sets
Micro-textons vocabulary

Figure 2.1: Relation of LBP to other texture methods [102].

Nowadays, we can found in the literature several modifications of the LBP descriptor [100, 103]. Most of them try to improve LBP performance in specific applications of texture analysis (e.g., texture classification, face recognition, object detection, etc). However, few works have investigated the applicability of each LBP variant at a specific application. This chapter is inspired by the work of Hadid *et al.* [100], who performed a comparison of the performance of 13 different LBP methods, focusing on gender recognition applications. This chapter describes the basic LBP operator in Section 2.2, reviews some of the state-of-the-art LBP variants in Section 2.3, and presents the proposed quality-aware LBP variants in Section 2.4. These variants are used as features for developing quality metrics in the next chapters.

2.2 Basic Local Binary Patterns (LBP)

The Local Binary Pattern (LBP) is arguably one of the most powerful texture descriptors. It was first proposed by Ojala *et al.* [97] and it has since been proven to be an effective



(a) $R = 1, P = 8$ (b) $R = 2, P = 8$ (c) $R = 2, P = 16$

Figure 2.2: Circularly symmetric P neighbors extracted from a distance R.

feature extractor for texture based problems. The traditional LBP operator takes the following form:

$$LBP_{R,P}(I_c) = \sum_{p=0}^{P-1} S(I_p - I_c)2^p, \tag{2.1}$$

where

$$S(t) = \begin{cases} 1, & \text{if } t \geq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

In eq. 2.1, $I_c = I(x, y)$ is an arbitrary central pixel at the position (x, y) and $I_p = I(x_p, y_p)$ is a neighboring pixel surrounding I_c, where

$$x_p = x + R\cos\left(2\pi\frac{p}{P}\right) \tag{2.3}$$

and

$$y_p = y - R\sin\left(2\pi\frac{p}{P}\right). \tag{2.4}$$

P is the total number of neighboring pixels I_p, sampled with a distance R from I_c. Fig. 2.2 illustrates examples of symmetric samplings with different numbers of neighboring points (P) and radius (R) values.

Fig. 2.3 illustrates the steps for applying the LBP operator on a single pixel ($I_c = 8$) located in the center of a 3×3 image block, as shown in the bottom-left of this figure. The numbers in the yellow squares of the block represent the order in which the operator is computed (counter-clockwise direction starting from 0). In this figure, we use an unitary neighborhood radius ($R = 1$) and eight neighboring pixels ($P = 8$). After calculating $S(t)$ (Eq. 2.2) for each neighboring pixel I_p, we obtain a binary output for each I_p ($0 \leq p \leq 7$), as illustrated in the block in the upper-left position of Fig. 2.3. In this block, black circles correspond to '0' and white circles to '1'. These binary outputs are stored in a binary format, according to their position (yellow squares). Then, the resulting binary number

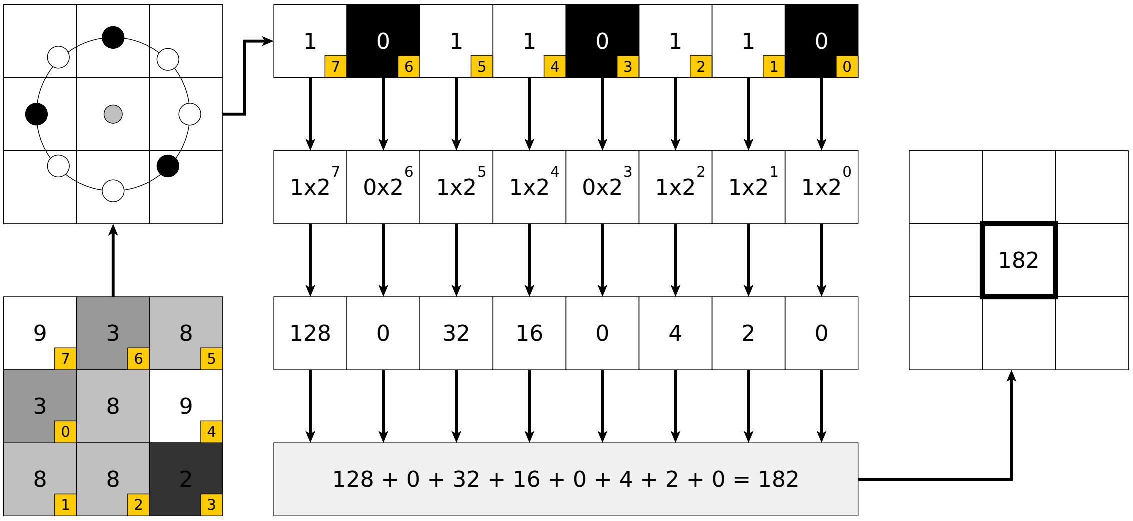


Figure 2.3: Calculation of LBP labels.

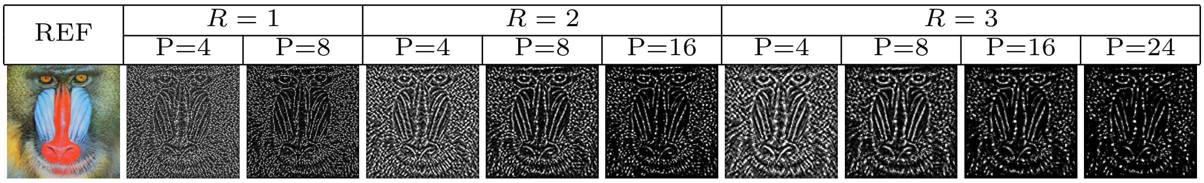


Figure 2.4: Reference image and its correspondent Local Binary Pattern (LBP) channels computed using three different radius (R) values.

is converted to the decimal format. This decimal number is the output generated by the LBP operator for I_c. After applying the operator for all pixels in an image, we obtain a set of labels, which is known as the *LBP channel*. Fig. 2.4 shows examples of LBP channels for the image 'Baboon', obtained using different radius values and number of neighboring points.

When an image is rotated, the I_p sampled values move along the perimeter of the circumference around I_c, generating a circular shift in the binary number generated. As a consequence, a different decimal $LBP_{R,P}(I_c)$ value is obtained. To remove this effect, we can use the following rotation invariant (ri) operator:

$$LBP^{ri}_{R,P}(I_c) = \min\{ROTR(LBP_{R,P}(I_c), k)\}, \tag{2.5}$$

where $k = \{0, 1, 2, \cdots, P-1\}$ and $ROTR(x, k)$ is the circular bit-wise right shift operator that shifts the tuple x by k positions.

Due to the crude quantization of the angular space and to the occurrence of specific frequencies in individual patterns, $LBP_{R,P}$ and $LBP^{ri}_{R,P}$ operators do not always provide a good discrimination [104]. To improve the discriminability, Ojala *et al.* [97] proposed an improved operator that captures fundamental pattern properties. These fundamental patterns are called ‘uniform’ and computed as follows:

$$LBP^{u}_{R,P}(I_c) = \begin{cases} \sum\limits_{p=0}^{P-1} S(I_p - I_c), & \text{if } U(LBP^{ri}_{R,P}) \leq 2, \\ P+1, & \text{otherwise,} \end{cases} \tag{2.6}$$

where

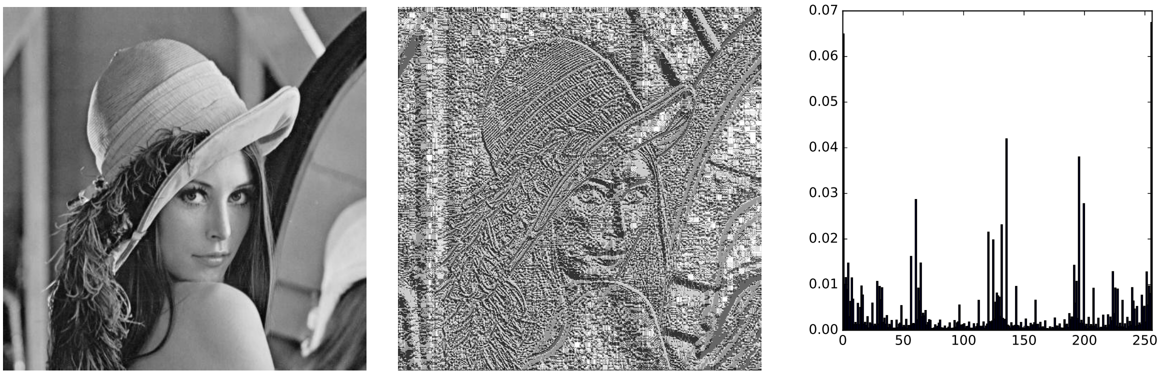
$$U(LBP_{P,R}) = \Delta(I_{P-1}, I_0) + \sum_{p=1}^{P-1} \Delta(I_p, I_{p-1}), \tag{2.7}$$

and

$$\Delta(I_x, I_y) = |S(I_x - I_c) - S(I_y - I_c)|. \tag{2.8}$$

In addition to a better discriminability, the uniform LBP operator described in Eq. 2.6 has the advantage of generating fewer distinct LBP labels.The ‘nonuniform’ operator (Eq. 2.1) produces 2^P different output values, while the ‘uniform’ operator produces only $P + 2$ distinct output values.

Finally, once calculated the LBP mask using any LBP approach above, we compute its histogram, as depicted in Fig. 2.5. Typically, the normalized LBP histogram is used as input feature vector to a machine learning algorithms[103].



(a) Lena image. (b) LBP map. (c) LBP histogram

Figure 2.5: Example of an input image, the corresponding LBP image and LBP histogram (extracted feature).

2.3 Variants of the LBP Available in the Literature

Because of its flexibility and performance, the LBP operators have become very popular. Nevertheless, the original version (presented in the previous section) has several limitations, what has inspired the development of several LBP modifications to better adapt it for different applications. In this section, we present some variants of the LBP operator that have been proposed with the goal of improving the robustness and discriminability of the original operator.

2.3.1 Local Ternary Patterns (LTP)

The LTP operator is an extension of the LBP operator that assumes up to 3 coded values ($\{-1, 0, 1\}$). This is achieved by changing the step function S in the following manner:

$$\hat{S}(t) = \begin{cases} 1, & t \geq \tau, \\ 0, & -\tau < t < \tau \\ -1, & t < -\tau, \end{cases} \tag{2.9}$$

where τ is a threshold which determines how sharp an intensity change should be in order to be considered as an edge. After computing the ternary codes using the above equation, each ternary pattern is split into two codes: a positive (upper pattern) and a negative (lower pattern) code. These codes are treated as two separate channels of LBP descriptors.

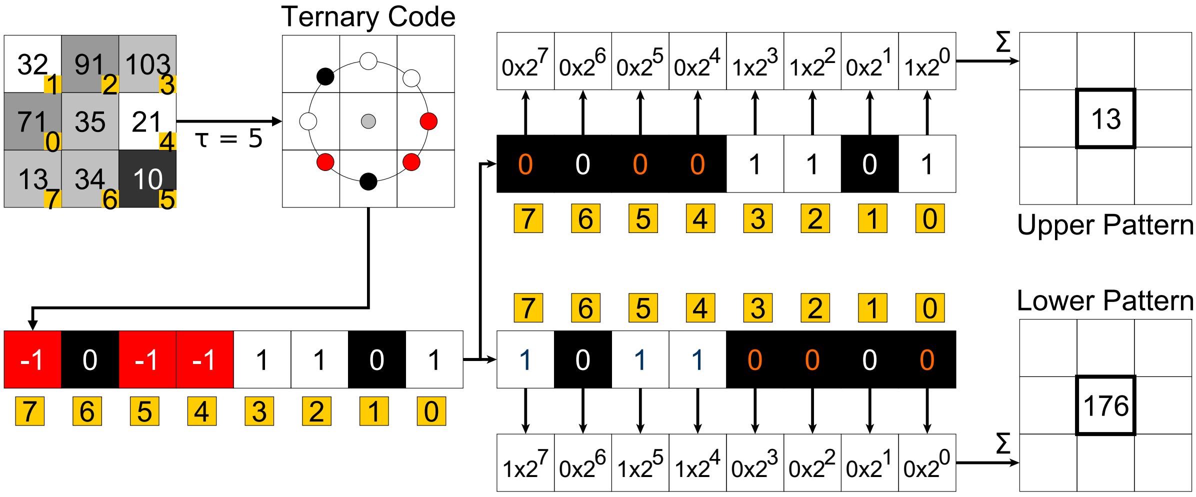


Figure 2.6: Illustration of the basic Local Ternary Pattern operator.

Fig. 2.6 illustrates the basic feature extraction procedure for a single pixel using LTP operator. The numbers in yellow squares represent the order in which the step function is computed (Eqs. 2.2 and 2.9). In this example, we consider an unitary neighborhood radius ($R = 1$), eight neighboring pixels ($P = 8$), and a threshold τ equal to five. While in the LBP the binary code takes only two values (0 or 1, represented by the colors black and white), the LTP operator generates three possible values (see Eq. 2.9) that are represented by the colors black ($\hat{S}(t) = 1$), white ($\hat{S}(t) = 0$), and red ($\hat{S}(t) = -1$).

Following the same counterclockwise order used for computing the LTP, we split the ternary code into two LBP codes with only positive values. First, we create the upper pattern by converting the negative codes to zero. Next, we create the lower pattern by setting the positives values to zero, converting the negative values to positive. By comparing Fig 2.6 and Fig 2.3, we notice that the LTP operator extends the LBP operator, generating two texture information maps. We treat these two maps as two separate LBP channels. Therefore, we compute independent histograms and similarity measures for each of these maps. Results are combined at the end of the process to generate the feature vector.

2.3.2 Local Phase Quantization (LPQ)

A limit of LBP descriptor is its relative sensitivity to blur. To tackle this problem, the local phase quantization (LPQ) descriptor is suggested for applications where blur insensitivity is demanded [105]. The LPQ descriptor performs a quantization of the Fourier transform phase in local neighborhoods. Assuming that $G(u)$ and $F(u)$ are the discrete Fourier transforms (DFT) of the blurred $g(z)$ and original $f(z)$ images, which are related by the following equation:

$$G(u) = F(u) \cdot H(u), \tag{2.10}$$

where $H(u)$ is the Fourier transform of the filter $h(z)$. The magnitude of $G(u)$ is given by:

$$|G(u)| = |F(u)| \cdot |H(u)|, \tag{2.11}$$

while its phase is given by:

$$\angle G(u) = \angle F(u) + \angle H(u). \tag{2.12}$$

Assuming that $h(x) = h(-x)$, its DFT is always real and the phase assumes only two values, namely:

$$\angle H(u) = \begin{cases} 0, & H(u) \geq 0 \\ \pi, & \text{otherwise.} \end{cases} \tag{2.13}$$

In LPQ, the phase is computed at the local neighborhood N_z, for each pixel position of $f(z)$. The local spectrum is computed with the following equation:

$$F(u, x) = \sum_{y \in N_z} f(y) w_R(y - x) e^{-j2\pi uy}, \tag{2.14}$$

where u is the frequency and w_R is a window given as

$$w_R(x) = \begin{cases} 1, & |x| < \frac{N_R}{2} \\ 0, & \text{otherwise.} \end{cases} \tag{2.15}$$

The local Fourier coefficients are computed at four frequencies for each pixel position, i.e.,

$$F(x) = \left[F(u_1, x), F(u_2, x), F(u_3, x), F(u_4, x)\right], \tag{2.16}$$

where $u_1 = [a, 0]^T$, $u_2 = [0, a]^T$, $u_3 = [a, a]^T$, and $u_4 = [a, a]^T$. In these cases, a is sufficiently small to satisfy $H(u_i) > 0$.

The phase information in Fourier coefficients is given by the signs of the real and imaginary parts of each component $F(x)$. This is done via scalar quantization:

$$q_j = \begin{cases} 1, & g_j \geq 0 \\ 0, & \text{otherwise,} \end{cases} \tag{2.17}$$

where g_j is the j-th component of $G(x) = [Re\{F(x)\}, Im\{F(x)\}]$. After generating the binary coefficients q_j, we follow the same steps of the LBP descriptor.

2.3.3 Binarized Statistical Image Features (BSIF)

The binarized statistical image features (BSIF) is a descriptor proposed by Kannala and Rahtu [106] inspired by LBP and LPQ. However, in contrast to LBP and LPQ approaches, BSIF do not use a manually predefined set of filters but learn the filters by employing statistics of natural images. BSIF are among the best texture analysis techniques applied for face recognition and texture classification [100, 106].

Differently from the previous descriptors, which operate on pixels, BSIF works on patches of pixels. Given an image patch X of size $l \times l$ pixels and a linear symmetric filter W_i of the same size, the filter response s_i is obtained computing the following expression:

$$s_i = \sum_{u,v} W_i(u, v) X(u, v) = w_i^T x, \tag{2.18}$$

where vectors w and x contain the pixels of W_i and X, respectively. The binarized feature is acquired using the following function:

$$b_i = \begin{cases} 1, & s_j > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2.19}$$

The filters W_i are learned via independent component analysis (ICA). The binarized features b_i are aggregated following the same procedure described for generating the LBP labels. The descriptive features are obtained by computing the histogram of the aggregated data.

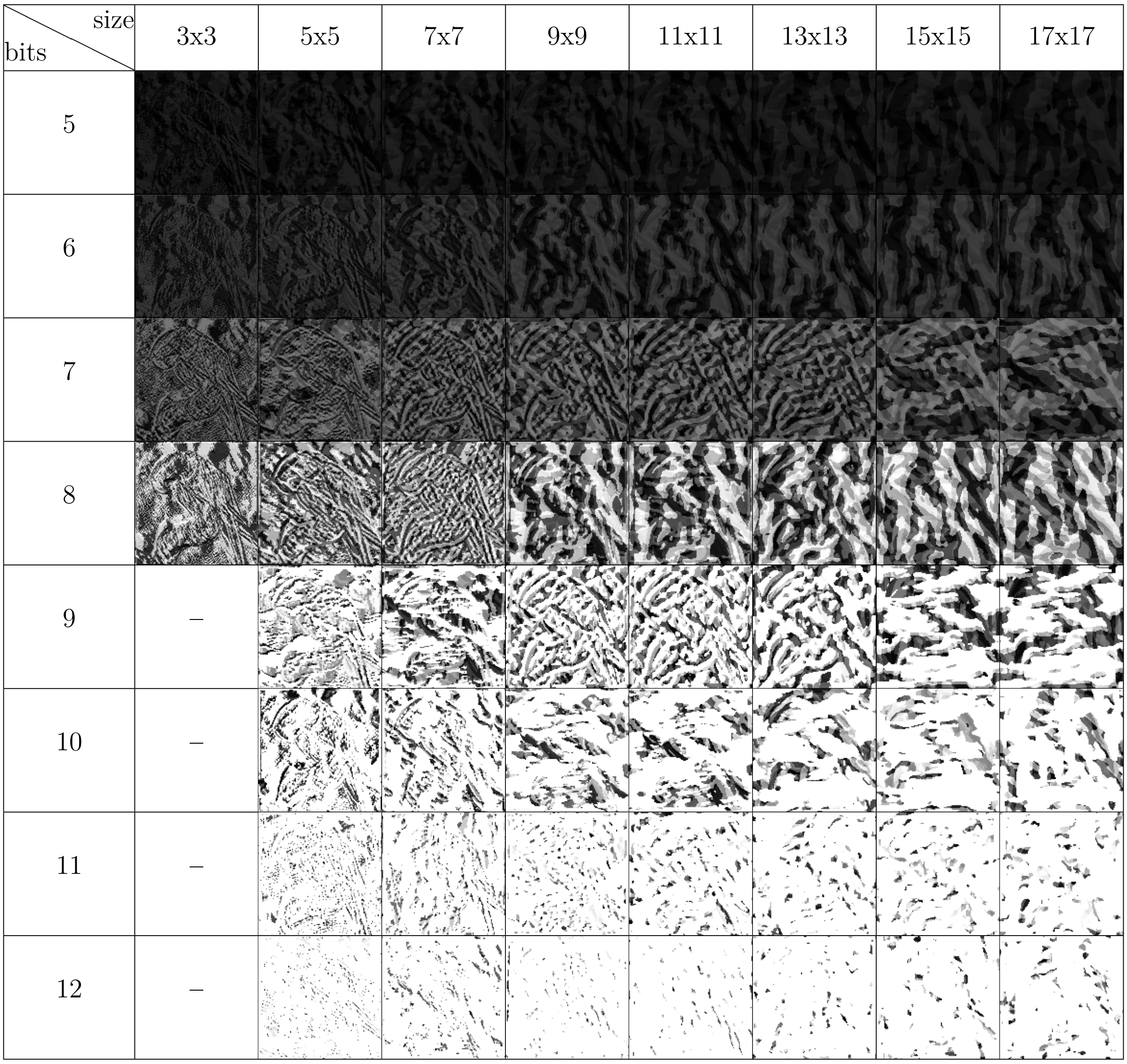


Figure 2.7: BSIF code images at different scales.

Similarly to the LBP descriptor, which generates the LBP channels, the BSIF descriptor generates coded images. These coded images are the set of labels generated after the binarized features are computed using Eq. 2.19 and aggregated using Eq. 2.1. Notice that the aggregation of BSIF results are based on a selected number of bits, instead of the number of neighbors of the labeled pixel. As mentioned previously in this section, labeling does not depends on the neighborhood. It depends on the relationship between the patch size l and the amount of binarized features b_i.

Fig. 2.7 shows the BSIF coded images, corresponding to the reference image depicted in Fig. 2.18. As can be seen in this figure, the textured information depends of the patch size l and on the number of bits. The number of bits is less or equal $l^2 - 1$. This is the reason why the second column does not contain BSIF coded images for 9, 10, 11, or 12 bits. Fig. 2.7 also shows that the choice of the number of bits and the patch sizes is an important step for texture analysis algorithms. Therefore, multiscale approaches that incorporate combinations of these parameters are recommended for machine learning approaches [107, 108, 109, 110].

2.3.4 Rotated Local Binary Patterns (RLBP)

For some applications, such as iris and fingerprint recognition, rotation variations in LBP results occur because of the fixed order of the weights. Since weights are distributed in a circular fashion, the effect of rotation can counterfeit by rotating the weights by the same angle. In cases where the angle of the rotation is not known, an adaptive arrangement of weights, based on the locally computed reference direction, must be determined. With this goal, Mehta and Egiazarian [111] proposed the rotated local binary (RLBP) descriptor, considering that, if an image is subjected to a rotation, it should also undergo a rotation by the same angle.

The idea of RLBP is to make the LBP invariant to rotation by circularly shifting the weights according to the dominant direction (D). In a neighborhood of a pixel I_c, D is the index of the neighbor whose difference to I_c is maximum, i.e.:

$$\mathrm{D} = \underset{p \in \{0,1,\cdots,P-1\}}{\operatorname{argmax}} |I_p - I_c|. \tag{2.20}$$

Since D is taken as reference, the weights are assigned with respect to it. In this manner, the RLBP operator is computed as follows:

$$RLBP_{R,P} = \sum_{p=0}^{P-1} S(I_p - I_c) 2^{(p-D \bmod P)}, \tag{2.21}$$

where $i \pmod{j}$ is the modulus operation that finds the remainder after the division of i by j.

Fig. 2.8 depicts the effect of the rotation on the LBP and RLBP operators. Notice that the LBP operator provides different values for a simple rotation. The red color indicates the pixels that have values above the threshold, while the yellow color indicates the pixels corresponding to D (maximum difference to I_c). The bit corresponding to D takes the smallest weight. The other weights are circularly shifted in relation to it. From Fig. 2.8-(g), we can notice that the weight corresponding to D is the same both for the original and rotated images, even when these pixels are at different angles. Therefore, the RLBP values acquired for two distinct rotated neighborhoods are the same.

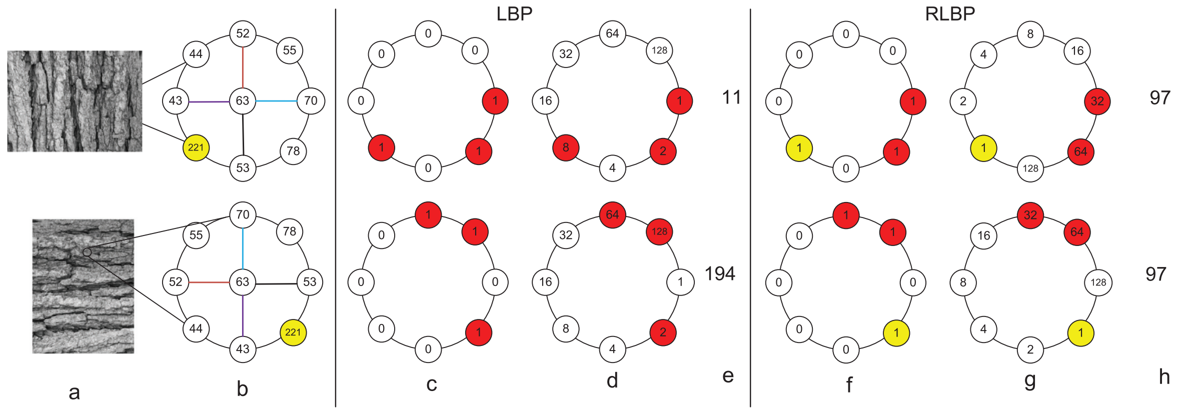


Figure 2.8: Rotation effect on LBP and RLBP descriptors: (a) original image and its rotated version, (b) Illustration of the neighbors rotation for the same pixel '63', (c) Thresholded neighbors, values above threshold are shown in red color, (d) The weights corresponding to the thresholded neighbors, (e) LBP values, (f) Thresholded neighbors for RLBP with reference denoted in yellow color, (g) The weights of the thresholded neighbors, (h) The RLBP values for the original and rotated image is same [112].

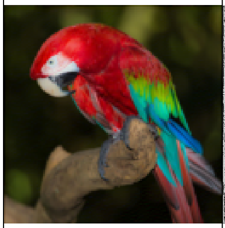
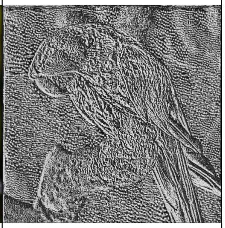
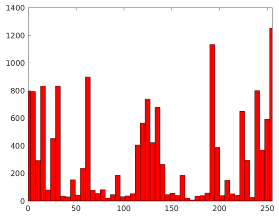
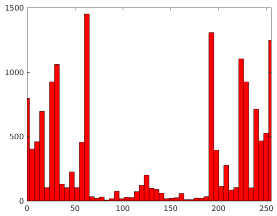
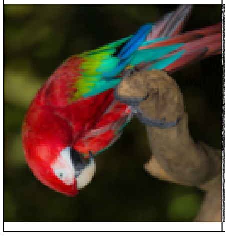
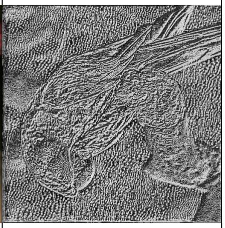
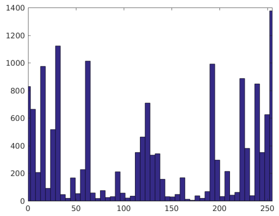
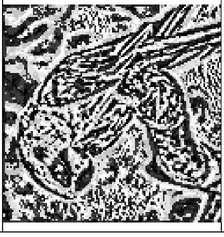
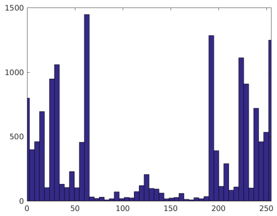
Input	LBP		RLBP	
	Map	Histogram	Map	Histogram
				
				

Figure 2.9: Effect of rotation on LBP and RLBP information.

Fig. 2.9 shows the effect of rotation after generating the LBP and RLBP channels. The first row of this figure shows the reference image used as input, the LBP map, the histogram of the LBP labels, the RLBP map, and the histogram of the RLBP values. The second row shows the same information computed after rotating the original image by 90 degrees. By comparing the LBP histogram before and after the rotation, we can observe a visual difference between them. Nevertheless, the corresponding RLBP histograms do not reflect these differences.

To illustrate the differences between the LBP and RLBP histograms shown in Fig. 2.9, we compute three statistical divergences measures: Kullback-Leibler divergence (KLD) [113], Jensen-Shannon divergence (JSD) [114], and chi-square distance (CSD) [115]. The KLD, JSD, and CSD of the LBP histograms are 2.92×10^{-2}, 6.96×10^{-3}, and 2.11×10^{2}, respectively. These divergences for RLBP are 2.06×10^{-4}, 5.12×10^{-5}, and 1.57, respectively. From these values, we notice that the order of magnitude of the LBP statistical divergences is two times higher than for the RLBP statistical divergences. These differences between demonstrate the rotation invariance of the RLBP operator.

2.3.5 Complete Local Binary Patterns (CLBP)

As described in Section 2.2, the LBP operator considers only the local differences of each pixel and its neighbors. The complete local binary patterns considers both signs (S) and magnitude (M) of the local differences, as well as the original intensity value of the center pixel [116]. Therefore, the CLBP feature is a combination of three descriptors, namely $CLBP_S$, $CLBP_M$, and $CLBP_C$. The overall view of CLBP feature computation is illustrated in Fig. 2.10.

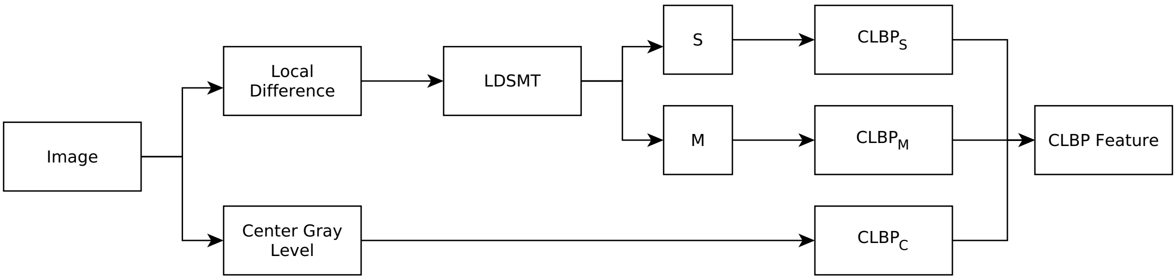


Figure 2.10: Framework of CLBP operator [116].

The $CLBP_S$ and $CLBP_M$ components are computed using the local difference sign-magnitude transform (LDSMT), which is defined as:

$$LDSMT_p = s_p \cdot m_p, \tag{2.22}$$

where $s_p = S(I_p - I_c)$ and $m_p = |I_p - I_c|$. The s_p is the sign operator used to compute $CLBP_S$, i.e. $CLBP_S$ is the same as the original LBP and it is used to code the sign information of the local differences. $CLBP_M$ is used to code the magnitude information of local differences:

$$CLBP_M = \sum_{p=0}^{P-1} thresh(m_p, c) \cdot 2^p, \tag{2.23}$$

where

$$thresh(x,c) = \begin{cases} 1 & x \geq c, \\ 0 & \text{otherwise.} \end{cases} \tag{2.24}$$

In the above equation, c is a threshold set as the mean value of the input image I. Finally, the $CLBP_C$ is used to code the information of original center gray level value:

$$CLBP_C = thresh(I_c, c). \tag{2.25}$$

The three operators, $CLBP_S$, $CLBP_M$ and $CLBP_C$, are combined. Individual histograms are computed and concatenated. This joint histogram is used as a CLBP feature.

2.3.6 Local Configuration Patterns (LCP)

Local configuration patterns (LCP) is a rotation invariant image descriptor proposed by Guo *et al.* is a LBP extension, which is more discriminative [117]. LCP decomposes the information architecture of the images into two levels: local structural information and microscopic configuration information. The local structural information are basically the LBP features, as described in Section 2.2. The microscopic configuration (MiC) information involves image configuration and pixel-wise interaction relationships.

To model the image configuration with respect to each pattern, we estimate optimal weights, associated with intensities of neighboring pixels, to linearly reconstruct central pixel intensity for each pattern type. This can be expressed by the following equation:

$$E(a_0, a_1, \cdots, a_{P-1}) = |I_c - \sum_{p=0}^{P-1} a_p I_p|, \tag{2.26}$$

where I_c and I_p denote the intensity values of the center pixel and neighboring pixels, a_p are weighting parameters associated with I_p, and $E(a_0, a_1, \ldots, a_{P-1})$ are the reconstruction errors with respect to the model parameters. To minimize the reconstruction errors, the optimal parameters for each pattern are determined by a least squares estimation .

Suppose the occurrence of a particular pattern type j is f_j. There are f_j pixels in the image with the pattern j. We denote intensities of those f_j pixels as $c_{j,i}$, where $i = 0, 1, \ldots, f_j - 1$. These intensities $c_{j,p}$ are organized into a vector:

$$\mathbf{c_j} = \begin{bmatrix} c_{j,0} \\ c_{j,1} \\ \vdots \\ c_{j,f_j-1}. \end{bmatrix} \tag{2.27}$$

We denote the intensities of neighboring pixels with respect to each $c_{j,i}$ as $v_{i,0}, \ldots, v_{i,P-1}$, which are organized into a matrix with the following form:

$$\mathbf{V_j} = \begin{bmatrix} v_{0,0} & v_{0,1} & \cdots & v_{0,P-1} \\ v_{1,0} & v_{1,1} & \cdots & v_{1,P-1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{f_j-1,0} & v_{f_j-1,1} & \cdots & v_{f_j-1,P-1} \end{bmatrix} \tag{2.28}$$

To minimize the reconstruction error (Eq. 2.26), the unknown parameters a_p are organized as a vector:

$$\mathbf{A_j} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{P-1}, \end{bmatrix} \tag{2.29}$$

and the optimal parameters are determined by solving the following equation:

$$\mathbf{A_j} = \left(\mathbf{V_j}^\top \mathbf{V_j}\right)^{-1} \mathbf{V_j}^\top \mathbf{c_j}. \tag{2.30}$$

After determining $\mathbf{A_j}$, we apply the Fourier transform to the estimated parameter, which can be expressed by:

$$\mathbf{H_j}(k) = \sum_{p=0}^{P-1} \mathbf{A_j}(p) e^{\frac{-i2\pi kp}{P}}, \tag{2.31}$$

where $\mathbf{H_j}(k$ is the k-th element of $\mathbf{H_j}$ and $\mathbf{A_j}(p)$ is the p-th element if $\mathbf{A_j}$. The magnitude part of each element of vector $\mathbf{H_j}$ is taken as the resulting MiC, which is defined by:

$$|\mathbf{H_j}| = [|\mathbf{H_j}(0)|, |\mathbf{H_j}(1)|, \ldots, |\mathbf{H_j}(P-1)|]\,. \tag{2.32}$$

The LCP feature is formed by both pixelwise interaction relationships and local shape information, which is expressed as:

$$LCP = [[|\mathbf{H_0}|; O_0], [|\mathbf{H_1}|; O_1], \ldots, [|\mathbf{H_{P-1}}|; O_{P-1}]]\,, \tag{2.33}$$

where $|\mathbf{H_j}|$ is computed using Eq. 2.32 with respect to the j-th pattern and O_j is the number of occurrences of the j-th LBP label.

2.3.7 Opposite Color Local Binary Patterns (OCLBP)

Although the LBP descriptor is efficient for describing grayscale textures, it is not sensitive to some types of impairments, such as contrast distortions or chromatic aberrations. As discussed by Maenpaa *et al.* [118], color and texture have complementary roles. When texture descriptors on luminance domain (e.g. LBP) obtain good results, color descriptors can also obtain good results. However, when color descriptors fail, luminance texture descriptors still can produce a good performance. Therefore, operators that combine both texture and color information are more effective in predicting a wider range of impairments.

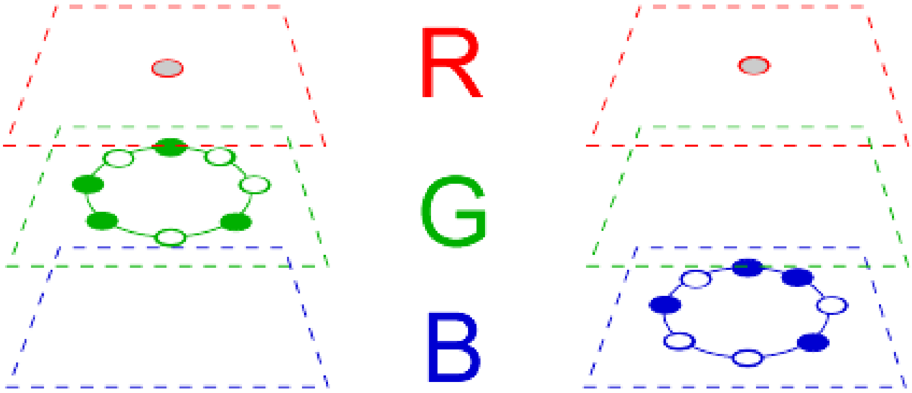
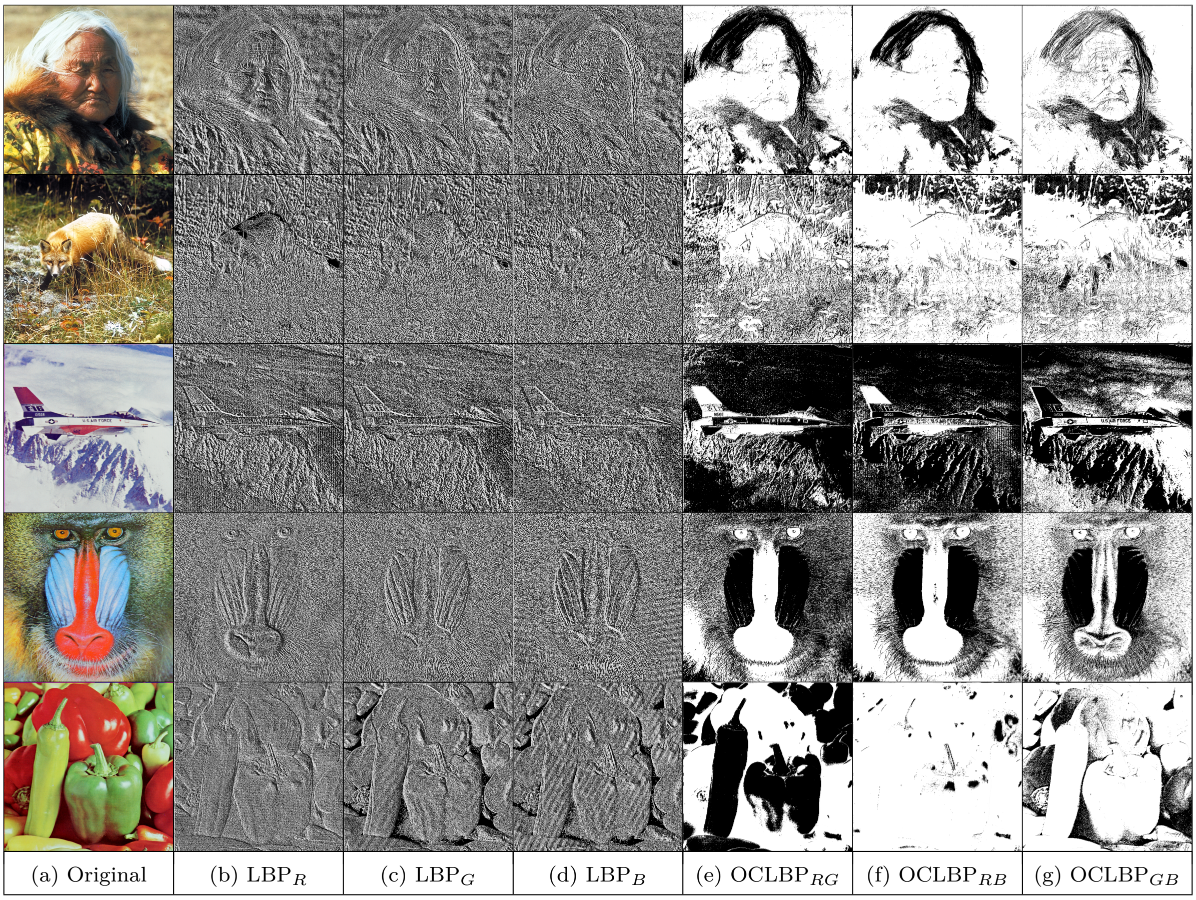


Figure 2.11: Sampling scheme for the OCLBP$_{RG}$ and OCLBP$_{RB}$ descriptors.

To combine both texture and color information into a joint descriptor, Maenpaa [119] proposed to use the Opponent Color Local Binary Pattern (OCLBP) operator. This operator improves the operator proposed by Jain & Healey [120] by substituting the Gabor filter with a variant of the LBP operator, which decreases the computational cost of the method.

The OCLBP operator has two approaches. In the first, the LBP operator is applied, individually, on each color channel, instead of being applyied only on a single luminance channel. This approach is called 'intra-channel' because the central pixel and the corresponding sampled neighboring points belong to the same color channels.

In the second approach, called 'inter-channel', the central pixel belongs to a color channel and its corresponding sampled neighboring points belong to another color channel. More specifically, for an OCLBP$_{MN}$ operator, the central pixel is positioned in the channel M, while the neighborhood is sampled in the channel N. For a three-channel color space, such as RGB, there are six possible combinations of channels: OCLBP$_{RG}$, OCLBP$_{RG}$, OCLBP$_{RB}$, OCLBP$_{RB}$, OCLBP$_{GB}$, and OCLBP$_{GB}$.



(a) Original (b) LBP$_R$ (c) LBP$_G$ (d) LBP$_B$ (e) OCLBP$_{RG}$ (f) OCLBP$_{RB}$ (g) OCLBP$_{GB}$

Figure 2.12: Original images and their output channels, computed using the OCLBP operator.

Fig. 2.11 depicts the sampling approach of OCLBP when the central pixel is sampled in R channel. From this figure, we can notice that two combinations are possible: OCLBP$_{RG}$ (left) and OCLBP$_{RB}$ (right). In this OCLBP$_{RG}$, the gray circle in the red channel is the central point, while the green circles in the green channel correspond to '0' sampling points and the white circles correspond to '1' sampling points, respectively. Similarly, in the OCLBP$_{RB}$ the blue circles correspond to '0' sampling points and the white circles correspond to '1' sampling points, respectively.

After computing the OCLBP operator for all pixels, a total of six texture channels are generated. As depicted in Fig. 2.12, three LBP intra-channels (LBP$_R$, LBP$_G$, and LBP$_B$) and three LBP inter-channels (OCLBP$_{RG}$, OCLBP$_{RB}$, and OCLBP$_{GB}$) are generated. Although all possible combinations of the opposite color channels allows six distinct channels, we observed that the symmetric opposing pairs are very redundant (e.g. OCLBP$_{RG}$ is equivalent to OCLBP$_{GR}$). Due to this redundancy, only the three more descriptive inter-channels are used.

2.3.8 Three-Patch Local Binary Patterns (TPLBP)

Wolf *et al.* [121] proposed a family of LBP-related descriptors each designed to encode additional types of local texture information. While variants of LBP descriptor use short binary strings to encode information about local micro-texture pixel-by-pixel, the authors considered different ways of using bitstrings to encode the similarities between patches of pixels, possibly capturing information which is complementary to that computed pixel-by-pixel. These patch-based descriptors are named Three-Patch LBP (TPLBP) and Four-Patch-LBP (FPLBP).

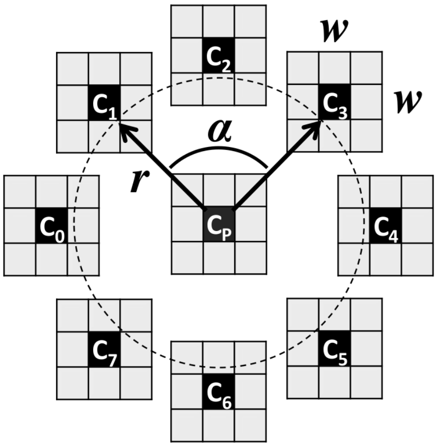


Figure 2.13: The Three-Patch LBP code with $\alpha = 2$, $w = 3$ and $\mathscr{S} = 8$ [121].

TPLBP considers a $w \times w$ patch centered on a pixel and and $\mathscr{S}$ additional patches distributed uniformly in a ring of radius r around it, as illustrated in Fig. 2.13. For a angle α, we get a set of neighboring patches along a circle and compare their values with those of the central patch. More specifically, the TPLBP is given by:

$$TPLBP_{r,S,w,\alpha}(p) = \sum_{i=0}^{S} \mathfrak{f}(\mathfrak{d}(C_i, C_p) - \mathfrak{d}(C_{i+\alpha \bmod S}, C_p)) \cdot 2^i, \tag{2.34}$$

where

$$\mathfrak{f}(t) = \begin{cases} 1, & \text{if } t \geq \tau, \\ 0, & \text{otherwise.} \end{cases} \tag{2.35}$$

The function $\mathfrak{d}(x, y)$ is any distance function between two patches under a vector representation. Examples of $\mathfrak{d}(x, y)$ are Manhattan [122], Mahalanobis [123], Minkowski [124], etc. The parameter τ is slightly larger than zero to provide some stability in uniform regions.

2.3.9 Four-Patch Local Binary Patterns (FPLBP)

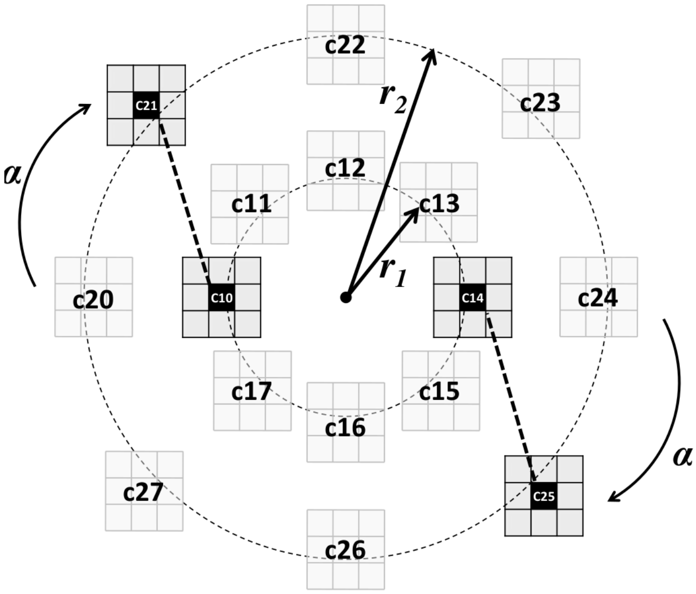


Figure 2.14: The Four-Patch LBP code with $\alpha = 1$, $w = 3$ and $\mathscr{S} = 8$ [121].

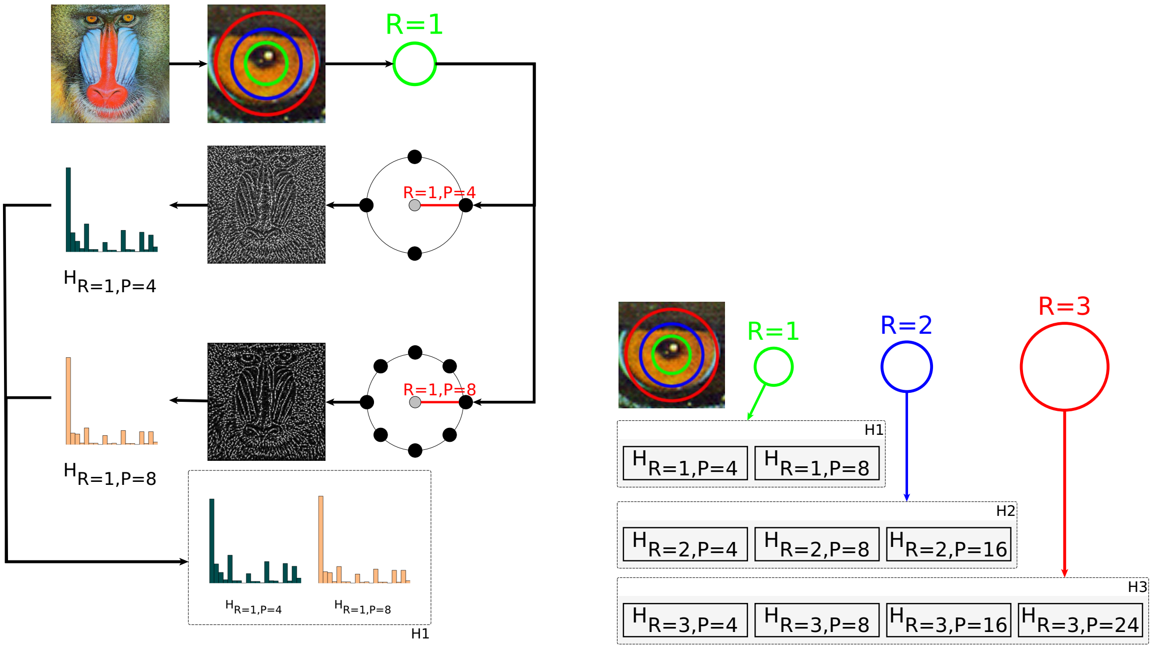
In FPLBP, two rings centered on the pixel are used, instead of only one ring as used in TPLBP. As depicted in Fig. 2.14, two rings of radii r_1 and r_2 (centered in the central pixel) are considered, with $\mathscr{S}$ patches of size $w \times w$ equally distributed on each ring, positioned α patches away along the circle. We compare the two center symmetric patches in the inner ring with the two center symmetric patches in the outer ring. The bit in each coded pixel is set according to which of the two pairs is being compared. Therefore, the FPLBP code is computed as follows:

$$FPLBP_{r,S,w,\alpha}(p) = \sum_{i=0}^{\frac{S}{2}} \mathfrak{f}(\mathfrak{d}(C_{1,i}, C_{2,i+\alpha \bmod S}) - \mathfrak{d}(C_{1,i+S/2}, C_{2,i+S/2+\alpha \bmod S})) \cdot 2^i. \quad (2.36)$$

2.4 Variants of the LBP Proposed in this Work

2.4.1 Multiscale Local Binary Patterns (MLBP)

The Multiscale local binary pattern (MLBP) is an extension of LBP operator proposed in this work [125], with the goal of extracting image quality information. A block diagram of the MLBP operator is depicted in Fig. 2.15. It is computed as follows: first, we generate several LBP channels, by varying the parameters R and P and performing a symmetrical sampling. For the smallest possible radius, $R = 1$, there are two possible P values that produce rotational symmetrical sampling ($P = 4$ and $P = 8$). When $R = 2$, there are three possible P values ($P = 4$, $P = 8$, and $P = 16$). In general, for a given radius R, there is a total of $R + 1$ distinct LBP channels.



(a) Multipoint LBP sampling.

(b) Multiple histogram generation from LBP.

Figure 2.15: Feature extraction steps.

Fig. 2.15(a) depicts the feature extraction for $R = 1$. The unitary radius generates only two distinct symmetrical patterns ($P = 4$ and $P = 8$). Each pattern generates a distinct LBP channel (see Fig. 2.4). For a radius R, LBP maps are generated and combined:

$$L_R = \{LBP^u_{R,4}, LBP^u_{R,8}, LBP^u_{R,16}, \cdots, LBP^u_{R,8R}\}, \tag{2.37}$$

where $LBP^u_{R,P}$ is computed according to Eq. 2.6 and L_R contains $R + 1$ elements. From these LBP channels, the texture features are obtained by computing the histogram of each member of L_R:

$$H_{R,P} = \left[h_{R,P}(l_1), h_{R,P}(l_2), \cdots, h_{R,P}(l_{P+2})\right], \tag{2.38}$$

where

$$h_{R,P}(l_i) = \sum_{x,y} \delta(LBP^u_{R,P}(x,y), i), \tag{2.39}$$

and

$$\delta(s,t) = \begin{cases} 1 & s = t, \\ 0 & \text{otherwise}. \end{cases} \tag{2.40}$$

In the above equations, (x, y) indicates the position of a given point of $LBP^u_{R,P}$ and l_i is the i-th LBP label. Notice that we are using 'uniform' LBP operators (Eq. 2.6) since their histograms provides a better discrimination of the texture properties.

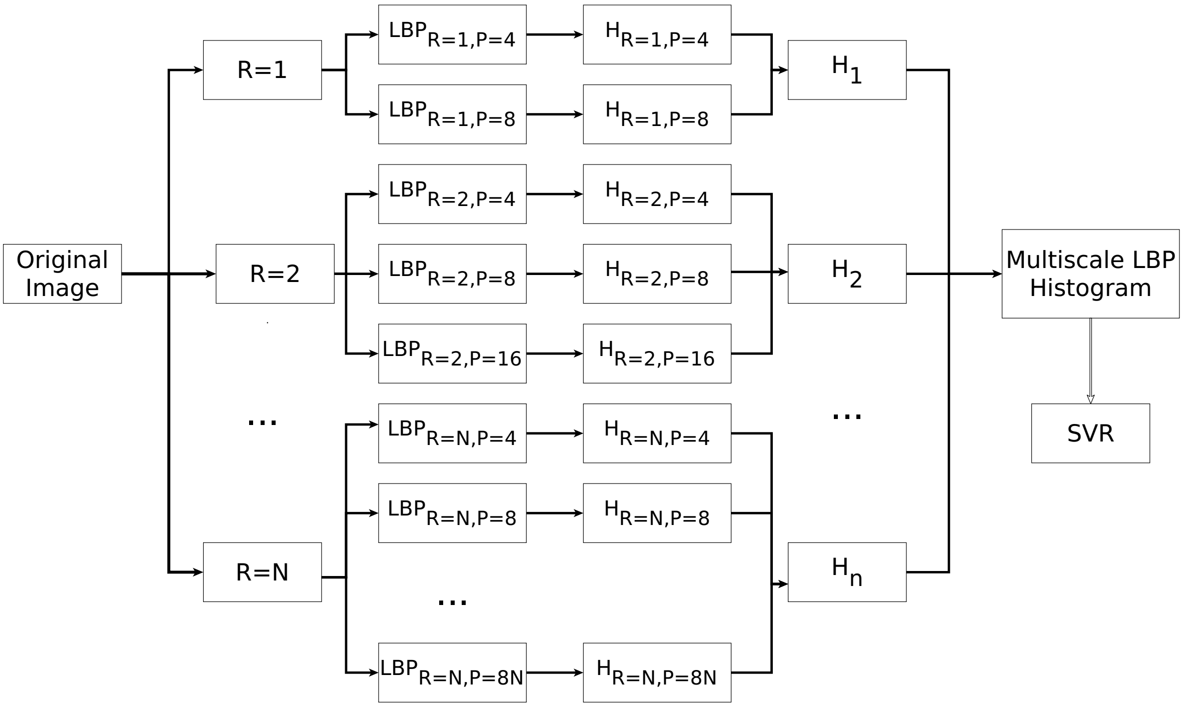


Figure 2.16: Feature extraction using MLBP histograms.

To obtain the feature vector, we vary the radius, compute all possible symmetric LBP patterns and their histograms, as illustrated in Fig. 2.15(b). For a radius R, we generate a vector of histograms by concatenating all individual LBP histograms:

$$H_R = H_{R,4} \oplus H_{R,8} \oplus H_{R,16} \oplus \cdots \oplus H_{R,8R}, \tag{2.41}$$

where $\oplus$ denotes the concatenation operator.

The steps for computing the multiscale LBP histogram are summarized in Fig. 2.16. For $R = N$, the final feature vector is generated by concatenating the histograms of the LBP channels with radius values smaller than N:

$$x = x_N = H_1 \oplus H_2 \oplus H_3 \oplus \cdots \oplus H_N, \tag{2.42}$$

where $R = N$ is the maximum radius value and x_N is the feature vector used to compute the histogram.

2.4.2 Multiscale Local Ternary Patterns (MLTP)

In general, LTP parameters must be adjusted for the target application. One important parameter that needs to be adequately chosen is the threshold τ in Eq. 2.9. This threshold was proposed by Opitz *et al.* [126], who estimated local thresholds from the directional gradient magnitude image. Anthimopoulos *et al.* [127] demonstrated that the τ values correspond to the gradient of the image. According to Anthimopoulos *et al.* [127], the choice of the threshold τ affects the discrimination between edge and non-edge pixels, which is an important step in the analysis of the edge patterns. Based on that, we propose [128] to choose an optimal set of thresholds for the multilevel edge description operation, making it possible to group gradient PDFs in clusters. The procedure is described as follows. First, the image gradients are fit using an exponential distribution:

$$PDF_e(z) = \lambda e^{-\lambda z}, \tag{2.43}$$

where λ is the rate parameter of the distribution. Then, the average value of the image gradient λ^{-1} is computed. The inverse cumulative distribution function of PDF_e is, then, obtained using the following equation:

$$F_e(\Delta_i) = \lambda^{-1} \ln(1 - \Delta_i), \tag{2.44}$$

where $\Delta_i \in [0, 1)$ according to

$$\Delta_i = \frac{i}{L+1}, \tag{2.45}$$

where $i \in \{1, 2, \cdots, L\}$ and L is the number of levels. To select a threshold, we take

$$\tau_i = F_e(\Delta_i) \tag{2.46}$$

for equally spaced values.

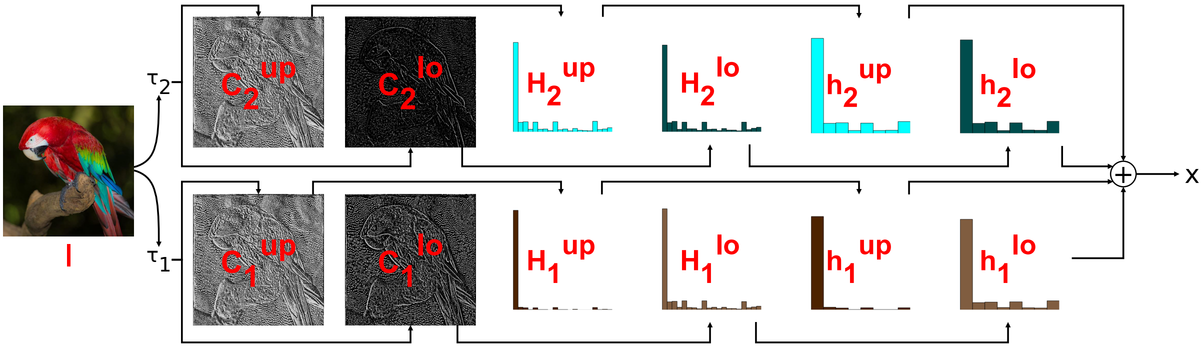


Figure 2.17: Illustration of process of extracting the feature vector x with $L = 2$.

The feature extraction process is illustrated in Fig. 2.17. We decompose the image into LTP channels. These channels are generated by varying the τ values according to Eq. 2.44, 2.45, and 2.46. As described in previous section and depicted in Fig. 2.6, for a single image the LTP operator produces two channels: one related to the upper patterns and one to the lower patterns. Therefore, for L numbers of τ_i, we have $2L$ LTP channels. These channels are illustrated in Figure 2.18. In this figure, we use $L = 4$, which generates eight distinct LTP channels. In the proposed LTP approach, instead of computing the differences between t_c and its neighbors on the grayscale image, we take the maximum difference on the R, G, or B channels.

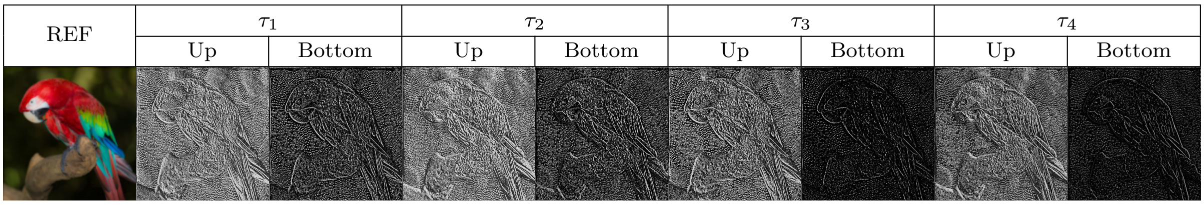


Figure 2.18: Reference image and its upper and lower patterns generated using the Local Ternary Pattern (LTP) operator with four different threshold values.

After the aforementioned steps are completed, we obtain a set of LTP channels with 2×L elements: $\{C_1^{up}, C_1^{lo}, C_2^{up}, C_2^{lo}, \cdots, C_L^{up}, C_L^{lo}\}$. In this set, the subscript index corresponds to the i-th τ value, while the superscript index indicates whether the element is an upper (up) or lower (lo) pattern. For each LTP channel C_i^j, where $j \in \{up, lo\}$, we compute the corresponding LTP histogram H_i^j. These histograms are used to build the feature vector. If we simply concatenate these histograms, we generate a feature vector with a $2^P \times 2 \times L$ dimension. Depending on the L and P parameters, the number of features can be very high, what has a direct impact on the performance of the proposed algorithm.

In order to limit the number of dimensions, the number of bins of the LTP histograms are reduced according to the following formula:

$$k_i^j = \left\lfloor \frac{\max H_i^j - \min H_i^j}{n} \right\rceil, \tag{2.47}$$

where $\lfloor \cdot \rceil$ is the operation of rounding to nearest integer, n defines the number of equal-width bins in the given range, and k_i^j is the reduced number of bins of histogram H_i^j. After this quantization, we acquire a set of quantized histograms $\{h_1^{up}, h_1^{lo}, h_2^{up}, h_2^{lo}, \cdots, h_L^{up}, h_L^{lo}\}$. This new set is used to generate the feature vector associated to the image I. More specifically, the feature vector x is generated by concatenating the quantized histograms h_i^j, i.e.:

$$\breve{x} = h_1^{up} \oplus h_1^{lo} \oplus h_2^{up} \oplus h_2^{lo} \oplus \cdots \oplus h_L^{up} \oplus h_L^{lo}, \tag{2.48}$$

where $\oplus$ is the concatenation operator and x is the feature vector.

2.4.3 Local Variance Patterns (LVP)

The Local Variance Pattern (LVP) is an extension of the LBP operator proposed in this work. This operator was developed specifically for quality assessment tasks. The LVP operator computes the texture local energy using the following formula:

$$LVP^{u}_{R,P}(I_c) = \left\lfloor \frac{P \cdot V_{R,P}(I_c) - \left[LBP_{R,P}(I_c)\right]^2}{P^2} \right\rfloor , \tag{2.49}$$

where:

$$V_{R,P}(I_c) = \sum_{p=0}^{P-1} \left[S(I_p - I_c) \cdot 2^p\right]^2 . \tag{2.50}$$

LVP operator estimates the spread of the texture local energy. By measuring the texture energy, the LVP operator is able to estimate the effect that specific impairments have on the texture. For example, a Gaussian blurring impairment decreases the local texture energy, while a noise impairment increases it.

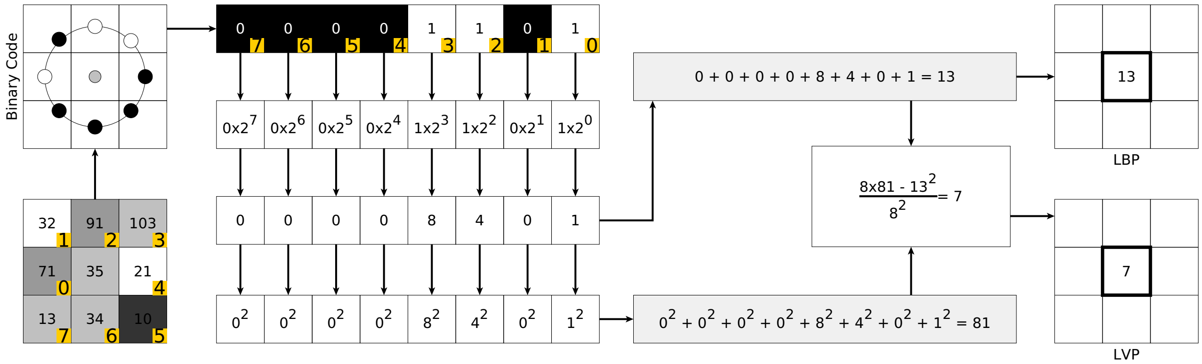


Figure 2.19: Pattern extraction process for a given pixel using LBP and LVP operators with $R = 1$, $P = 8$, $t_c = 35$, and $t_p = \{71, 32, 91, 103, 21, 10, 34, 13\}$.

Fig. 2.19 shows a comparison of the steps used to extract texture information using the LBP and LVP operators, assuming that $R = 1$ and $P = 8$. The numbers in the yellow squares represent the order in which the steps are computed. The LBP operator generates two possible values (see Eq. 2.2), which are represented by the colors white ($S(t) = 1$) and black ($S(t) = 0$). Next, we use Eq. 2.6 to compute the LBP label and Eq. 2.49 to compute the LVP label.

After computing the LBP and LVP labels for all pixels of a given image, we obtain two channels for each image. These channels, C_{LBP} and C_{LVP}, correspond to the LBP and LVP patterns, respectively. Examples of these channels are shown in Fig. 2.20. The first row of this figure shows the unimpaired reference image and three impaired images, degraded with different types of distortions. The second and third rows show the C_{LBP} and C_{LVP} channels for each image, respectively.

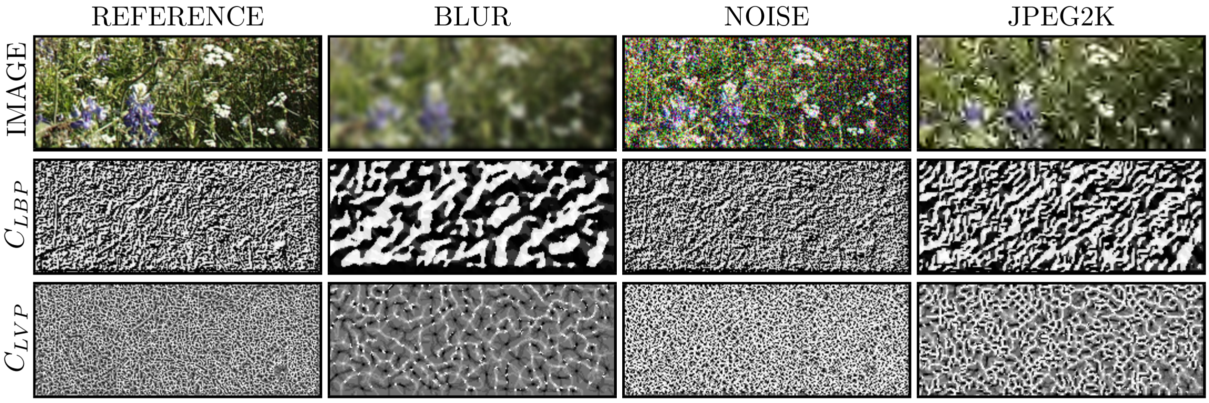


Figure 2.20: Reference image, its impaired versions, and their respective LBP and LVP maps (C_{LBP} and C_{LVP}).

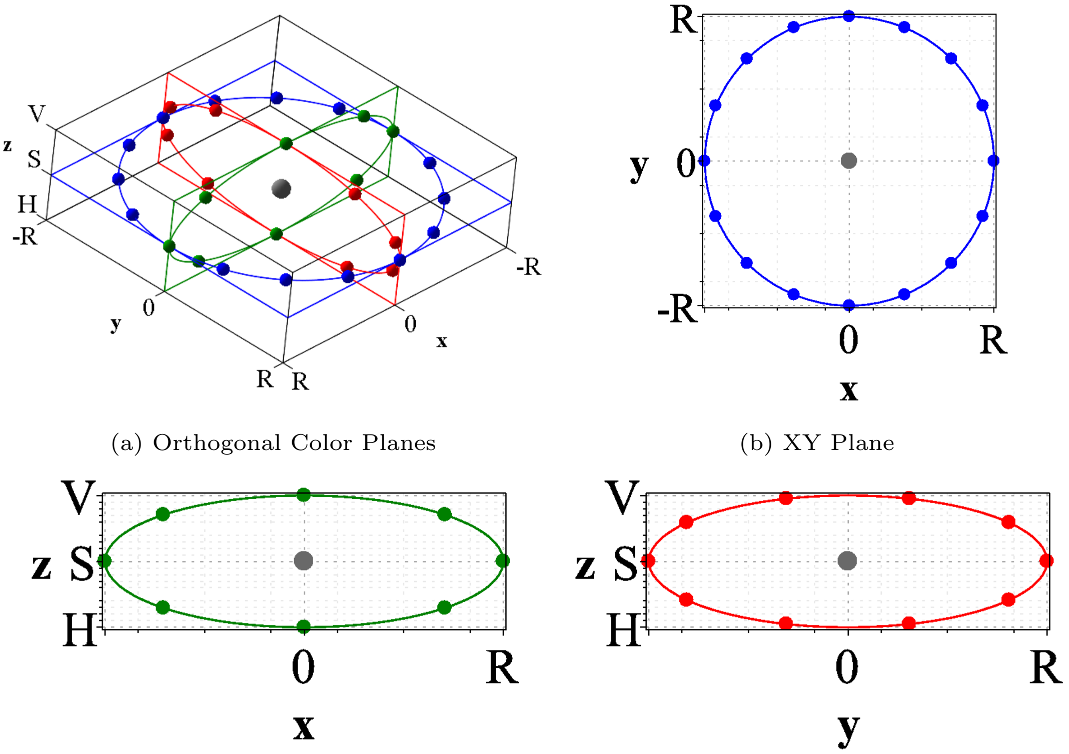
Observing the C_{LBP} and C_{LVP} patterns in Fig. 2.20, we notice that textures are affected differently by the different impairments. Comparing the C_{LBP} channels corresponding to the noisy, blurry, and jpeg2k compressed images (2nd line of Fig. 2.20), we can notice that they are very different among themselves. The C_{LBP} channels corresponding to the blurry and jpeg2k images are also very different from the C_{LBP} channel corresponding to the reference (unimpaired) image. Nevertheless, the C_{LBP} channel corresponding to the noisy and reference images are visually similar. This similarity makes it difficult to discriminate between unimpaired and impaired images, what affects the quality prediction. Nevertheless, the C_{LVP} channels clearly show the differences between impaired and reference images, as can be seen in the 3rd line of Fig. 2.20.

2.4.4 Orthogonal Color Planes Patterns (OCPP)

The Orthogonal Color Planes Pattern (OCPP) descriptor is an extension of the LBP. This operator extends the LBP to make it more sensitive to color and contrast distortions.

Consider a pixel $\tau_c = \mathfrak{I}(x, y, z)$ of a tri-dimensional (XYZ) color image $\mathfrak{I}$. This image can be decomposed into a set of individual XY planes stacked along the Z axis, a set of YZ planes stacked along the X axis, or a set of XZ planes stacked along the Y axis. In this work, we concatenate the LBP descriptors corresponding to the XY, XZ, and YZ planes to build an orthogonal color planes pattern (OCPP) texture descriptor.

As can be noticed from the aforementioned formulation, the LBP operator corresponding to the XY, XZ, and YZ planes can be computed independently to generate the thee LBP maps: LBP$_{XY}$, LBP$_{XZ}$, and LBP$_{YZ}$. But, since the spatial dimensions of the XY, XZ, and YZ planes are generally different, the radius (R_X, R_Y, and R_Z) and the number



(a) Orthogonal Color Planes

(b) XY Plane

(c) XZ Plane

(d) YZ Plane

Figure 2.21: (a) General view of OCCP, (b) XY ($\mathrm{P}_{XY} = 16$) plane, (c) XZ ($\mathrm{P}_{XZ} = 8$) plane, and (d) YZ ($\mathrm{P}_{YZ} = 10$) plane.

of sampled points (P_{XY}, P_{XZ}, and P_{YZ}) corresponding to each of the LBP maps can vary. Fig. 2.21-(a) illustrates how the points along the tri-dimensional HSV color space are sampled, while Figs. 2.21-(b), (c) and (d) illustrate how each of the XY, XZ, and YZ planes are sampled.

Considering $R_Z = 1$ and $R_X = R_Y = R$, the coordinates of the neighboring points in the XY, XZ, and YZ orthogonal planes are given by:

$$x_{XY} = x + R\cos\left(2\pi\frac{p_{XY}}{P_{XY}}\right) \qquad y_{XY} = y - R\sin\left(2\pi\frac{p_{XY}}{P_{XY}}\right),$$

$$x_{XZ} = x + R\cos\left(2\pi\frac{p_{XZ}}{P_{XZ}}\right) \qquad z_{XZ} = z - \sin\left(2\pi\frac{p_{XZ}}{P_{XZ}}\right),$$

and

$$y_{YZ} = y + R\cos\left(2\pi\frac{p_{YZ}}{P_{YZ}}\right) \qquad z_{YZ} = z - \sin\left(2\pi\frac{p_{YZ}}{P_{YZ}}\right).$$

We compute the LBP for each plane using the following equations:

$$\mathfrak{L}_{XY} = LBP_R^{P_{XY}}(\tau_c) = \sum_{p_{XY}=0}^{P_{XY}-1} S(\tau_c - \tau_{XY})2^{p_{XY}},$$

$$\mathfrak{L}_{XZ} = LBP_R^{P_{XZ}}(\tau_c) = \sum_{p_{XZ}=0}^{P_{XZ}-1} S(\tau_c - \tau_{XZ})2^{p_{XZ}},$$

and

$$\mathfrak{L}_{YZ} = LBP_R^{P_{YZ}}(\tau_c) = \sum_{p_{YZ}=0}^{P_{YZ}-1} S(\tau_c - \tau_{YZ})2^{p_{YZ}}.$$

The OCPP descriptor is built by concatenating these individual LBP descriptors:

$$OCPP_R^P(\tau_c) = \left[\mathfrak{L}_{XY}, \mathfrak{L}_{XZ}, \mathfrak{L}_{YZ}\right]^T. \tag{2.51}$$

2.4.5 Salient Local Binary Patterns (SLBP)

The salient local binary pattern (SLBP) is an extension of the LBP which is designed to be used in image quality assessment methods. The operator incorporates visual salient information, given that recent results show that visual attention models improve the performance of visual quality assessment methods [129, 130].

To estimate the saliency of the different areas of an image I, we use a computational visual attention model. More especifically, to keep the computational complexity low, we chose the Boolean map-based saliency (BMS) model [131]. When compared with other state-of-the-art visual attention models, BMS is noticeable faster, while still providing a good performance.

After computing the LBP operator of all pixels of image I, we obtain a LBP map $\mathcal{L}$, where each $\mathcal{L}[x, y]$ gives the local texture associated to the pixel $I[x, y]$. Similarly, the output of BMS is a saliency map $\mathcal{W}$, where each element $\mathcal{W}[x, y]$ corresponds to the probability that the pixel $I[x, y]$ attracts the attention of a human observer. The first, second, and third columns of Fig. 2.22 depict a set of original images I, their corresponding LBP maps $\mathcal{L}$, and their corresponding saliency maps $\mathcal{W}$, respectively.

We generate the feature vector by computing the histogram of $\mathcal{L}$ weighted by $\mathcal{W}$. The histogram $\mathbf{H} = \{h[0], h[1], \cdots, h[P+1]\}$ is given by the following expression:

$$h[\phi] = \sum_i \sum_j \mathcal{W}[i, j]\Delta(\mathcal{L}[i, j], \phi), \tag{2.52}$$

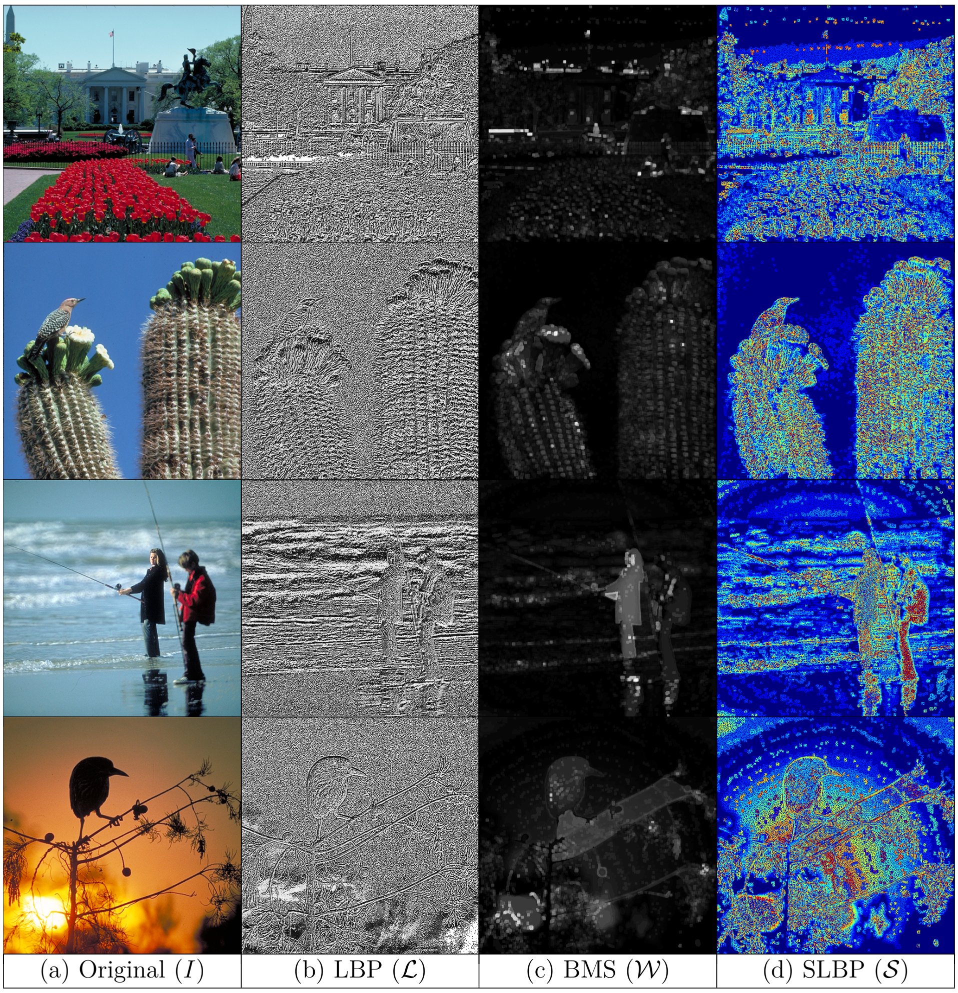


Figure 2.22: Example of original images (a), their LBP (b), BMS (c), and SLBP (d) maps.

where

$$\Delta(v, u) = \begin{cases} 1 & v = u, \\ 0 & \text{otherwise.} \end{cases} \tag{2.53}$$

The number of bins of this histogram is similar to the number of distinct LBP patterns of $\mathcal{L}$. So, we can remap each $\mathcal{L}[i, j]$ to its weighted form, generating the map $\mathcal{S}$ displayed in Fig. 2.22-(d). This figure depicts a heatmap representing the importance of each local texture. We name this weighted LBP map as the *salient local binary patterns (SLBP)*.

2.4.6 Multiscale Salient Local Binary Patterns (MSLBP)

The multiscale salient local binary patterns (MSLBP) is an extension of SLBP in combination with MLBP. The idea behind MSLBP is to achieve fine information about frame texture by varying the parameters of LBP and combining the multiple generated LBP maps with saliency maps. In other words, we variate the SLBP to obtain multiple maps, as illustrated in Fig. 2.23. For each combination of radius (R) and sampled points (P), we have an associated histogram $H_{R,P}$.

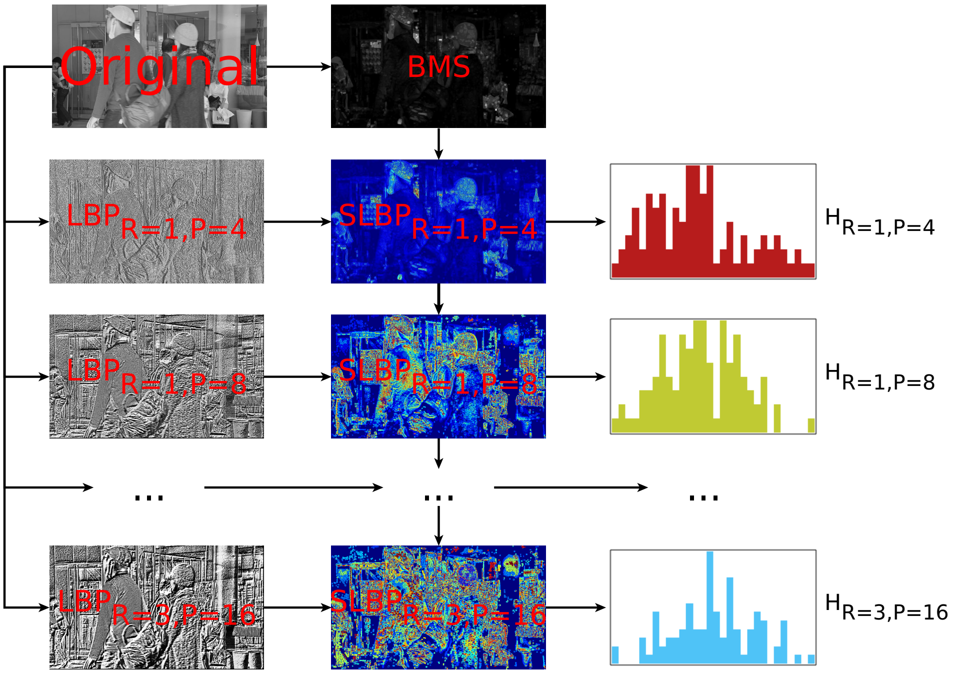


Figure 2.23: Multiple histogram generation from SLBP.

2.5 Summary and Conclusions

The recent advances in texture descriptors, especially the variants of LBP, have driven a great progress in texture analysis for a large variety of computer vision problems. In this chapter, we presented several of the most powerful state-of-the-art LBP variants. Moreover, we also proposed new LBP-based operators in order to measure visual quality information. In the next chapters, we present how to use these operators to predict image and video quality.

Chapter 3

Image Quality Assessment Using Texture Measures

3.1 Overview

In last chapter, we presented a series of texture descriptors. Most of these descriptors were proposed for pattern recognition and computer vision applications. Nevertheless, we also presented a set of proposed descriptors (MLBP, MLTP, LVP, OCPP, and SLBP), which were specially designed for the application of visual quality measurement. In this chapter, our goal is to investigate which operators are more suitable for no-reference (blind) image quality assessment (NR-IQA) method. Moreover, we are concerned on the relation between operator and the performance accuracy of the IQA method.

3.2 Image Quality Assessment Method

The NR-IQA methods presented in this work use a supervised machine learning approach. The supervised machine learning algorithm learns a model from a set of labeled data, which allows predicting results on a different dataset. In regression, a number of predictor variables (features) and a continuous response variable (outcome) are provided to the algorithm, which, then, finds a relationship between these variables, which allows it to predict the outcome. In order to predict a quality score (continuous scale), with small values for poor quality and high scores for good quality, an IQA method uses a regression algorithm. Therefore, the key components of a machine learning IQA method include (1) feature extraction, (2) feature mapping, (3) model learning via regression, and (4) quality prediction using the learned model.

Fig. 3.1 depicts the first part of the set of IQA methods proposed in this work. First, we collect subjective scores corresponding to each image of a training set. This procedure generates a set of labeled images, where each training set entry is composed by a pair of an image marker and its associated MOS (subjective score). In other words, for the *k-th* unlabeled image I_k the algorithm associates a real value v_k, which represents the overall quality of I_k.

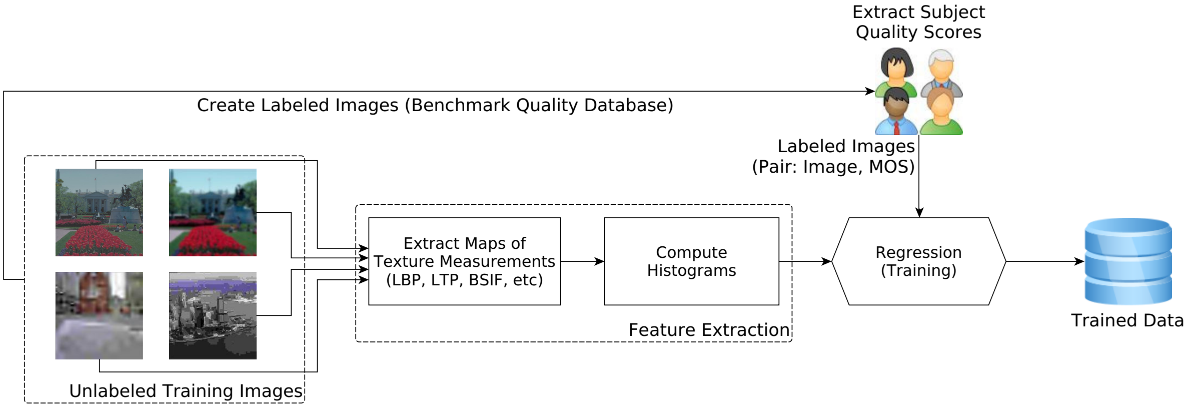

Extract Subject Quality Scores
Create Labeled Images (Benchmark Quality Database)
Labeled Images (Pair: Image, MOS)
Extract Maps of Texture Measurements (LBP, LTP, BSIF, etc)
Compute Histograms
Feature Extraction
Regression (Training)
Trained Data
Unlabeled Training Images

Figure 3.1: Training the quality metric.

After generating the labeled database formed by the set of pairs (I_k, v_k), the features are extracted in order to generate the IQA model. In this work, the features are produced using one of the LBP variants described in Section 2. For each image I_k, we compute the histogram of the given LBP variant H_k and concatenate all histograms to produce the feature vector. Therefore, the training data is composed by the set (H_k, v_k). The model is created using (H_k, v_k), which is formed by a matrix $H \in \mathbb{R}^{K \times Q}$ and a vector $v \in \mathbb{R}^{1 \times K}$. In this case, K is the number of training entries (rows of H) and Q is the number of features (columns and the numbers of bins of H_k).

The prediction model is built using a regression model. This model maps each H_k into a real value v_k that predicts a corresponding quality score. Based on the documentation of Sklearn framework [132], we follow the flowchart depicted in Fig. 3.2 to choose the best estimator to create the regression model. The chosen regression model is the random forest (RF) regressor [133], highlighted in red. RF was chosen because it is an ensemble regressor that combines the predictions of several base estimators that are built with a given learning algorithm, which improves robustness. Moreover, although linear and support vector machine methods are commonly employed in the design of NR-IQA methods [134, 60, 55, 56], our tests showed that RF methods provide improved accuracy performance levels [135].

The quality assessment task is depicted in Fig. 3.3. After generating the prediction model, image quality can be assessed using the trained model. The procedure is the same for every image I_k in the training set. The same feature (LBP histogram) is computed

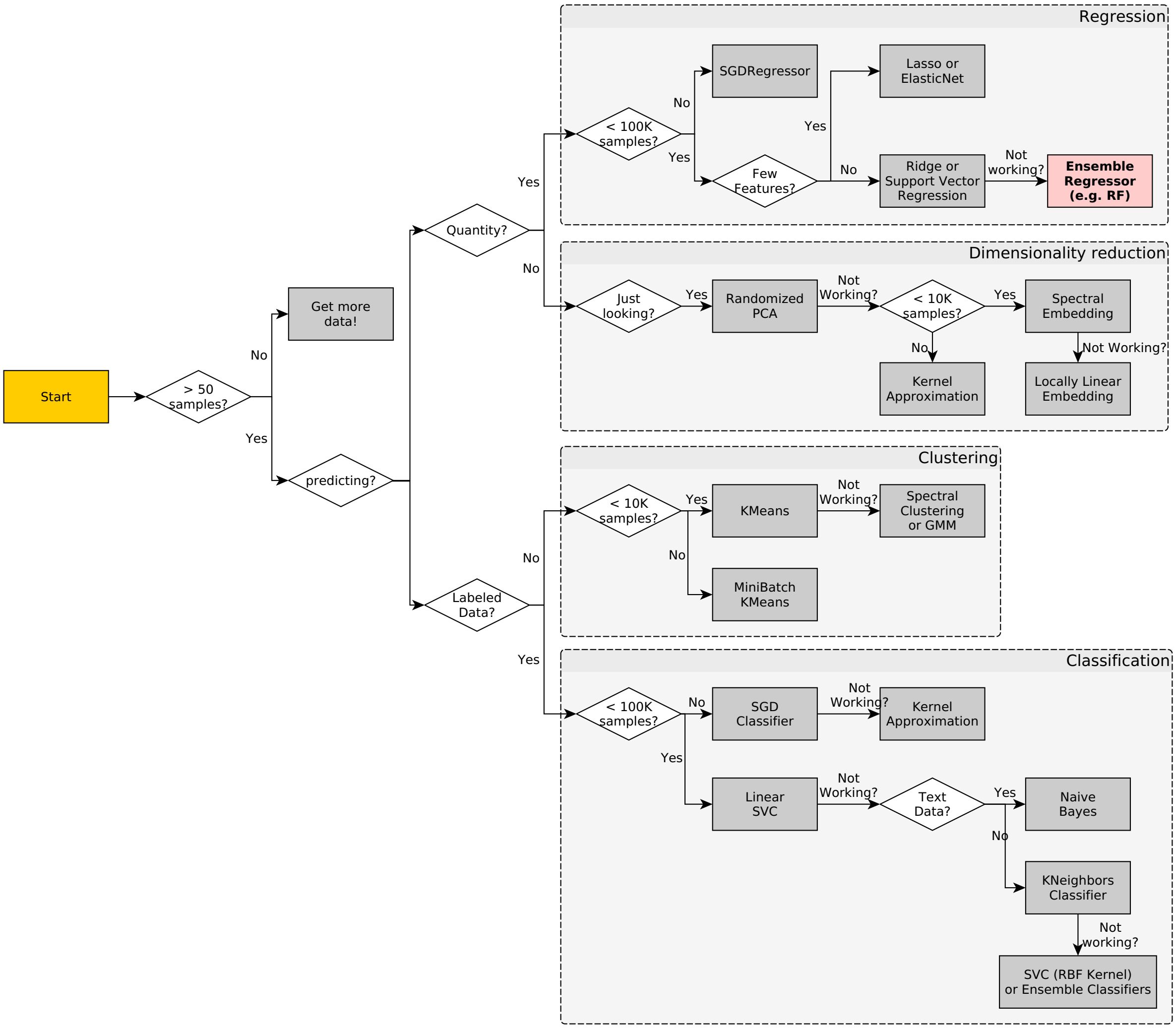


Figure 3.2: Cheat sheet.

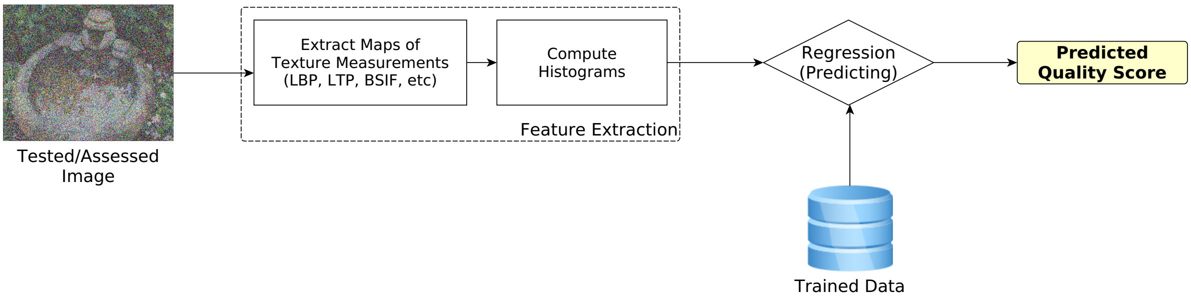


Figure 3.3: Predicting quality scores.

using the test image as input. Then, using this feature, the trained model predicts the quality score.

3.3 Experimental Setup and Protocol

Results were generated using an Intel i7-4790 processor at 3.60GHz. To assess the performance of the proposed NR-IQA method, we compute the Spearman's Rank Ordered Correlation (SROCC) between the mean opinion scores (MOS) and the predicted scores. The proposed method is compared with the fastest state-of-the-art NR-IQA methods, including BRISQUE [134], CORNIA [60], CQA [55], SSEQ [56], and LTP [128]. These methods were chosen because they are all based on machine learning techniques, making the comparison with the proposed method straightforward. Moreover, these methods are publicly available for download.

For all machine learning NR-IQA methods, we use the same procedure for training and testing. In order to avoid overlapping between content detection and quality prediction, we divide the benchmark databases into content-independent training and testing subsets. Specifically, image content in the training subset was not used in the testing subset, and vice-versa. This division is made in a way that 80% of images are used for training and 20% are used for testing. This procedure is repeated 100 times, with randomly selected training and testing subsets. This split is adopted because it is a common approach for several ML-based NR-IQA methods [134, 60, 55]. For the machine learning NR-IQA methods that are based on SVR, we use a LibSVR implementation accessed via Python interface provided by the Sklearn library [132]. The optimal SVR metaparameters (C, γ, ν, etc) are automatically found using exhaustive gridsearch methods provided by Sklearn's API. No optimized search methods are used for the RF version of proposed method.

The tests were performed using three image quality databases, which include subjective scores collected from psychophysical experiments. These databases are:

- LIVE2 [136] database has 982 test images, including 29 originals. This database includes 5 categories of distortions: JPEG, JPEG 2000 (JPEG2k), white noise (WN), Gaussian blur (GB), fast fading (FF).

- CSIQ [137] database has a total fo 866 test images, consisting of 30 originals and 6 different categories of distortions. The distortions include JPEG, JPEG 2000 (JPEG2k), JPEG, white noise (WN), Gaussian blur (GB), fast fading (FF), global contrast decrements (CD), and additive Gaussian pink noise (PN).

- TID2013 [138] database contains 25 reference images with the following distortions: Additive Gaussian noise (AGN), Additive noise in color components (AGC), Spatially correlated noise (SCN), Masked noise (MN), High frequency noise (HFN), Impulse noise (IN), Quantization noise (QN), Gaussian blur (GB), Image denoising (ID), JPEG, JPEG2k, JPEG transmission errors (JPEGTE), JPEG2k transmission errors (JPEG2kTE), Non eccentricity pattern noise (NEPN), Local block-wise distortions (LBD), Intensity shift (IS), Contrast change (CC), Change of color saturation (CCS), Multiplicative Gaussian noise (MGN), Comfort noise (CN), Lossy compression (LC), Image color quantization with dither (ICQ), Chromatic aberration (CA), and Sparse sampling and reconstruction (SSR).

Table 3.1: Tested LBP variants.

Abbreviation	Name	Parameters
LBP^{ri}	Basic Local Binary Patterns with rotation invariance	Radius (R) and number of neighbors (P)
LBP^{u}	Uniform Local Binary Patterns	Radius (R) and number of neighbors (P)
LBP^{riu2}	Uniform Local Binary Patterns with rotation invariance	Radius (R) and number of neighbors (P)
BSIF	Binarized Statistical Image Features	Window size and number of bits
LPQ	Local Phase Quantization)	Local frequency estimation
CLBP	Complete Local Binary Patterns	CLBP-S, CLBP-C, and CLBP-M
LCP	Local Configuration Patterns	Radius (R) and number of neighbors (P)
LTP	Local Ternary Patterns	Threshold (τ), Radius (R) and number of neighbors (P)
RLBP	Rotated Local Binary Patterns	Radius (R) and number of neighbors (P)
TPLBP	Three-Patch Local Binary Patterns	Patch size (w), Radius (R), and angle between neighboring patches
FPLBP	Four-Patch Local Binary Patterns	Patch size (w), Radius of first ring (R1), Radius of second ring (R2), and angle between neighboring patches
LVP	Local Variance Patterns	Radius (R) and number of neighbors (P)
OCLBP	Opposite Color Local Binary Patterns	Radius (R) and number of neighbors (P)
OCPP	Orthogonal Color Planes Patterns	Radius (R) and number of neighbors (P)
SLBP	Salient Local Binary Patterns	Radius (R) and number of neighbors (P)
MLBP	Multiscale Local Binary Patterns	Automatically found
MLTP	Multiscale Local Ternary Patterns	Automatically found
MSLBP	Multiscale Salient Local Binary Patterns	Automatically found

In order to test the LBP and its variants, we vary some parameters of each algorithm. Specifically, we vary the parameters of LBP, BSIF, CLBP, and LPQ. For the other tested variants, we choose the parameters R=1 and P=8. Table 3.1 depicts the parameters used by the tested algorithms.

3.4 Experimental Results

To investigate the suitability of the basic LBP operator, we variate the parameters R and P using the Rotation Invariant LBP (LBP^{ri}), the Uniform LBP (LBP^{u}), and the Uniform LBP with Rotation Invariance (LBP^{riu2}), described in Section 2.2. Fig. 3.4 depicts the distribution of SROCC over simulations on the general case (i.e., when all distortions are considered). Table 3.2 shows the average SROCC correlation values for 100 simulations, following the aforementioned protocol. In this table, STD represents the standard deviation and Δ is the subtraction between the maximum and minimum value in a given row or column.

From Table 3.2, we can notice that the basic LBP operator is suitable for predicting quality. This suitability is indicated by the high correlation indices obtained on LIVE2

Table 3.2: Average SROCC of 100 runs of simulations on tested image databases using basic LBP variations.

DB	DIST	LBP^{ri}								LBP^{u}								LBP^{riu2}								Average	STD	Max	Min	Δ
		R=1		R=2			R=3			R=1		R=2			R=3			R=1		R=2			R=3							
		P=4	P=8	P=4	P=8	P=16	P=4	P=8	P=16	P=4	P=8	P=4	P=8	P=16	P=4	P=8	P=16	P=4	P=8	P=4	P=8	P=16	P=4	P=8	P=16					
LIVE 2	JPEG	0.8959	0.9306	0.8238	0.9058	0.9124	0.7759	0.8683	0.9065	0.8921	0.9275	0.8376	0.9063	0.9176	0.8176	0.8301	0.9069	0.8955	0.9204	0.8343	0.8481	0.8906	0.7716	0.7971	0.8813	0.8706	0.0486	0.9306	0.7716	0.1590
	JPEG2k	0.9062	0.9423	0.8772	0.9161	0.9324	0.7812	0.8999	0.9238	0.9056	0.9353	0.8691	0.9149	0.9277	0.8181	0.8464	0.9023	0.9088	0.9245	0.8742	0.8724	0.8895	0.7857	0.8241	0.8816	0.8858	0.0455	0.9423	0.7812	0.1611
	WN	0.9753	0.9794	0.9521	0.9671	0.9694	0.9309	0.9553	0.9676	0.9743	0.9782	0.9356	0.9661	0.9703	0.9285	0.9465	0.9687	0.9753	0.9771	0.9538	0.9607	0.9661	0.9294	0.9407	0.9642	0.9597	0.0164	0.9794	0.9285	0.0509
	GB	0.9123	0.9621	0.9169	0.9474	0.9551	0.8873	0.9331	0.9479	0.9253	0.9611	0.9168	0.9494	0.9632	0.8771	0.9349	0.9666	0.9137	0.9481	0.9197	0.9144	0.9317	0.8808	0.8946	0.9134	0.9280	0.0265	0.9666	0.8771	0.0895
	FF	0.8341	0.8871	0.7878	0.8687	0.9054	0.6459	0.8027	0.8539	0.8521	0.8755	0.7692	0.8493	0.9026	0.6588	0.6756	0.8714	0.8325	0.8974	0.7821	0.7959	0.8755	0.6488	0.7585	0.8672	0.8124	0.0824	0.9054	0.6459	0.2595
	ALL	0.9015	0.9532	0.8713	0.9288	0.9422	0.8038	0.8988	0.9274	0.9101	0.9417	0.8631	0.9235	0.9427	0.8208	0.8501	0.9282	0.9048	0.9366	0.8704	0.8826	0.9174	0.8034	0.8493	0.9079	0.8950	0.0445	0.9532	0.8034	0.1498
	Average	0.9042	0.9425	0.8715	0.9223	0.9362	0.8042	0.8930	0.9212	0.9099	0.9366	0.8652	0.9183	0.9374	0.8202	0.8473	0.9240	0.9051	0.9340	0.8724	0.8790	0.9118	0.8033	0.8441	0.9026					
	STD	0.0450	0.0318	0.0598	0.0342	0.0246	0.0993	0.0535	0.0391	0.0402	0.0351	0.0594	0.0405	0.0264	0.0906	0.0973	0.0384	0.0456	0.0271	0.0608	0.0563	0.0336	0.0970	0.0661	0.0349					
	MAX	0.9753	0.9794	0.9521	0.9671	0.9694	0.9309	0.9553	0.9676	0.9743	0.9782	0.9356	0.9661	0.9703	0.9285	0.9465	0.9687	0.9753	0.9771	0.9538	0.9607	0.9661	0.9294	0.9407	0.9642					
	MIN	0.8341	0.8871	0.7878	0.8687	0.9054	0.6459	0.8027	0.8539	0.8521	0.8755	0.7692	0.8493	0.9026	0.6588	0.6756	0.8714	0.8325	0.8974	0.7821	0.7959	0.8755	0.6488	0.7585	0.8672					
CSIQ	JPEG	0.8245	0.8861	0.8135	0.8908	0.8705	0.8142	0.8682	0.8806	0.8241	0.8912	0.8513	0.8631	0.8725	0.8446	0.8506	0.8701	0.8176	0.8521	0.8073	0.8518	0.8642	0.8083	0.8323	0.8688	0.8508	0.0271	0.8912	0.8073	0.0839
	JPEG2k	0.7695	0.8532	0.7867	0.8379	0.8414	0.6964	0.8272	0.8339	0.7654	0.8266	0.7658	0.8065	0.8123	0.7025	0.7452	0.7977	0.7699	0.7851	0.7738	0.7571	0.7625	0.6844	0.7063	0.7524	0.7775	0.0479	0.8532	0.6844	0.1688
	WN	0.7079	0.8452	0.6404	0.7926	0.8229	0.5241	0.7984	0.8905	0.6328	0.9133	0.7658	0.7185	0.7499	0.6801	0.7176	0.6588	0.7149	0.8173	0.6403	0.6615	0.7428	0.6793	0.7031	0.6704	0.7287	0.0902	0.9133	0.5241	0.3892
	GB	0.8592	0.9078	0.8378	0.8891	0.9125	0.7889	0.8808	0.9141	0.8669	0.8856	0.8273	0.8738	0.8972	0.7946	0.8455	0.8873	0.8547	0.8923	0.8335	0.8718	0.8778	0.7969	0.8457	0.8828	0.8635	0.0361	0.9141	0.7889	0.1252
	PN	0.5786	0.8827	0.5289	0.8333	0.8768	0.6654	0.7331	0.8541	0.7821	0.8511	0.6184	0.7446	0.7648	0.5857	0.6698	0.6801	0.5735	0.8258	0.5323	0.7571	0.7191	0.5238	0.6301	0.6729	0.7035	0.1159	0.8827	0.5238	0.3589
	CD	0.3066	0.5901	0.3159	0.4791	0.4968	0.2615	0.3857	0.4577	0.3884	0.4714	0.4561	0.2929	0.3536	0.4051	0.3607	0.3052	0.2661	0.3788	0.2967	0.3245	0.3145	0.2731	0.3976	0.2093	0.3661	0.0909	0.5901	0.2093	0.3808
	ALL	0.6735	0.8278	0.6471	0.7946	0.8019	0.6274	0.7561	0.7961	0.6854	0.8028	0.6635	0.7341	0.7457	0.6365	0.7059	0.7086	0.6638	0.7718	0.6421	0.7091	0.7181	0.6211	0.6796	0.6861	0.7124	0.0624	0.8278	0.6211	0.2067
	Average	0.6743	0.8276	0.6529	0.7882	0.8033	0.6254	0.7499	0.8039	0.7064	0.8060	0.7069	0.7191	0.7423	0.6642	0.6993	0.7011	0.6658	0.7605	0.6466	0.7047	0.7141	0.6267	0.6850	0.6775					
	STD	0.1877	0.1082	0.1857	0.1419	0.1400	0.1878	0.1695	0.1575	0.1611	0.1525	0.1384	0.1978	0.1814	0.1445	0.1645	0.1965	0.2001	0.1731	0.1882	0.1833	0.1881	0.1843	0.1494	0.2250					
	MAX	0.8592	0.9078	0.8378	0.8908	0.9125	0.8142	0.8808	0.9141	0.8669	0.9133	0.8513	0.8738	0.8972	0.8446	0.8506	0.8873	0.8547	0.8923	0.8335	0.8718	0.8778	0.8083	0.8457	0.8828					
	MIN	0.3066	0.5901	0.3159	0.4791	0.4968	0.2615	0.3857	0.4577	0.3884	0.4714	0.4561	0.2929	0.3536	0.4051	0.3607	0.3052	0.2661	0.3788	0.2967	0.3245	0.3145	0.2731	0.3976	0.2093					
TID 2013	AGC	0.4781	0.6135	0.2353	0.1084	0.4713	0.3703	0.2131	0.3554	0.1954	0.3496	0.1742	0.1309	0.2519	0.2509	0.2154	0.2912	0.4607	0.3273	0.1975	0.1665	0.1469	0.3681	0.2061	0.2746	0.2855	0.1273	0.6135	0.1084	0.5051
	AGN	0.7861	0.7757	0.4346	0.5881	0.6799	0.5642	0.4426	0.6969	0.6201	0.6138	0.3626	0.2873	0.6726	0.5726	0.2207	0.4581	0.7619	0.5353	0.4434	0.4146	0.5673	0.5342	0.4753	0.5957	0.5460	0.1474	0.7861	0.2207	0.5654
	CA	0.2186	0.2052	0.3674	0.2211	0.2453	0.2967	0.2693	0.2061	0.2035	0.2186	0.2797	0.2216	0.2475	0.3032	0.2962	0.2771	0.2065	0.2407	0.4155	0.3651	0.2828	0.2505	0.3781	0.2939	0.2713	0.0604	0.4155	0.2035	0.2120
	CC	0.1287	0.1007	0.1178	0.1181	0.0869	0.0971	0.1476	0.1696	0.1284	0.1742	0.1131	0.0623	0.0957	0.1238	0.0607	0.0749	0.1551	0.1438	0.1161	0.0996	0.0773	0.0938	0.1098	0.1073	0.1126	0.0306	0.1742	0.0607	0.1135
	CCS	0.1891	0.1241	0.1666	0.1255	0.2309	0.1898	0.2131	0.1473	0.1751	0.1319	0.1938	0.2195	0.2903	0.1754	0.1881	0.2311	0.1699	0.1786	0.1684	0.1587	0.1599	0.1671	0.2374	0.1852	0.1840	0.0386	0.2903	0.1241	0.1662
	CN	0.3052	0.1979	0.1655	0.1425	0.3253	0.1959	0.1181	0.1384	0.1834	0.1851	0.1491	0.1384	0.1742	0.1465	0.1257	0.1842	0.3645	0.1473	0.1748	0.1467	0.1365	0.1701	0.2301	0.2325	0.1866	0.0639	0.3645	0.1181	0.2464
	GB	0.8216	0.8384	0.8041	0.8006	0.8122	0.7781	0.8208	0.8261	0.8139	0.8341	0.8027	0.8253	0.8038	0.7391	0.8152	0.8075	0.8073	0.8199	0.8023	0.8095	0.8253	0.7766	0.7969	0.8276	0.8087	0.0213	0.8384	0.7391	0.0993
	HFN	0.7934	0.8126	0.6968	0.7648	0.8365	0.7793	0.6121	0.8473	0.7901	0.7541	0.6431	0.6719	0.8648	0.7248	0.5231	0.7821	0.7891	0.6511	0.7048	0.6717	0.6701	0.7604	0.6415	0.7375	0.7301	0.0833	0.8648	0.5231	0.3417
	ICQ	0.7741	0.7715	0.7638	0.8246	0.7973	0.6748	0.8088	0.8196	0.7642	0.7633	0.7498	0.7904	0.8183	0.7099	0.7703	0.8173	0.7634	0.7908	0.7554	0.7911	0.8011	0.6383	0.7542	0.7818	0.7706	0.0446	0.8246	0.6383	0.1863
	ID	0.3503	0.8107	0.6211	0.7631	0.7238	0.6084	0.6892	0.6938	0.2738	0.5346	0.5523	0.7415	0.7919	0.5349	0.5742	0.7081	0.3534	0.4192	0.6384	0.6038	0.5019	0.5901	0.4749	0.4761	0.5846	0.1448	0.8107	0.2738	0.5369
	IN	0.1384	0.3423	0.1394	0.5396	0.5431	0.1327	0.3873	0.5954	0.1169	0.0932	0.1551	0.4252	0.4021	0.1269	0.2188	0.3059	0.1665	0.1384	0.1323	0.5894	0.4722	0.1202	0.2169	0.4401	0.2891	0.1738	0.5954	0.0932	0.5022
	IS	0.1378	0.0631	0.1201	0.0977	0.0692	0.1183	0.0795	0.0894	0.1068	0.0598	0.0995	0.0743	0.0659	0.1075	0.0742	0.1054	0.1652	0.0936	0.1322	0.1328	0.0982	0.1025	0.0866	0.1271	0.1003	0.0268	0.1652	0.0598	0.1054
	JPEG	0.7241	0.8392	0.6678	0.8016	0.7973	0.6265	0.7814	0.7861	0.6912	0.8035	0.6523	0.7615	0.7657	0.6311	0.6751	0.7448	0.6888	0.7519	0.6762	0.6631	0.6831	0.6431	0.6367	0.6941	0.7161	0.0642	0.8392	0.6265	0.2127
	JPEGTE	0.1273	0.2942	0.1361	0.3361	0.2784	0.1353	0.3007	0.2869	0.1434	0.1988	0.1261	0.3026	0.3599	0.1452	0.1092	0.2523	0.1707	0.1534	0.1351	0.2103	0.2803	0.1591	0.1453	0.1888	0.2073	0.0783	0.3599	0.1092	0.2507
	JPEG2k	0.7949	0.8669	0.6876	0.8057	0.8384	0.7751	0.8153	0.8373	0.8103	0.8057	0.8151	0.8511	0.8323	0.7634	0.8126	0.7996	0.7888	0.8411	0.8311	0.8218	0.8107	0.7673	0.7515	0.7673	0.8038	0.0384	0.8669	0.6876	0.1793
	JPEG2kTE	0.3888	0.5015	0.8326	0.6049	0.5934	0.5526	0.7203	0.7073	0.4142	0.4981	0.6149	0.7099	0.7131	0.5823	0.5888	0.7007	0.3765	0.4057	0.6853	0.6238	0.5121	0.5581	0.6584	0.5642	0.5878	0.1186	0.8326	0.3765	0.4561
	LBD	0.1634	0.1739	0.1462	0.1657	0.1175	0.1184	0.1442	0.1894	0.1502	0.1605	0.1569	0.1331	0.1332	0.1566	0.1411	0.1323	0.1753	0.1343	0.1263	0.1562	0.1392	0.1335	0.1288	0.1556	0.1472	0.0186	0.1894	0.1175	0.0719
	LC	0.4419	0.5581	0.2869	0.2731	0.4507	0.2769	0.2996	0.4807	0.1542	0.1515	0.1865	0.2107	0.1553	0.1901	0.1476	0.1769	0.3596	0.2734	0.3473	0.1519	0.1284	0.3092	0.2984	0.1146	0.2676	0.1227	0.5581	0.1146	0.4435
	MGN	0.6977	0.6947	0.5239	0.4977	0.7766	0.5871	0.3519	0.7002	0.5971	0.4191	0.2848	0.2084	0.4796	0.4731	0.1605	0.4014	0.7214	0.4916	0.5139	0.4658	0.4893	0.5911	0.4483	0.4081	0.4993	0.1570	0.7766	0.1605	0.6161
	MN	0.2677	0.4295	0.1952	0.3469	0.1832	0.1448	0.1501	0.1615	0.1667	0.3236	0.1531	0.1286	0.1398	0.1449	0.3087	0.1288	0.2438	0.2652	0.1631	0.1573	0.1319	0.1611	0.2252	0.1771	0.2041	0.0807	0.4295	0.1286	0.3009
	NEPN	0.1413	0.2054	0.2107	0.2358	0.3383	0.1611	0.2721	0.3708	0.1329	0.1273	0.1795	0.2862	0.2706	0.1391	0.2917	0.2094	0.1254	0.1416	0.1667	0.2787	0.3373	0.1533	0.2252	0.1996	0.2167	0.0740	0.3708	0.1254	0.2454
	QN	0.7733	0.8584	0.7871	0.8073	0.8353	0.7306	0.7965	0.8115	0.8254	0.8631	0.8069	0.8226	0.8757	0.7957	0.8019	0.8384	0.7769	0.8042	0.7772	0.7764	0.8053	0.7431	0.7828	0.8242	0.8050	0.0350	0.8757	0.7306	0.1451
	SCN	0.6399	0.6603	0.7103	0.6426	0.5357	0.5411	0.6003	0.6807	0.6111	0.6303	0.5681	0.6257	0.7496	0.5811	0.4169	0.6084	0.6673	0.6803	0.6965	0.5442	0.4238	0.5538	0.5457	0.6853	0.6083	0.0828	0.7496	0.4169	0.3327
	SSR	0.8246	0.8846	0.8151	0.8507	0.9142	0.7042	0.7911	0.8873	0.7126	0.7776	0.7596	0.7603	0.8188	0.7503	0.7431	0.7884	0.8215	0.6981	0.8203	0.7142	0.7338	0.6931	0.6653	0.7431	0.7780	0.0667	0.9142	0.6653	0.2489
	ALL	0.4593	0.5859	0.4618	0.5174	0.5356	0.4171	0.4781	0.5198	0.4253	0.4661	0.4031	0.4751	0.5281	0.3848	0.4059	0.4682	0.4413	0.4431	0.4621	0.4688	0.4604	0.4169	0.4224	0.4728	0.4633	0.0478	0.5859	0.3848	0.2011
	Average	0.4626	0.5283	0.4438	0.4792	0.5207	0.4231	0.4521	0.5202	0.4082	0.4375	0.3993	0.4346	0.4920	0.4101	0.3874	0.4517	0.4608	0.4228	0.4433	0.4393	0.4270	0.4182	0.4217	0.4430					
	STD	0.2720	0.2877	0.2703	0.2782	0.2736	0.2515	0.2648	0.2796	0.2789	0.2799	0.2645	0.2863	0.2933	0.2568	0.2627	0.2765	0.2654	0.2631	0.2730	0.2559	0.2599	0.2503	0.2369	0.2592					
	MAX	0.8246	0.8846	0.8326	0.8507	0.9142	0.7793	0.8208	0.8873	0.8254	0.8631	0.8151	0.8511	0.8757	0.7957	0.8152	0.8384	0.8215	0.8411	0.8311	0.8218	0.8253	0.7766	0.7969	0.8276					
	MIN	0.1273	0.0631	0.1178	0.0977	0.0692	0.0971	0.0795	0.0894	0.1068	0.0598	0.0995	0.0623	0.0659	0.1075	0.0607	0.0749	0.1254	0.0936	0.1161	0.0996	0.0773	0.0938	0.0866	0.1073					

database. On this database, the average SROCC vary from 0.8034 to 0.9532 in general case, from 0.6459 to 0.9054 for FF distortion, from 0.8771 to 0.9666 for GB, from 0.9285 to 0.9794 for WN, from 0.7812 to 0.9423 for JPEG2k, and from 0.7716 to 0.9306 for JPEG. These values suggest that basic LBP variations are well appropriate to model quality of images under WN and GB distortions. Independently of basic LBP parameters, the prediction performance of WN and GB are less affected in comparison with other distortions, as indicated by the variance and Δ values.

Although the basic LBP works well for WN and GB distortions independently of its parameters, the performance for other distortions varies according to rotation invariance and other parameters. This variation is also observed on CSIQ and TID2013 databases. For example, on CSIQ database, the SROCC values varies from 0.8073 to 0.8912 in the best case (JPEG) and from 0.2093 to 0.5901 in the worst case (CD). These values indicate that prediction performance is related to basic LBP parameters. Actually, this is the premise used by Freitas *et al.* [125], who assumes that different parameters of LBP can be used to achieve quality information. The aggregation of features obtained via different LBP parameters is used to achieve a more robust quality assessment model. This is the idea begin the MLBP.

Once it has been demonstrated that basic LBP variants present a suitable operator to describe image quality, we check the performance of other LBP extensions described in Chapter 2. To perform the tests, we variate the parameters of BSIF, LPQ, and CLBP operators. For the remaining extensions (i.e., LCP, LTP, RLBP, TPLBP, FPLBP, LVP, OCLBP, OCPP, SLBP, MLBP, MLTP, and MSLBP), we do not variate the parameters. Fig. 3.5 depicts the distribution of SROCC over simulations on the general case using the tested LBP variants.

To investigate the suitability of the basic BSIF operator, we performed the simulations by changing the patch size and the number of selected binarized feature (see Section 2.3.3). The results of the performed simulations on the LIVE2 and CSIQ databases are depicted in Table 3.3 and Table 3.4, respectively.

Based on results of Table 3.3, we can notice that the BSIF is a valuable descriptor to describe quality. In LIVE2 database, the BSIF performs well for almost all configurations. However, the results are better for smaller patch sizes. In these cases, the average SROCC values are higher with low variance. From Table 3.4, the performance of BSIF decreases for CSIQ database. When compared with LIVE2 database, the average SROCC values are lower and the variance is higher. The values in both Table 3.3 and Table 3.4 indicate that exists a relation between patch size and number of bits. More specially, the larger the patch size, the higher the number of bits required to a good quality prediction. For example, using a 3×3 patch, the better performance is obtained using 8 bits and the worse

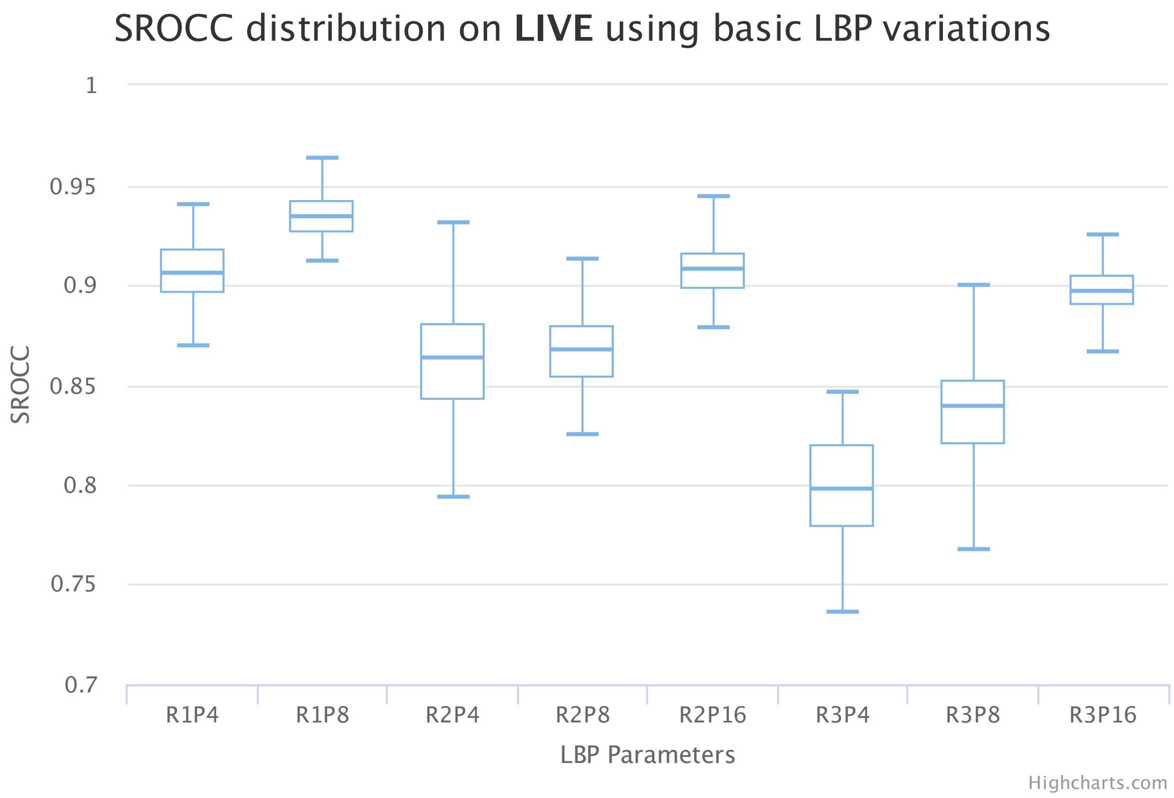
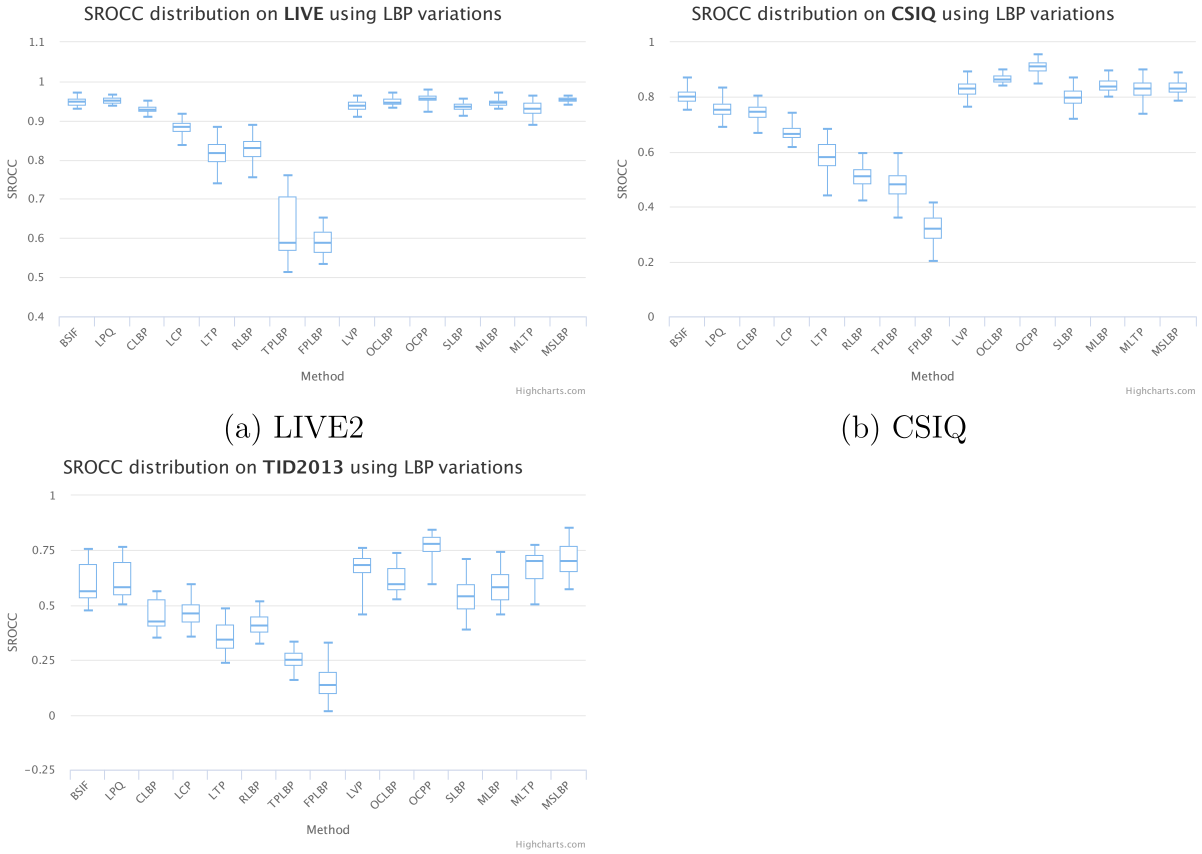


Figure 3.4: SROCC distribution on LIVE2 using basic LBP

performance is obtained using only 5 bits in both LIVE2 and CSIQ databases.

Table 3.5 shows the results of simulations using seven different LPQ configurations. These configurations depend of the LPQ parameters. The main parameters of LPQ descriptor include the size of the local window and the method used for local frequency estimation. The size of local window was fixed on 3×3. The tests were performed by varying the method used for local frequency estimation. The configurations are as follows:

- C1: Short-term Fourier transform (STFT) with uniform window (corresponds to basic version of LPQ);
- C2: STFT with Gaussian window;
- C3: Gaussian derivative quadrature filter pair;
- C4: STFT with uniform window + STFT with Gaussian window (i.e., concatenation of the feature vectors produced by C1 and C2);
- C5: STFT with uniform window + STFT with Gaussian derivative quadrature filter pair (i.e., concatenation of the feature vectors produced by C1 and C3);
- C6: STFT with Gaussian window + Gaussian derivative quadrature filter pair (i.e., concatenation of the feature vectors produced by C2 and C3);



(a) LIVE2

(b) CSIQ

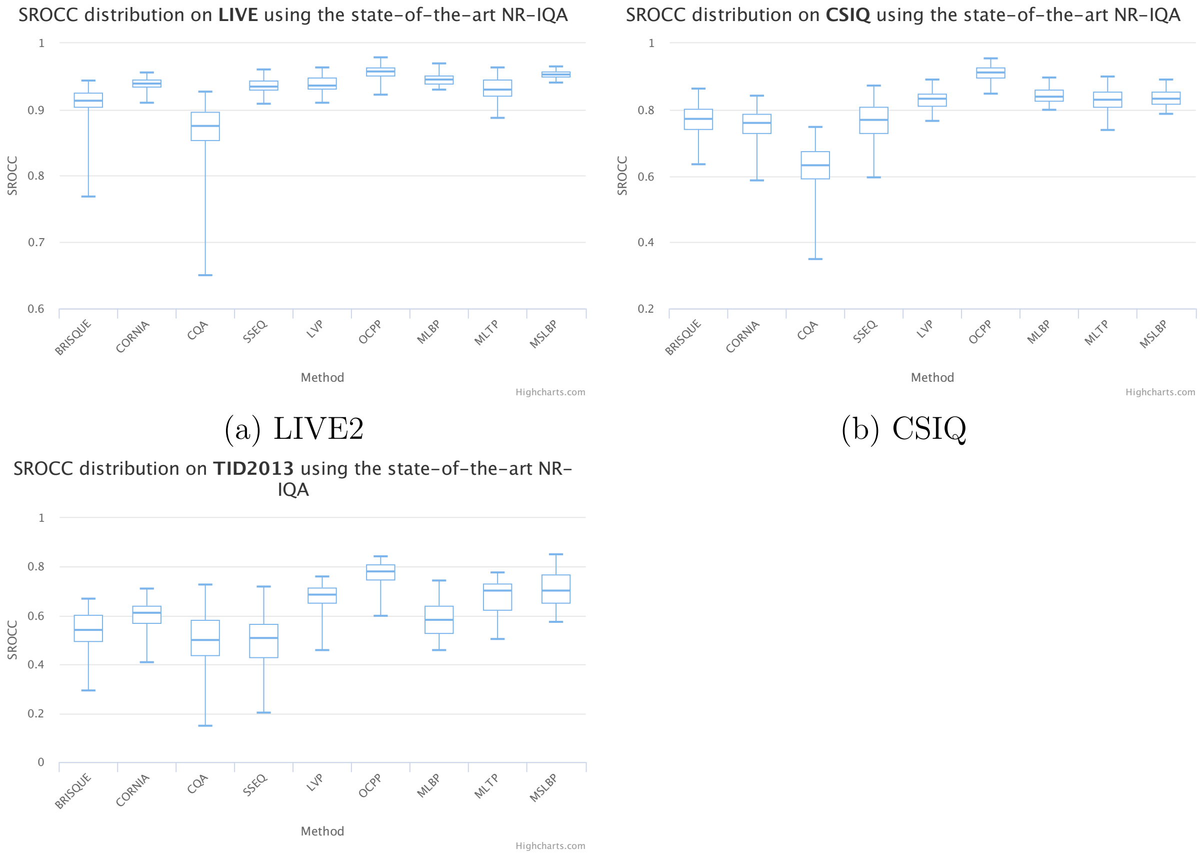
(c) TID2013

Figure 3.5: Distribution of average SROCC after 1000 simulations using different LBP variantions.

- C7: Concatenation of the feature vectors produced by C1, C2, and C3.

From Table 3.5, we can observe that LPQ perform with a high correlation on LIVE2 database. For this database, the mean SROCC values are above 0.9 for all distortions, independently of the configuration. The low variance and high average of SROCC values on LIVE2 indicate that LPQ is a suitable descriptor for measuring quality of JPEG, JPEG2k, WN, GB, and FF distortions. However, the performance of the prediction decreases on the CSIQ and TID2013 databases. This decrease of performance is due to the presence of the CD distortion on CSIQ and other distortions of TID2013 database.

Table 3.6 shows the average SROCC of simulations using CLBP as texture descriptor. For this descriptors, we tested the influence of each feature set (see CLBP$_S$, CLBP$_M$, and CLBP$_C$ in Fig. 2.10) and its combinations on the image quality prediction. From Table 3.6, we can notice that the feature sets, CLBP$_M$ and CLBP$_C$, are individually unsatisfactory for measuring image quality. This is due to the low SROCC scores obtained via these feature sets on the three tested databases. On the other hand, CLBP$_S$ is the dominant feature set for quality description, since it present the higher SROCC values in almost all cases.



(a) LIVE2

(b) CSIQ

(c) TID2013

Figure 3.6: Distribution of average SROCC after 1000 simulations using different state-of-the-art methods.

Interestingly, the combination of CLBP feature sets produces a better performance, as indicate by the overall and the average performances of CLBP$_{SM}$ (CLBP$_S$ + CLBP$_M$) and CLBP$_{SMC}$ (CLBP$_S$ + CLBP$_M$ + CLBP$_C$). From Table 3.6, we can observe that the mean SROCC value of overall case increases from 0.91 (CLBP$_S$) to 0.93 (CLBP$_{MC}$ and CLBP$_{SMC}$) on LIVE2 database. The combination of feature sets also improves the average SROCC values of TID2013 database, increasing from 0.35 (CLBP$_S$) to 0.44 (CLBP$_{MC}$ and CLBP$_{SMC}$). The average values on CSIQ database shows that the best performance is obtained using CLBP$_{MC}$. Based on these SROCC values, we can conclude that CLBP$_{MC}$ is the best combination of features to assess image quality, since the incorporation of CLBP$_C$ does not improve or even deteriorate the general prediction performance.

Table 3.7 depicts the mean SROCC values of simulations using other LBP variants. From this table, we can noticed that almost all variants present an acceptable performance on LIVE2 database. The exceptions are TPLBP and FPLBP that presented mean SROCC below 0.65, which is poorer than other methods. Based on the average values of mean SROCC on LIVE2, the methods LTP, RLBP, LCP, LVP, MLTP, SLBP, OCLBP, MLBP, MSLBP, and OCPP are in ascending order of performance. On CSIQ and TID2013 databases, the methods perform similarly, but RLBP performs worse than LTP on CSIQ.

It is noticeable that multiscale approaches (MLBP, MLTP, and MSLBP) present the best results. These results on the three tested databases are in agreement with the assumptions of Freitas *et al.* [125], who demonstrated that combining multiple LBP operator parameters increases the prediction performance. However, we also can observe that OCPP operator presents the best performance when compared with any other operator, even the multiscale approaches.

Although the OCPP operator performs similarly to MSLBP operator on LIVE2 database, this performance is not maintained in the other databases. While MSLBP presents average SROCC values of 0.8147 on CSIQ database, the OCPP presents a SROCC average value of 0.9140 on the same database. Similarly, on TID2013, the MSLBP and OCPP average values are 0.5919 and 0.7035, respectively.

By observing the SROCC scores per distortion in the CSIQ database, we can notice that the superiority of OCPP is due to the prediction performance of CD distortion. While the quality prediction of contrast-distorted images have mean SROCC of 0.5299 using MSLBP, the mean SROCC of this same images are 0.7753. Similarly, on TID2013 database, the OCPP presents superior performance for the in several cases, mainly on the cases of color and contrast-related distortions (AGC, AGN, CA, CC, CCS, etc).

Finally, Fig. 3.6 depicts the box plots of different no-reference methods. Moreover, Table 3.8 depict the results of six IQA methods, including two established full-reference metrics (PSNR and SSIM) and four state-of-the-art no-reference metrics (BRISQUE, CORNIA, CQA, and SSEQ). From this table, we can notice that CORNIA and SSEQ present the best performance on LIVE2 database, even when compared with full-reference approaches, such as PSNR and SSIM. On the LIVE2 database, the average SROCC values of CORNIA and SSEQ is 0.92, a score similar to some LBP-based operators, such as CLBP$_{SM}$ and BSIF. However, several LBP-based operators present a notable performance, being superior to the state-of-the-art methods, as we can observe from results of LPQ, MLBP, MSLBP, and OCPP with average SROCC above 0.94 on LIVE2.

By comparing Table 3.8 with Tables 3.7, 3.6, and 3.5, we can notice that LBP-based NR-IQA approaches present better performance also on CSIQ and TID2013 databases. On CSIQ, we can observe that, on average, the best state-of-the-art NR-IQA method is BRISQUE, followed by SSEQ and CORNIA. The average SROCC scores are 0.7406, 0.6979, and 0.6886 for BRISQUE, SSEQ, and CORNIA, respectively. However, when we notice from Tables 3.7, 3.4 and 3.5 that LPQ, BSIF, LVP, OCLBP, OCPP, SLBP, MLBP, MLTP, and MSLBP operators present better results on CSIQ when compared with the state-of-the-art methods. Similarly, on TID2013 database, the best state-of-the-art method is CORNIA, which presents an average SROCC of 0.5361. This value is outperformed by several LBP-based operators, such as LVP (0.5428), OCLBP (0.5902), OCPP (0.7035), MLBP (0.5284), MLTP (0.5652), MSLBP (0.5919), and LPQ (0.5518).

3.5 Summary and Conclusions

In this chapter, we compared three basic LBP (LBP^{ri}, LBP^{u}, and LBP^{riu2}) with eight different parameter combinations each. This comparison was used to verify whether LBP can be adopted as feature for measuring image quality. From the results, LBP can be used as feature to measure, although the performance for each distortion varies greatly on its parameters. This variation is surpasses by the MLBP operator, which combines the histograms of LBPs at multiple parameters in order to make it more precise on quality prediction. In addition to LBP and MLBP, we investigated other 14 operators that extends the basic LBP. Using the proposed framework with these operators, we perceived that the BSIF, LPQ, LVP, and CLBP present a good mean correlation on LIVE2 database, but the performance decrease significantly on CSIQ and TID2013 due to color and contrast distortions. We also observed that multiscale approaches increase substantially the overall quality prediction performance. Among the multiscale approaches, the MSLBP operator, which incorporates visual saliency in multiple scales, is the best one. MSLBP competes with the OCPP operator on LIVE2 database. However, OCPP presents the best performance on all tested databases, whether compared to other LBP-based operators or the state-of-the-art methods.

Table 3.3: Average SROCC of 100 runs of simulations on LIVE2 database using BSIF variations.

SIZE	BITS	DISTORTION						Average	STD	MAX	MIN
		JPEG	JPEG2k	WN	GB	FF	ALL				
3x3	5 bits	0.8991	0.9444	0.9594	0.9542	0.8644	0.9302	0.9253	0.0368	0.9594	0.8644
	6 bits	0.9159	0.9403	0.9664	0.9621	0.8917	0.9397	0.9360	0.0283	0.9664	0.8917
	7 bits	0.9184	0.9448	0.9688	0.9672	0.9105	0.9471	0.9428	0.0242	0.9688	0.9105
	8 bits	0.9441	0.9399	0.9719	0.9687	0.9064	0.9528	0.9473	0.0238	0.9719	0.9064
5x5	5 bits	0.8391	0.8727	0.9109	0.9297	0.7539	0.8654	0.8620	0.0621	0.9297	0.7539
	6 bits	0.8854	0.9035	0.9419	0.9501	0.8001	0.9052	0.8977	0.0538	0.9501	0.8001
	7 bits	0.8991	0.9185	0.9498	0.9661	0.8174	0.9184	0.9116	0.0521	0.9661	0.8174
	8 bits	0.8912	0.9186	0.9621	0.9654	0.8749	0.9278	0.9233	0.0366	0.9654	0.8749
	9 bits	0.9285	0.9462	0.9646	0.9654	0.9113	0.9488	0.9441	0.0211	0.9654	0.9113
	10 bits	0.9368	0.9423	0.9636	0.9689	0.9257	0.9521	0.9482	0.0164	0.9689	0.9257
	11 bits	0.9295	0.9418	0.9635	0.9693	0.9135	0.9487	0.9444	0.0209	0.9693	0.9135
	12 bits	0.9252	0.9387	0.9573	0.9662	0.9065	0.9434	0.9396	0.0216	0.9662	0.9065
7x7	5 bits	0.8157	0.7896	0.8896	0.7899	0.6391	0.7861	0.7850	0.0815	0.8896	0.6391
	6 bits	0.8292	0.8614	0.9093	0.9174	0.6251	0.8438	0.8310	0.1068	0.9174	0.6251
	7 bits	0.8811	0.8808	0.9233	0.9369	0.6652	0.8691	0.8594	0.0988	0.9369	0.6652
	8 bits	0.8881	0.8896	0.9289	0.9572	0.7507	0.8958	0.8851	0.0712	0.9572	0.7507
	9 bits	0.8894	0.9067	0.9345	0.9499	0.7471	0.8978	0.8876	0.0725	0.9499	0.7471
	10 bits	0.9018	0.9151	0.9463	0.9661	0.7681	0.9108	0.9014	0.0696	0.9661	0.7681
	11 bits	0.9034	0.9096	0.9522	0.9615	0.7796	0.9144	0.9035	0.0652	0.9615	0.7796
	12 bits	0.9141	0.9179	0.9553	0.9638	0.8348	0.9241	0.9183	0.0457	0.9638	0.8348
9x9	5 bits	0.7457	0.6802	0.8488	0.6631	0.4938	0.6877	0.6866	0.1162	0.8488	0.4938
	6 bits	0.8125	0.8202	0.8721	0.8439	0.6036	0.8035	0.7926	0.0959	0.8721	0.6036
	7 bits	0.8318	0.8364	0.8946	0.9071	0.6073	0.8229	0.8167	0.1084	0.9071	0.6073
	8 bits	0.8612	0.8568	0.9097	0.9311	0.6336	0.8464	0.8398	0.1063	0.9311	0.6336
	9 bits	0.8419	0.8919	0.8988	0.9209	0.6446	0.8511	0.8415	0.1010	0.9209	0.6446
	10 bits	0.8631	0.8943	0.9168	0.9373	0.6572	0.8654	0.8557	0.1014	0.9373	0.6572
	11 bits	0.8761	0.9038	0.9262	0.9299	0.6566	0.8692	0.8603	0.1029	0.9299	0.6566
	12 bits	0.8882	0.9036	0.9277	0.9369	0.6737	0.8793	0.8682	0.0978	0.9369	0.6737
11x11	5 bits	0.6866	0.5856	0.7771	0.5996	0.4302	0.6172	0.6161	0.1155	0.7771	0.4302
	6 bits	0.7319	0.7085	0.8009	0.7696	0.5244	0.7206	0.7093	0.0968	0.8009	0.5244
	7 bits	0.7631	0.7736	0.8561	0.8497	0.5387	0.7616	0.7571	0.1152	0.8561	0.5387
	8 bits	0.8362	0.7963	0.8761	0.8637	0.5801	0.7983	0.7918	0.1087	0.8761	0.5801
	9 bits	0.8014	0.8723	0.8801	0.8921	0.6315	0.8211	0.8164	0.0973	0.8921	0.6315
	10 bits	0.8511	0.8811	0.8921	0.9058	0.6221	0.8391	0.8319	0.1058	0.9058	0.6221
	11 bits	0.8516	0.8652	0.8951	0.9011	0.6173	0.8343	0.8274	0.1060	0.9011	0.6173
	12 bits	0.8641	0.8843	0.9077	0.9193	0.6284	0.8534	0.8429	0.1080	0.9193	0.6284
13x13	5 bits	0.6561	0.4312	0.7253	0.5049	0.3396	0.5402	0.5329	0.1419	0.7253	0.3396
	6 bits	0.6681	0.6397	0.7518	0.7141	0.5153	0.6571	0.6577	0.0809	0.7518	0.5153
	7 bits	0.7066	0.6553	0.8076	0.7515	0.5169	0.6941	0.6887	0.0991	0.8076	0.5169
	8 bits	0.7927	0.6897	0.8127	0.8047	0.5471	0.7347	0.7303	0.1014	0.8127	0.5471
	9 bits	0.7946	0.7879	0.8379	0.8479	0.5494	0.7712	0.7648	0.1096	0.8479	0.5494
	10 bits	0.8088	0.8578	0.8563	0.8939	0.5885	0.8081	0.8022	0.1097	0.8939	0.5885
	11 bits	0.8081	0.8331	0.8721	0.8666	0.5925	0.7993	0.7953	0.1037	0.8721	0.5925
	12 bits	0.8435	0.8461	0.8776	0.8712	0.6031	0.8178	0.8099	0.1035	0.8776	0.6031
15x15	5 bits	0.4443	0.3415	0.5823	0.4565	0.2401	0.4288	0.4156	0.1155	0.5823	0.2401
	6 bits	0.6107	0.6164	0.6665	0.6611	0.4639	0.6096	0.6047	0.0735	0.6665	0.4639
	7 bits	0.6624	0.6111	0.7849	0.6947	0.4953	0.6465	0.6492	0.0957	0.7849	0.4953
	8 bits	0.6773	0.6456	0.7709	0.7264	0.4962	0.6641	0.6634	0.0938	0.7709	0.4962
	9 bits	0.7161	0.7597	0.8041	0.7745	0.5258	0.7172	0.7162	0.0993	0.8041	0.5258
	10 bits	0.7754	0.7078	0.7842	0.7973	0.4647	0.7103	0.7066	0.1245	0.7973	0.4647
	11 bits	0.7363	0.7602	0.8259	0.7905	0.4975	0.7291	0.7233	0.1163	0.8259	0.4975
	12 bits	0.7893	0.8118	0.8451	0.8421	0.5505	0.7741	0.7688	0.1106	0.8451	0.5505
17x17	5 bits	0.5746	0.3345	0.4937	0.4158	0.2169	0.4205	0.4093	0.1243	0.5746	0.2169
	6 bits	0.6169	0.5015	0.6572	0.5849	0.3836	0.5491	0.5489	0.0972	0.6572	0.3836
	7 bits	0.5986	0.4765	0.6779	0.5907	0.4261	0.5526	0.5537	0.0906	0.6779	0.4261
	8 bits	0.6782	0.5758	0.7171	0.6325	0.4642	0.6091	0.6128	0.0883	0.7171	0.4642
	9 bits	0.6752	0.6297	0.6922	0.6635	0.4538	0.6287	0.6239	0.0870	0.6922	0.4538
	10 bits	0.7221	0.6581	0.7475	0.7121	0.4322	0.6618	0.6556	0.1149	0.7475	0.4322
	11 bits	0.6976	0.7171	0.7705	0.7171	0.4611	0.6804	0.6740	0.1086	0.7705	0.4611
	12 bits	0.7139	0.7305	0.7928	0.7731	0.5183	0.7058	0.7057	0.0979	0.7928	0.5183
Average		0.7968	0.7833	0.8526	0.8306	0.6195	0.7834				
STD		0.1091	0.1558	0.1045	0.1427	0.1722	0.1342				
Max		0.9441	0.9462	0.9719	0.9693	0.9257	0.9528				
Min		0.4443	0.3345	0.4937	0.4158	0.2169	0.4205				
Max-Min		0.4998	0.6117	0.4782	0.5535	0.7088	0.5323				

Table 3.4: Average SROCC of 100 runs of simulations on CSIQ database using BSIF variations.

SIZE	BITS	DISTORTION							Average	STD	MAX	MIN
		JPEG	JPEG2k	WN	GB	PN	CD	ALL				
3x3	5 bits	0.8492	0.8237	0.7476	0.8565	0.6381	0.2055	0.6986	0.6885	0.2279	0.8565	0.2055
	6 bits	0.8618	0.8183	0.6945	0.8856	0.6405	0.1057	0.7261	0.6761	0.2671	0.8856	0.1057
	7 bits	0.8643	0.8511	0.7498	0.8993	0.7988	0.3181	0.7686	0.7500	0.1978	0.8993	0.3181
	8 bits	0.8819	0.8514	0.8034	0.9055	0.8415	0.3439	0.8015	0.7756	0.1941	0.9055	0.3439
5x5	5 bits	0.8099	0.7551	0.6021	0.8269	0.6162	0.0849	0.6461	0.6202	0.2531	0.8269	0.0849
	6 bits	0.8526	0.7957	0.6542	0.8952	0.6564	0.0722	0.6861	0.6589	0.2760	0.8952	0.0722
	7 bits	0.8552	0.8335	0.6496	0.9092	0.7489	0.0688	0.7185	0.6834	0.2850	0.9092	0.0688
	8 bits	0.8501	0.8508	0.6974	0.9166	0.8162	0.1841	0.7472	0.7232	0.2484	0.9166	0.1841
	9 bits	0.8658	0.8416	0.7984	0.9134	0.7846	0.1022	0.7501	0.7223	0.2788	0.9134	0.1022
	10 bits	0.8511	0.8507	0.8031	0.9204	0.8197	0.1213	0.7574	0.7320	0.2739	0.9204	0.1213
	11 bits	0.8619	0.8402	0.8099	0.9085	0.8356	0.0751	0.7572	0.7269	0.2911	0.9085	0.0751
	12 bits	0.8671	0.8468	0.7804	0.9116	0.7952	0.1544	0.7611	0.7309	0.2596	0.9116	0.1544
7x7	5 bits	0.8048	0.6903	0.5314	0.7519	0.5811	0.1255	0.5999	0.5836	0.2242	0.8048	0.1255
	6 bits	0.8376	0.7998	0.6021	0.8892	0.7704	0.0601	0.6981	0.6653	0.2829	0.8892	0.0601
	7 bits	0.8453	0.7918	0.6379	0.8871	0.7761	0.0785	0.7051	0.6745	0.2757	0.8871	0.0785
	8 bits	0.8427	0.7911	0.6419	0.8913	0.7569	0.0508	0.7061	0.6687	0.2848	0.8913	0.0508
	9 bits	0.8421	0.8291	0.6832	0.9036	0.8275	0.0752	0.7401	0.7001	0.2848	0.9036	0.0752
	10 bits	0.8478	0.8428	0.7221	0.9067	0.8333	0.0841	0.7451	0.7117	0.2839	0.9067	0.0841
	11 bits	0.8557	0.8378	0.7107	0.9078	0.8302	0.0742	0.7384	0.7078	0.2876	0.9078	0.0742
	12 bits	0.8553	0.8451	0.7231	0.9151	0.8357	0.0617	0.7411	0.7110	0.2940	0.9151	0.0617
9x9	5 bits	0.7272	0.5976	0.1411	0.7603	0.3754	0.1409	0.4773	0.4600	0.2554	0.7603	0.1409
	6 bits	0.8337	0.7481	0.5016	0.8542	0.6653	0.1262	0.6475	0.6252	0.2508	0.8542	0.1262
	7 bits	0.8192	0.7587	0.5737	0.8416	0.7074	0.1009	0.6651	0.6381	0.2540	0.8416	0.1009
	8 bits	0.8216	0.7699	0.6371	0.8757	0.7819	0.0787	0.6908	0.6651	0.2704	0.8757	0.0787
	9 bits	0.8138	0.7808	0.5979	0.8727	0.7751	0.0712	0.6891	0.6572	0.2733	0.8727	0.0712
	10 bits	0.8399	0.8161	0.6482	0.8963	0.7921	0.0733	0.7088	0.6821	0.2809	0.8963	0.0733
	11 bits	0.8405	0.8083	0.6994	0.8872	0.7951	0.0605	0.7101	0.6859	0.2838	0.8872	0.0605
	12 bits	0.8465	0.8166	0.7399	0.8989	0.8423	0.0538	0.7197	0.7025	0.2928	0.8989	0.0538
11x11	5 bits	0.6606	0.5265	0.1288	0.7187	0.3999	0.1751	0.4154	0.4321	0.2246	0.7187	0.1288
	6 bits	0.7581	0.6958	0.2838	0.8275	0.6201	0.1386	0.5683	0.5560	0.2539	0.8275	0.1386
	7 bits	0.8041	0.7061	0.3808	0.8278	0.6177	0.1652	0.5982	0.5857	0.2383	0.8278	0.1652
	8 bits	0.7798	0.7121	0.4981	0.8358	0.6617	0.1008	0.6267	0.6021	0.2465	0.8358	0.1008
	9 bits	0.7867	0.7494	0.5154	0.8614	0.7151	0.0717	0.6489	0.6212	0.2659	0.8614	0.0717
	10 bits	0.8287	0.7769	0.5805	0.8851	0.7261	0.0946	0.6748	0.6524	0.2655	0.8851	0.0946
	11 bits	0.8318	0.7701	0.6053	0.8724	0.7759	0.0814	0.6807	0.6597	0.2703	0.8724	0.0814
	12 bits	0.8408	0.7918	0.6431	0.8742	0.7938	0.1032	0.6887	0.6765	0.2655	0.8742	0.1032
13x13	5 bits	0.5675	0.4666	0.1739	0.6523	0.2555	0.1092	0.3509	0.3680	0.2033	0.6523	0.1092
	6 bits	0.6497	0.6107	0.1067	0.7119	0.4467	0.1131	0.4558	0.4421	0.2467	0.7119	0.1067
	7 bits	0.7081	0.6655	0.1468	0.7993	0.5247	0.1585	0.5091	0.5017	0.2589	0.7993	0.1468
	8 bits	0.7681	0.7185	0.3119	0.8438	0.6205	0.1478	0.5984	0.5727	0.2531	0.8438	0.1478
	9 bits	0.7708	0.7208	0.3795	0.8271	0.6565	0.0831	0.6124	0.5786	0.2618	0.8271	0.0831
	10 bits	0.8022	0.7801	0.3915	0.8765	0.6563	0.0605	0.6246	0.5988	0.2852	0.8765	0.0605
	11 bits	0.8181	0.7272	0.5228	0.8596	0.6896	0.1177	0.6437	0.6255	0.2500	0.8596	0.1177
	12 bits	0.8246	0.7508	0.5181	0.8654	0.7181	0.1665	0.6539	0.6425	0.2389	0.8654	0.1665
15x15	5 bits	0.5115	0.4193	0.2073	0.5791	0.1534	0.0662	0.2665	0.3148	0.1920	0.5791	0.0662
	6 bits	0.6201	0.5856	0.1494	0.7127	0.4924	0.1421	0.4533	0.4508	0.2248	0.7127	0.1421
	7 bits	0.6823	0.6263	0.1271	0.7573	0.5156	0.1422	0.4898	0.4772	0.2514	0.7573	0.1271
	8 bits	0.7126	0.6816	0.1271	0.8028	0.6139	0.1488	0.5381	0.5178	0.2722	0.8028	0.1271
	9 bits	0.6951	0.6905	0.1461	0.8088	0.5686	0.0904	0.5371	0.5052	0.2795	0.8088	0.0904
	10 bits	0.7891	0.7014	0.2863	0.8366	0.6284	0.1181	0.5865	0.5638	0.2660	0.8366	0.1181
	11 bits	0.7971	0.7567	0.3305	0.8611	0.6041	0.1221	0.6052	0.5824	0.2679	0.8611	0.1221
	12 bits	0.7979	0.7332	0.3431	0.8495	0.6344	0.1971	0.5956	0.5930	0.2409	0.8495	0.1971
17x17	5 bits	0.5552	0.4113	0.2081	0.5275	0.1476	0.1005	0.2703	0.3172	0.1824	0.5552	0.1005
	6 bits	0.6235	0.5535	0.2499	0.6895	0.3437	0.0926	0.3943	0.4210	0.2139	0.6895	0.0926
	7 bits	0.5818	0.5511	0.1445	0.6742	0.4045	0.1432	0.4027	0.4146	0.2085	0.6742	0.1432
	8 bits	0.6884	0.6583	0.1571	0.7726	0.5681	0.1959	0.5055	0.5066	0.2413	0.7726	0.1571
	9 bits	0.7028	0.6662	0.1281	0.8012	0.5457	0.1981	0.5089	0.5073	0.2552	0.8012	0.1281
	10 bits	0.7455	0.6623	0.1596	0.8244	0.6236	0.1037	0.5581	0.5253	0.2825	0.8244	0.1037
	11 bits	0.7605	0.7166	0.1371	0.8301	0.6219	0.1753	0.5403	0.5403	0.2787	0.8301	0.1371
	12 bits	0.7641	0.6926	0.1142	0.8213	0.5789	0.1712	0.5408	0.5262	0.2799	0.8213	0.1142
Average		0.7795	0.7293	0.4631	0.8329	0.6477	0.1208	0.6157				
STD		0.0894	0.1082	0.2420	0.0842	0.1662	0.0576	0.1258				
Max		0.8819	0.8514	0.8099	0.9204	0.8423	0.3439	0.8015				
Min		0.5115	0.4113	0.1067	0.5275	0.1476	0.0508	0.2665				
Max-Min		0.3704	0.4401	0.7032	0.3929	0.6947	0.2931	0.5350				

Table 3.5: Average SROCC of 100 runs of simulations on tested databases using LPQ variations.

DB	DIST	C1	C2	C3	C4	C5	C6	C7	Average	STD	Max	Min	Δ
LIVE 2	JPEG	0.9477	0.9521	0.9471	0.9542	0.9496	0.9512	0.9527	0.9507	0.0026	0.9542	0.9471	0.0071
	JPEG2k	0.9321	0.9533	0.9347	0.9526	0.9363	0.9501	0.9491	0.9440	0.0092	0.9533	0.9321	0.0212
	WN	0.9644	0.9711	0.9671	0.9666	0.9671	0.9682	0.9681	0.9675	0.0020	0.9711	0.9644	0.0067
	GB	0.9481	0.9501	0.9567	0.9542	0.9561	0.9576	0.9567	0.9542	0.0037	0.9576	0.9481	0.0095
	FF	0.9261	0.8792	0.8939	0.9153	0.9111	0.8861	0.9079	0.9028	0.0169	0.9261	0.8792	0.0469
	ALL	0.9498	0.9463	0.9471	0.9539	0.9511	0.9512	0.9541	0.9505	0.0030	0.9541	0.9463	0.0078
	Average	0.9447	0.9420	0.9411	0.9495	0.9452	0.9441	0.9481					
	STD	0.0137	0.0320	0.0255	0.0175	0.0195	0.0292	0.0207					
	MAX	0.9644	0.9711	0.9671	0.9666	0.9671	0.9682	0.9681					
	MIN	0.9261	0.8792	0.8939	0.9153	0.9111	0.8861	0.9079					
CSIQ	JPEG	0.8896	0.9028	0.8873	0.8808	0.8766	0.8802	0.8779	0.8850	0.0092	0.9028	0.8766	0.0262
	JPEG2k	0.8326	0.8506	0.8279	0.7993	0.7939	0.8164	0.8028	0.8176	0.0206	0.8506	0.7939	0.0567
	WN	0.8307	0.7908	0.8121	0.6776	0.7149	0.7051	0.6992	0.7472	0.0620	0.8307	0.6776	0.1531
	GB	0.8946	0.9068	0.8973	0.8749	0.8745	0.8922	0.8818	0.8889	0.0122	0.9068	0.8745	0.0323
	PN	0.6696	0.7514	0.6581	0.7621	0.7071	0.7719	0.7691	0.7270	0.0484	0.7719	0.6581	0.1138
	CD	0.3371	0.2951	0.3127	0.4405	0.3499	0.4481	0.4326	0.3737	0.0649	0.4481	0.2951	0.1530
	ALL	0.7517	0.7974	0.7482	0.7601	0.7545	0.7713	0.7651	0.7640	0.0167	0.7974	0.7482	0.0492
	Average	0.7437	0.7564	0.7348	0.7422	0.7245	0.7550	0.7469					
	STD	0.1960	0.2116	0.2035	0.1506	0.1789	0.1503	0.1530					
	MAX	0.8946	0.9068	0.8973	0.8808	0.8766	0.8922	0.8818					
	MIN	0.3371	0.2951	0.3127	0.4405	0.3499	0.4481	0.4326					
TID 2013	AGC	0.5007	0.3534	0.5556	0.4446	0.4741	0.4501	0.4611	0.4628	0.0614	0.5556	0.3534	0.2022
	AGN	0.6031	0.6619	0.6515	0.7669	0.7284	0.7726	0.7773	0.7088	0.0697	0.7773	0.6031	0.1742
	CA	0.3051	0.2556	0.3006	0.2788	0.1663	0.2675	0.2492	0.2604	0.0466	0.3051	0.1663	0.1388
	CC	0.1376	0.1551	0.1342	0.1115	0.0926	0.1165	0.1253	0.1247	0.0202	0.1551	0.0926	0.0625
	CCS	0.1351	0.1303	0.1223	0.1615	0.1853	0.1865	0.1831	0.1577	0.0282	0.1865	0.1223	0.0642
	CN	0.1449	0.3619	0.1111	0.3881	0.3381	0.4338	0.3996	0.3111	0.1289	0.4338	0.1111	0.3227
	GB	0.8235	0.7824	0.8542	0.7549	0.6406	0.7462	0.7432	0.7636	0.0684	0.8542	0.6406	0.2136
	HFN	0.7732	0.8107	0.8271	0.8178	0.7696	0.8067	0.8097	0.8021	0.0220	0.8271	0.7696	0.0575
	ICQ	0.7946	0.8176	0.8096	0.8331	0.8095	0.8362	0.8253	0.8180	0.0147	0.8362	0.7946	0.0416
	ID	0.7392	0.6126	0.7457	0.7181	0.6796	0.6757	0.7265	0.6996	0.0471	0.7457	0.6126	0.1331
	IN	0.4723	0.5992	0.4826	0.6325	0.6281	0.6222	0.6406	0.5825	0.0730	0.6406	0.4723	0.1683
	IS	0.0761	0.0691	0.0846	0.0928	0.0791	0.0902	0.0854	0.0825	0.0083	0.0928	0.0691	0.0237
	JPEG	0.8165	0.8383	0.8065	0.8238	0.7876	0.8346	0.8138	0.8173	0.0173	0.8383	0.7876	0.0507
	JPEGTE	0.3511	0.3881	0.3842	0.4238	0.4142	0.4284	0.4273	0.4024	0.0290	0.4284	0.3511	0.0773
	JPEG2k	0.8649	0.8526	0.8576	0.6696	0.6523	0.6919	0.6651	0.7506	0.1016	0.8649	0.6523	0.2126
	JPEG2kTE	0.5501	0.6115	0.5226	0.5407	0.5057	0.5665	0.5761	0.5533	0.0353	0.6115	0.5057	0.1058
	LBD	0.1841	0.1103	0.2096	0.2551	0.2823	0.2856	0.3215	0.2355	0.0724	0.3215	0.1103	0.2112
	LC	0.4461	0.2326	0.4992	0.5292	0.5711	0.5442	0.5653	0.4840	0.1188	0.5711	0.2326	0.3385
	MGN	0.5153	0.7012	0.5645	0.7634	0.7575	0.7762	0.7713	0.6928	0.1083	0.7762	0.5153	0.2609
	MN	0.3383	0.1901	0.3115	0.2317	0.2435	0.2571	0.2441	0.2595	0.0500	0.3383	0.1901	0.1482
	NEPN	0.1502	0.1463	0.1348	0.2668	0.3266	0.2661	0.2833	0.2249	0.0786	0.3266	0.1348	0.1918
	QN	0.8565	0.8296	0.8603	0.8634	0.8881	0.8673	0.8734	0.8627	0.0179	0.8881	0.8296	0.0585
	SCN	0.6592	0.4769	0.7315	0.7126	0.7607	0.7742	0.7538	0.6956	0.1037	0.7742	0.4769	0.2973
	SSR	0.7696	0.7751	0.7969	0.8853	0.8469	0.8919	0.8938	0.8371	0.0558	0.8938	0.7696	0.1242
	ALL	0.5419	0.5474	0.5673	0.5696	0.5595	0.5812	0.5804	0.5639	0.0152	0.5812	0.5419	0.0393
	Average	0.5020	0.4924	0.5170	0.5414	0.5275	0.5508	0.5518					
	STD	0.2645	0.2740	0.2752	0.2577	0.2517	0.2553	0.2544					
	MAX	0.8649	0.8526	0.8603	0.8853	0.8881	0.8919	0.8938					
	MIN	0.0761	0.0691	0.0846	0.0928	0.0791	0.0902	0.0854					

Table 3.6: Average SROCC of 100 runs of simulations on tested databases using CLBP variations.

DB	DIST	CLBP						Average	STD	Max	Min	Δ
		C	M	S	MC	SM	SMC					
LIVE 2	JPEG	0.1444	0.7687	0.9042	0.7527	0.9198	0.9258	0.7359	0.2998	0.9258	0.1444	0.7814
	JPEG2k	0.1102	0.5711	0.8839	0.6198	0.9201	0.9201	0.6709	0.3151	0.9201	0.1102	0.8099
	WN	0.1019	0.8979	0.9733	0.8636	0.9773	0.9713	0.7976	0.3440	0.9773	0.1019	0.8754
	GB	0.0902	0.6143	0.9121	0.5764	0.9397	0.9467	0.6799	0.3333	0.9467	0.0902	0.8565
	FF	0.0876	0.6676	0.8779	0.6281	0.8793	0.8902	0.6718	0.3087	0.8902	0.0876	0.8026
	ALL	0.0878	0.6933	0.9114	0.7076	0.9386	0.9355	0.7124	0.3259	0.9386	0.0878	0.8508
	Average	0.1037	0.7022	0.9105	0.6914	0.9291	0.9316					
	STD	0.0219	0.1174	0.0339	0.1059	0.0322	0.0272					
	MAX	0.1444	0.8979	0.9733	0.8636	0.9773	0.9713					
	MIN	0.0876	0.5711	0.8779	0.5764	0.8793	0.8902					
CSIQ	JPEG	0.1602	0.7726	0.8438	0.7621	0.8891	0.8911	0.7198	0.2797	0.8911	0.1602	0.7309
	JPEG2k	0.0678	0.5228	0.7401	0.3847	0.7902	0.8037	0.5516	0.2894	0.8037	0.0678	0.7359
	WN	0.2646	0.6242	0.8255	0.4952	0.8052	0.7531	0.6280	0.2171	0.8255	0.2646	0.5609
	GB	0.2805	0.7736	0.8637	0.6481	0.8872	0.8887	0.7236	0.2359	0.8887	0.2805	0.6082
	PN	0.1718	0.6456	0.7047	0.4988	0.7198	0.6879	0.5714	0.2116	0.7198	0.1718	0.5480
	CD	0.0981	0.1064	0.2673	0.1369	0.1981	0.1887	0.1659	0.0645	0.2673	0.0981	0.1692
	ALL	0.0494	0.6241	0.7653	0.5291	0.7587	0.7463	0.5788	0.2757	0.7653	0.0494	0.7159
	Average	0.1561	0.5813	0.7158	0.4936	0.7212	0.7085					
	STD	0.0913	0.2274	0.2060	0.1985	0.2390	0.2411					
	MAX	0.2805	0.7736	0.8637	0.7621	0.8891	0.8911					
	MIN	0.0494	0.1064	0.2673	0.1369	0.1981	0.1887					
TID 2013	AGC	0.1251	0.1235	0.2998	0.1431	0.1981	0.3902	0.2133	0.1093	0.3902	0.1235	0.2667
	AGN	0.2001	0.1738	0.4469	0.3923	0.4723	0.5834	0.3781	0.1609	0.5834	0.1738	0.4096
	CA	0.1912	0.2045	0.1934	0.2448	0.2941	0.1636	0.2153	0.0468	0.2941	0.1636	0.1305
	CC	0.1541	0.0953	0.1201	0.0815	0.1081	0.0681	0.1045	0.0305	0.1541	0.0681	0.0860
	CCS	0.1938	0.1339	0.1364	0.1276	0.1319	0.2196	0.1572	0.0393	0.2196	0.1276	0.0920
	CN	0.1671	0.1653	0.1671	0.1263	0.1338	0.0896	0.1415	0.0312	0.1671	0.0896	0.0775
	GB	0.1159	0.1344	0.7789	0.1471	0.7879	0.7148	0.4465	0.3451	0.7879	0.1159	0.6720
	HFN	0.1639	0.2408	0.4899	0.3112	0.6762	0.6851	0.4279	0.2236	0.6851	0.1639	0.5212
	ICQ	0.0982	0.1957	0.6291	0.2757	0.7296	0.7583	0.4478	0.2912	0.7583	0.0982	0.6601
	ID	0.1292	0.1581	0.2426	0.1584	0.6031	0.3423	0.2723	0.1798	0.6031	0.1292	0.4739
	IN	0.1188	0.1192	0.1128	0.1307	0.4315	0.4711	0.2307	0.1714	0.4711	0.1128	0.3583
	IS	0.1565	0.1301	0.0687	0.0835	0.0702	0.0655	0.0958	0.0383	0.1565	0.0655	0.0910
	JPEG	0.1483	0.6049	0.7311	0.5319	0.7873	0.7661	0.5949	0.2402	0.7873	0.1483	0.6390
	JPEGTE	0.1091	0.0911	0.1176	0.1121	0.2907	0.1634	0.1473	0.0742	0.2907	0.0911	0.1996
	JPEG2k	0.2411	0.1417	0.7673	0.1698	0.7819	0.7365	0.4731	0.3184	0.7819	0.1417	0.6402
	JPEG2kTE	0.1673	0.6049	0.3873	0.1265	0.4138	0.4353	0.3559	0.1794	0.6049	0.1265	0.4784
	LBD	0.1269	0.1229	0.1711	0.1611	0.2688	0.1081	0.1598	0.0585	0.2688	0.1081	0.1607
	LC	0.0809	0.1707	0.1707	0.2044	0.2831	0.2819	0.1986	0.0768	0.2831	0.0809	0.2022
	MGN	0.1145	0.2296	0.3784	0.3309	0.5006	0.5834	0.3562	0.1721	0.5834	0.1145	0.4689
	MN	0.2028	0.3076	0.1911	0.3325	0.2323	0.3956	0.2770	0.0812	0.3956	0.1911	0.2045
	NEPN	0.1425	0.1182	0.1454	0.1571	0.1944	0.1663	0.1540	0.0256	0.1944	0.1182	0.0762
	QN	0.4102	0.5626	0.6864	0.5081	0.8851	0.8534	0.6510	0.1914	0.8851	0.4102	0.4749
	SCN	0.1167	0.4926	0.5526	0.4935	0.5346	0.6869	0.4795	0.1915	0.6869	0.1167	0.5702
	SSR	0.1381	0.1523	0.6665	0.1434	0.7996	0.8146	0.4524	0.3411	0.8146	0.1381	0.6765
	ALL	0.0431	0.1262	0.3464	0.0911	0.4608	0.4267	0.2491	0.1835	0.4608	0.0431	0.4177
	Average	0.1542	0.2240	0.3599	0.2234	0.4428	0.4388					
	STD	0.0684	0.1610	0.2377	0.1373	0.2551	0.2640					
	MAX	0.4102	0.6049	0.7789	0.5319	0.8851	0.8534					
	MIN	0.0431	0.0911	0.0687	0.0815	0.0702	0.0655					

Table 3.7: Average SROCC of 100 runs of simulations on tested image databases using other LBP variations.

DB	DIST	LCP	LTP	RLBP	TPLBP	FPLBP	LVP	OCLBP	OCPP	SLBP	MLBP	MLTP	MSLBP
LIVE 2	JPEG	0.8921	0.8278	0.8052	0.7047	0.6626	0.9363	0.9312	0.9678	0.9151	0.9249	0.9395	0.9373
	JPEG2k	0.8913	0.8029	0.8299	0.6491	0.5552	0.9461	0.9411	0.9597	0.9334	0.9342	0.9372	0.9406
	WN	0.9628	0.9358	0.9225	0.6354	0.6774	0.9764	0.9731	0.9861	0.9825	0.9822	0.9646	0.9831
	GB	0.9304	0.8824	0.9111	0.5923	0.5884	0.9531	0.9571	0.9612	0.9432	0.9524	0.9530	0.9619
	FF	0.8034	0.7004	0.7821	0.6724	0.6443	0.8848	0.8936	0.9141	0.9079	0.9487	0.8758	0.9364
	ALL	0.9006	0.8251	0.8487	0.6308	0.6171	0.9376	0.9418	0.9562	0.9405	0.9238	0.9316	0.9528
	Average	0.8968	0.8291	0.8499	0.6475	0.6242	0.9391	0.9397	0.9575	0.9371	0.9444	0.9336	0.9520
	STD	0.0534	0.0794	0.0566	0.0384	0.0465	0.0303	0.0269	0.0238	0.0263	0.0220	0.0307	0.0182
	MAX	0.9628	0.9358	0.9225	0.7047	0.6774	0.9764	0.9731	0.9861	0.9825	0.9822	0.9646	0.9831
	MIN	0.8034	0.7004	0.7821	0.5923	0.5552	0.8848	0.8936	0.9141	0.9079	0.9238	0.8758	0.9364
CSIQ	JPEG	0.8412	0.8011	0.7186	0.7524	0.7179	0.9221	0.8943	0.9596	0.8754	0.8847	0.9292	0.9064
	JPEG2k	0.7746	0.6371	0.6552	0.5699	0.6118	0.8946	0.8865	0.9331	0.7913	0.8095	0.8877	0.8156
	WN	0.8152	0.5057	0.6064	0.1931	0.3599	0.7063	0.8441	0.9186	0.8495	0.9014	0.6454	0.8939
	GB	0.7724	0.7901	0.7939	0.8517	0.6972	0.9137	0.9203	0.9390	0.8539	0.9159	0.9244	0.8816
	PN	0.7049	0.5356	0.2078	0.0815	0.3367	0.7091	0.8361	0.9471	0.7502	0.8872	0.7828	0.8431
	CD	0.1382	0.2246	0.1072	0.3174	0.1025	0.2659	0.4914	0.7753	0.4515	0.5172	0.2082	0.5299
	ALL	0.6672	0.5804	0.5109	0.4815	0.3214	0.8238	0.8421	0.9253	0.7971	0.8399	0.8280	0.8324
	Average	0.6734	0.5821	0.5143	0.4639	0.4496	0.7479	0.8164	0.9140	0.7670	0.8223	0.7437	0.8147
	STD	0.2435	0.1958	0.2607	0.2847	0.2300	0.2312	0.1468	0.0626	0.1457	0.1395	0.2559	0.1300
	MAX	0.8412	0.8011	0.7939	0.8517	0.7179	0.9221	0.9203	0.9596	0.8754	0.9159	0.9292	0.9064
	MIN	0.1382	0.2246	0.1072	0.0815	0.1025	0.2659	0.4914	0.7753	0.4515	0.5172	0.2082	0.5299
TID 2013	AGC	0.3683	0.3654	0.2273	0.1942	0.1207	0.4688	0.5315	0.8308	0.3999	0.5708	0.5963	0.6018
	AGN	0.3903	0.4211	0.5903	0.1731	0.2111	0.6069	0.7253	0.8634	0.6369	0.7884	0.6631	0.7811
	CA	0.2844	0.2267	0.3356	0.2884	0.1604	0.6944	0.4254	0.8821	0.2379	0.3144	0.6749	0.3891
	CC	0.1089	0.1857	0.0816	0.0953	0.1331	0.1756	0.0846	0.4785	0.1261	0.0881	0.1886	0.2161
	CCS	0.1251	0.1503	0.1934	0.2148	0.1296	0.1997	0.5704	0.5577	0.1402	0.1375	0.2384	0.2757
	CN	0.4769	0.2896	0.2682	0.1101	0.1942	0.2101	0.5849	0.5309	0.2725	0.3249	0.3880	0.5229
	GB	0.8455	0.5795	0.8084	0.8072	0.4096	0.8551	0.8607	0.8914	0.8215	0.8769	0.7465	0.8721
	HFN	0.6226	0.6678	0.7125	0.2735	0.3503	0.8181	0.8118	0.9445	0.7361	0.8676	0.7626	0.9031
	ICQ	0.7273	0.6334	0.4951	0.5592	0.5123	0.8261	0.7849	0.8350	0.8329	0.8134	0.7603	0.8302
	ID	0.5307	0.2249	0.4969	0.3623	0.2738	0.8694	0.7719	0.9102	0.5684	0.6434	0.7063	0.7488
	IN	0.4342	0.4257	0.4649	0.1107	0.1534	0.2866	0.5069	0.6696	0.1842	0.4551	0.6484	0.5838
	IS	0.0746	0.0821	0.1058	0.0757	0.0527	0.1406	0.1061	0.1699	0.0992	0.1165	0.3291	0.2092
	JPEG	0.6823	0.6914	0.6653	0.3506	0.5738	0.8961	0.8201	0.9158	0.7123	0.7964	0.6631	0.7907
	JPEGTE	0.4361	0.1138	0.2523	0.1024	0.0896	0.2925	0.5153	0.3795	0.2511	0.2131	0.2314	0.4353
	JPEG2k	0.8057	0.5692	0.7138	0.6557	0.3661	0.9099	0.8769	0.9407	0.8661	0.8507	0.7780	0.9369
	JPEG2kTE	0.6015	0.7531	0.3476	0.3769	0.1531	0.4394	0.5984	0.6552	0.5046	0.6711	0.6594	0.7388
	LBD	0.0969	0.1046	0.1453	0.1215	0.1135	0.1944	0.1311	0.1885	0.2374	0.1464	0.3813	0.2365
	LC	0.3242	0.1819	0.3226	0.2776	0.0876	0.5289	0.5692	0.8326	0.2565	0.3711	0.6533	0.3819
	MGN	0.4211	0.1281	0.5488	0.3085	0.1541	0.5324	0.6753	0.8471	0.6335	0.6666	0.6209	0.7512
	MN	0.1436	0.1988	0.1981	0.1546	0.2959	0.4168	0.5146	0.7290	0.3329	0.1535	0.4243	0.1638
	NEPN	0.1583	0.1009	0.1207	0.2603	0.0908	0.1534	0.2198	0.1545	0.3026	0.2558	0.1256	0.3712
	QN	0.7961	0.7711	0.6524	0.3618	0.5676	0.7869	0.8207	0.7890	0.8769	0.8623	0.7361	0.9173
	SCN	0.6546	0.6576	0.7911	0.1331	0.1126	0.6584	0.7192	0.8914	0.5803	0.7434	0.7015	0.6042
	SSR	0.7588	0.5781	0.6569	0.6623	0.5988	0.9088	0.8892	0.9391	0.6638	0.8488	0.8457	0.8357
	ALL	0.4631	0.3437	0.4072	0.2512	0.1377	0.6997	0.6417	0.7621	0.5901	0.6339	0.6078	0.7012
	Average	0.4532	0.3778	0.4241	0.2912	0.2417	0.5428	0.5902	0.7035	0.4746	0.5284	0.5652	0.5919
	STD	0.2460	0.2353	0.2308	0.1958	0.1705	0.2767	0.2418	0.2524	0.2562	0.2873	0.2098	0.2530
	MAX	0.8455	0.7711	0.8084	0.8072	0.5988	0.9099	0.8892	0.9445	0.8769	0.8769	0.8457	0.9369
	MIN	0.0746	0.0821	0.0816	0.0757	0.0527	0.1406	0.0846	0.1545	0.0992	0.0881	0.1256	0.1638

Table 3.8: Average SROCC of 100 runs of simulations on tested image databases using state-of-the-art IQA methods.

DB	DISTORTION	PSNR	SSIM	BRISQUE	CORNIA	CQA	SSEQ
LIVE 2	JPEG	0.8515	0.9481	0.8641	0.9002	0.8257	0.9122
	JPEG2k	0.8822	0.9438	0.8838	0.9246	0.8366	0.9388
	WN	0.9851	0.9793	0.9750	0.9500	0.9764	0.9544
	GB	0.7818	0.8889	0.9304	0.9465	0.8377	0.9157
	FF	0.8869	0.9335	0.8469	0.9132	0.8262	0.9038
	ALL	0.8013	0.8902	0.9098	0.9386	0.8606	0.9356
	Average	0.8648	0.9306	0.9017	0.9289	0.8605	0.9268
	STD	0.0726	0.0353	0.0469	0.0197	0.0582	0.0192
	MAX	0.9851	0.9793	0.9750	0.9500	0.9764	0.9544
	MIN	0.7818	0.8889	0.8469	0.9002	0.8257	0.9038
CSIQ	JPEG	0.9009	0.9309	0.8525	0.8319	0.6506	0.8066
	JPEG2k	0.9309	0.9251	0.8458	0.8405	0.8214	0.7302
	WN	0.9345	0.8761	0.6931	0.6187	0.7276	0.7876
	GB	0.9358	0.9089	0.8337	0.8526	0.7486	0.7766
	PN	0.9315	0.8871	0.7740	0.5340	0.5463	0.6661
	CD	0.8862	0.8128	0.4255	0.4458	0.5383	0.4172
	ALL	0.8088	0.8116	0.7597	0.6969	0.6369	0.7007
	Average	0.9041	0.8789	0.7406	0.6886	0.6671	0.6979
	STD	0.0462	0.0495	0.1502	0.1624	0.1053	0.1335
	MAX	0.9358	0.9309	0.8525	0.8526	0.8214	0.8066
	MIN	0.8088	0.8116	0.4255	0.4458	0.5383	0.4172
TID 2013	AGC	0.8568	0.7912	0.4166	0.2605	0.3964	0.3949
	AGN	0.9337	0.6421	0.6416	0.5689	0.6051	0.6040
	CA	0.7759	0.7158	0.7310	0.6844	0.4380	0.4366
	CC	0.4608	0.3477	0.1849	0.1400	0.2043	0.2006
	CCS	0.6892	0.7641	0.2715	0.2642	0.2461	0.2547
	CN	0.8838	0.6465	0.2176	0.3553	0.1623	0.1642
	GB	0.8905	0.8196	0.8063	0.8341	0.7019	0.7058
	HFN	0.9165	0.7962	0.7103	0.7707	0.7104	0.7061
	ICQ	0.9087	0.7271	0.7663	0.7044	0.6829	0.6834
	ID	0.9457	0.8327	0.5243	0.7227	0.6711	0.6716
	IN	0.9263	0.8055	0.6848	0.5874	0.4231	0.4272
	IS	0.7647	0.7411	0.2224	0.2403	0.2011	0.2013
	JPEG	0.9252	0.8275	0.7252	0.7815	0.6317	0.6284
	JPEGTE	0.7874	0.6144	0.3581	0.5679	0.2221	0.2195
	JPEG2k	0.8934	0.7531	0.7337	0.8089	0.7219	0.7205
	JPEG2kTE	0.8581	0.7067	0.7277	0.6113	0.6529	0.6529
	LBD	0.1301	0.6213	0.2833	0.2157	0.2382	0.2290
	LC	0.9386	0.8311	0.5726	0.6682	0.4561	0.4460
	MGN	0.9085	0.7863	0.5548	0.4393	0.4969	0.4897
	MN	0.8385	0.7388	0.2650	0.2342	0.2506	0.2575
	NEPN	0.6931	0.5326	0.1821	0.2855	0.1308	0.1275
	QN	0.8636	0.7428	0.5383	0.4922	0.7242	0.7214
	SCN	0.9152	0.7934	0.7238	0.7043	0.7121	0.7064
	SSR	0.9241	0.7774	0.7101	0.8594	0.8115	0.8084
	ALL	0.6869	0.5758	0.5416	0.6006	0.4925	0.4900
	Average	0.8126	0.7172	0.5238	0.5361	0.4794	0.4779
	STD	0.1814	0.1135	0.2145	0.2258	0.2191	0.2186
	MAX	0.9457	0.8327	0.8063	0.8594	0.8115	0.8084
	MIN	0.1301	0.3477	0.1821	0.1400	0.1308	0.1275

Chapter 4

Video Quality Assessment Using Spatiotemporal Texture Measures

4.1 Overview

4.2 Video Quality Assessment Method

In the proposed OQVA method, separate feature sets are computed independently from each other. These features sets are the following:

- Multiscale salient local binary patterns (MSLBP),
- Multiscale structural similarity (MSSIM) [74],
- Gradient magnitude similarity deviation (GMSD) [75],
- Riesz pyramids similarity deviation (RPSD, proposed in this work),
- Spatial activity (SA) and Temporal distortion measures (TDM) [139].

Each of these feature sets are computed for the reference (SRC) and the test (PVS) videos. For each feature component, a pooling strategy is adopted and the pooled values are concatenated to generate a feature vector. Finally, the feature vector is used as input in a random forest regressor (RFR) to predict the quality score. Fig. 4.1 depicts a block diagram of the proposed method. In the next sections, we describe each feature set used in the proposed method.

4.2.1 Spatial Activity

The Spatial Activity (SA) of a pair frames (SRC and PVS) is computed by taking the root mean square (RMS) difference between the Sobel maps of each of the frames. More

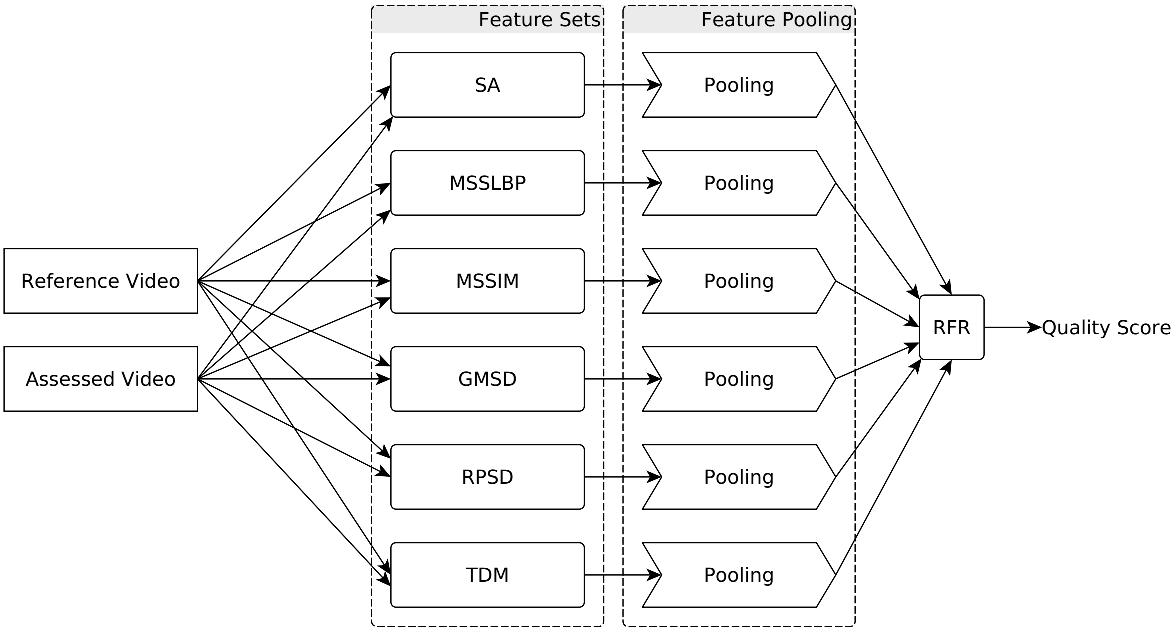


Figure 4.1: Block diagram of proposed method.

specifically, let $\mathcal{S}$ be the Sobel operator [140] defined as:

$$\mathcal{S}(z) = \sqrt{(G_1 * z)^2 + (G_1^\top * z)^2}, \tag{4.1}$$

where z is the frame picture, $*$ denotes the 2-dimensional convolution operation, G_1 is the vertical Sobel filter, given by:

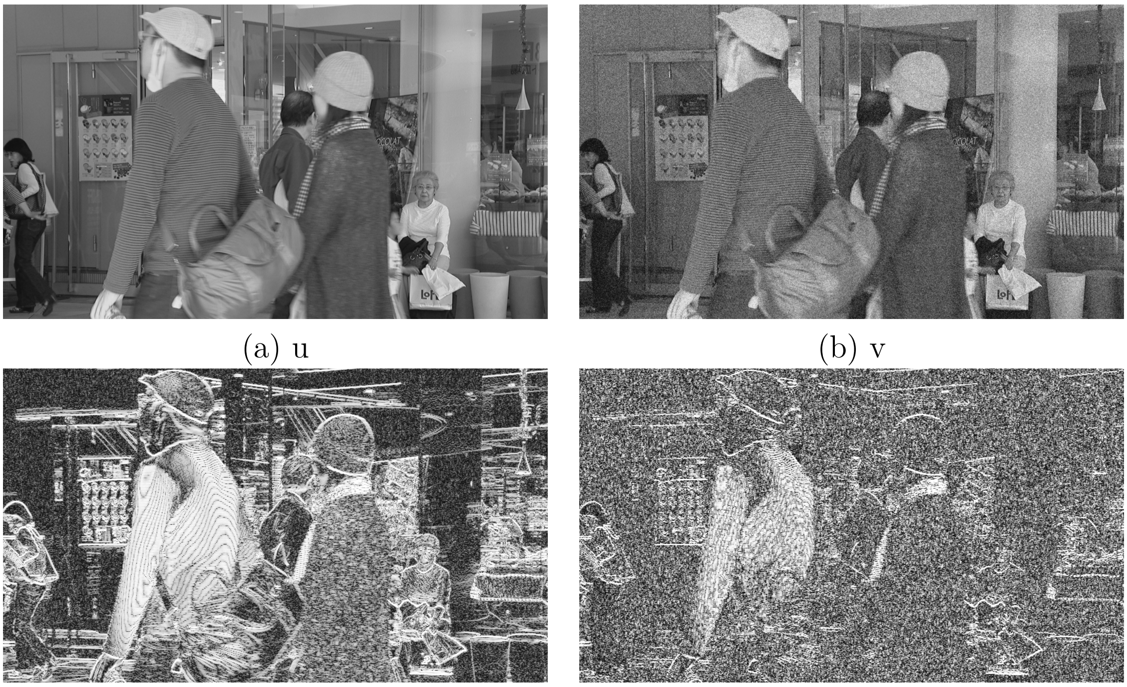
$$G_1 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \tag{4.2}$$

and $G_1^\top$ is the transpose of G_1 (horizontal Sobel filter).

Fig. 4.2 illustrates how Sobel operator captures spatial distortions. Fig. 4.2-(a) and (b) show the frames with and without distortions, respectively. Their corresponding Sobel maps are shown in Fig. 4.2-(c) and (d). Notice that the small differences between distorted and original frames are emphasized by the Sobel operator.

Considering that v is a frame from SRC and u is the same frame from PVS, we first compute the Sobel map of both frames and take the difference between these two maps:

$$s = \mathcal{S}(u) - \mathcal{S}(v). \tag{4.3}$$



(a) u

(b) v

(c) $\mathcal{S}$(u)

(d) $\mathcal{S}$(v)

Figure 4.2: Effect of spatial activity on Sobel filter: (a) Reference frame, (b) distorted frame, (c) Sobel map of reference frame, and (d) Sobel map of distorted frame.

Then, we compute SA using the following equation:

$$SA(v, u) = \sqrt{\frac{1}{MN} \sum_{i,j} |s_{ij}|^2}, \tag{4.4}$$

where i and j correspond to the horizontal and vertical indices of s, respectively, and M and N are the height and width of the frames, respectively.

4.2.2 Multiscale Salient Local Binary Patterns

The second feature set is composed of histograms of MSLBP, computed as described in Section 2.4.6. Although the resulting histograms of the MSLBP operator describes the frame textures, it does not describe the relationships between SRC and PVS distortions. Moreover, the amount of components generated from these histograms can be an issue, especially when we take into account the various frames of the video.

To overcome the above mentioned issue, we designate $H^{v}_{R,P}$ and $H^{u}_{R,P}$ the SRC and PVS frame histograms associated to R and P, respectively. To quantify the relationship between SRC and PVS texture information, we compute the similarity between the $H^{V}_{R,P}$

and $H_{R,P}^{U}$ using the following equations:

$$JSD(p,q) = \frac{1}{2}\left(KLD(p,r) + KLD(q,r)\right), \tag{4.5}$$

where

$$KLD(p,q) = \sum p(x)\log\left(\frac{p(x)}{q(x)}\right), \tag{4.6}$$

and

$$r = \frac{p(x)+q(x)}{2}. \tag{4.7}$$

In the above equations, $p(x)$ and $q(x)$ are probability mass functions (PMF) representing the bins of $H_{R,P}^{V}$ and $H_{R,P}^{U}$, JSD is the Jensen-Shannon divergence [141], and KLD is the Kullback-Leibler divergence [142]. JSD is chosen because it is the symmetric version of mutual information (KLD) and it is always a finite value [143].

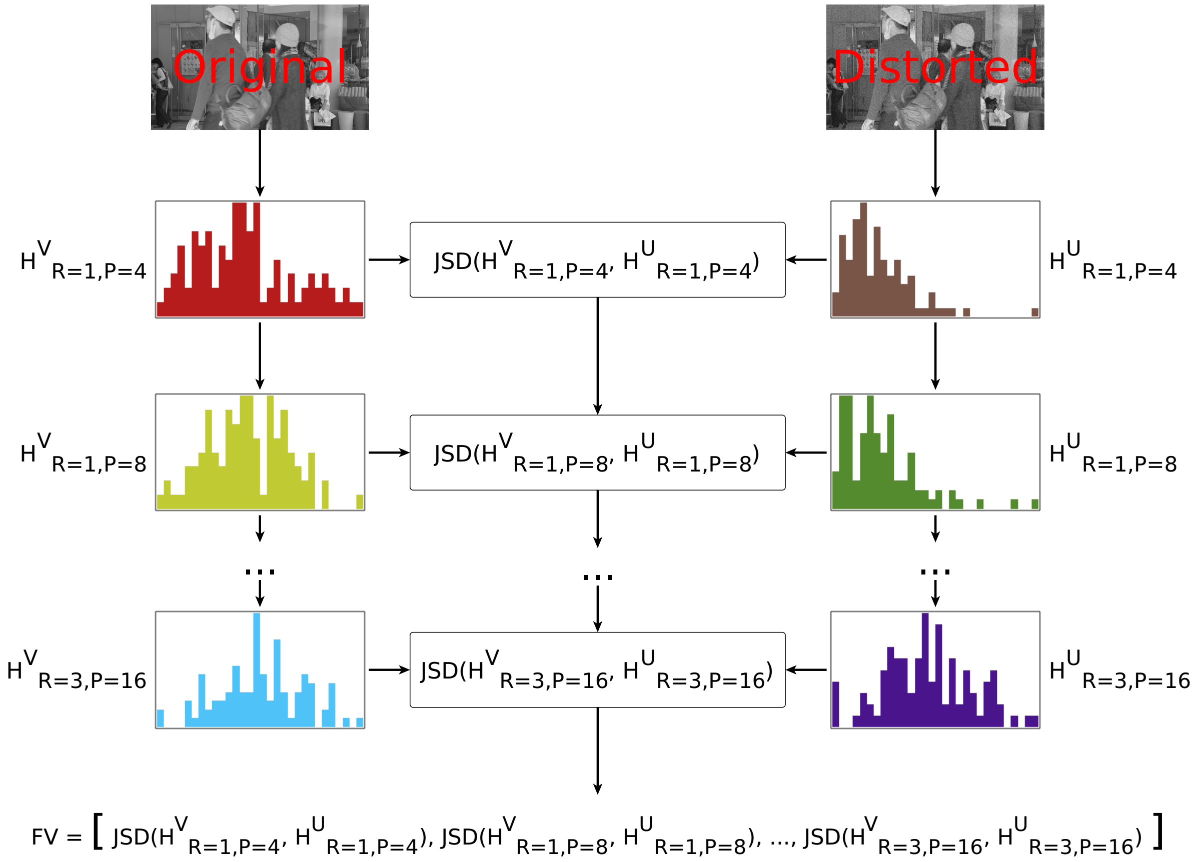

Original
Distorted
H^V R=1,P=4
JSD(H^V R=1,P=4, H^U R=1,P=4)
H^U R=1,P=4
H^V R=1,P=8
JSD(H^V R=1,P=8, H^U R=1,P=8)
H^U R=1,P=8
...
H^V R=3,P=16
JSD(H^V R=3,P=16, H^U R=3,P=16)
H^U R=3,P=16
FV = [JSD(H^V R=1,P=4, H^U R=1,P=4), JSD(H^V R=1,P=8, H^U R=1,P=8), ..., JSD(H^V R=3,P=16, H^U R=3,P=16)]

Figure 4.3: Generation of MSSLBP feature set.

Fig. 4.3 illustrates the final construction of MSLBP feature set. After perform the steps depicted in Fig. 2.22 for both SRC and PVS frames, the JSD is used to compute the divergences between the SRC and PVS histograms in each scale. These divergences compose the feature vector (FV) equivalent to the MSLBP feature set.

4.2.3 Multiscale Structural Similarity

The Structural SIMilarity (SSIM) index is a popular IQA method based on luminance, contrast, and structure measures [74]. It is calculated using the following equation:

$$SSIM(u,v) = \frac{(2\mu_u\mu_v + C_1)(2\sigma_{uv} + C_2)}{(\mu_u^2 + \mu_v^2 + C_1)(\sigma_u^2 + \sigma_v^2 + C_2)}, \tag{4.8}$$

where μ_f, σ_f are the average and standard deviation of the frame f, σ_{fg} is the covariance of frames f and g, and C_1 and C_2 are constants used to stabilize divisions with a weak denominator. In this work, we use the mean of the SSIM index map to quantify the quality difference between of u and v, at different scales [144]. Fig. 4.4 depicts the feature extraction using the SSIM index for different scales, while Fig. 4.5 shows the SSIM maps for four different scales.

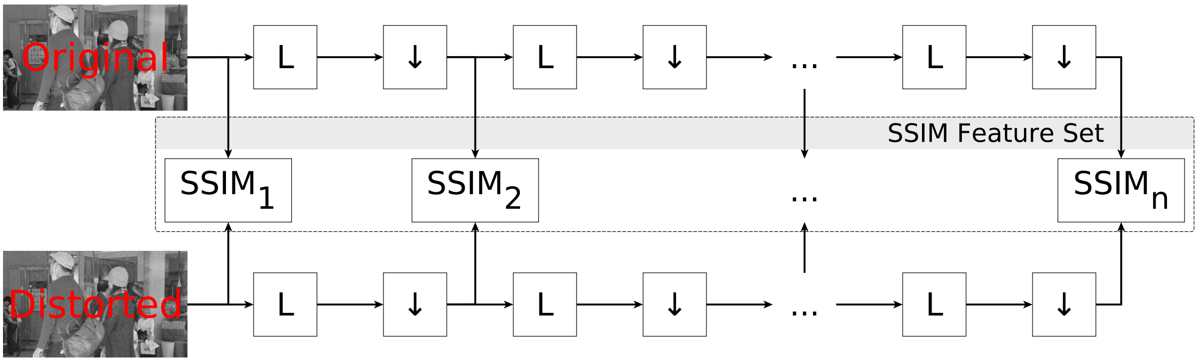


Figure 4.4: Multi-scale structural similarity measurement. L: low-pass filtering, ↓: downsampling by factor 2.

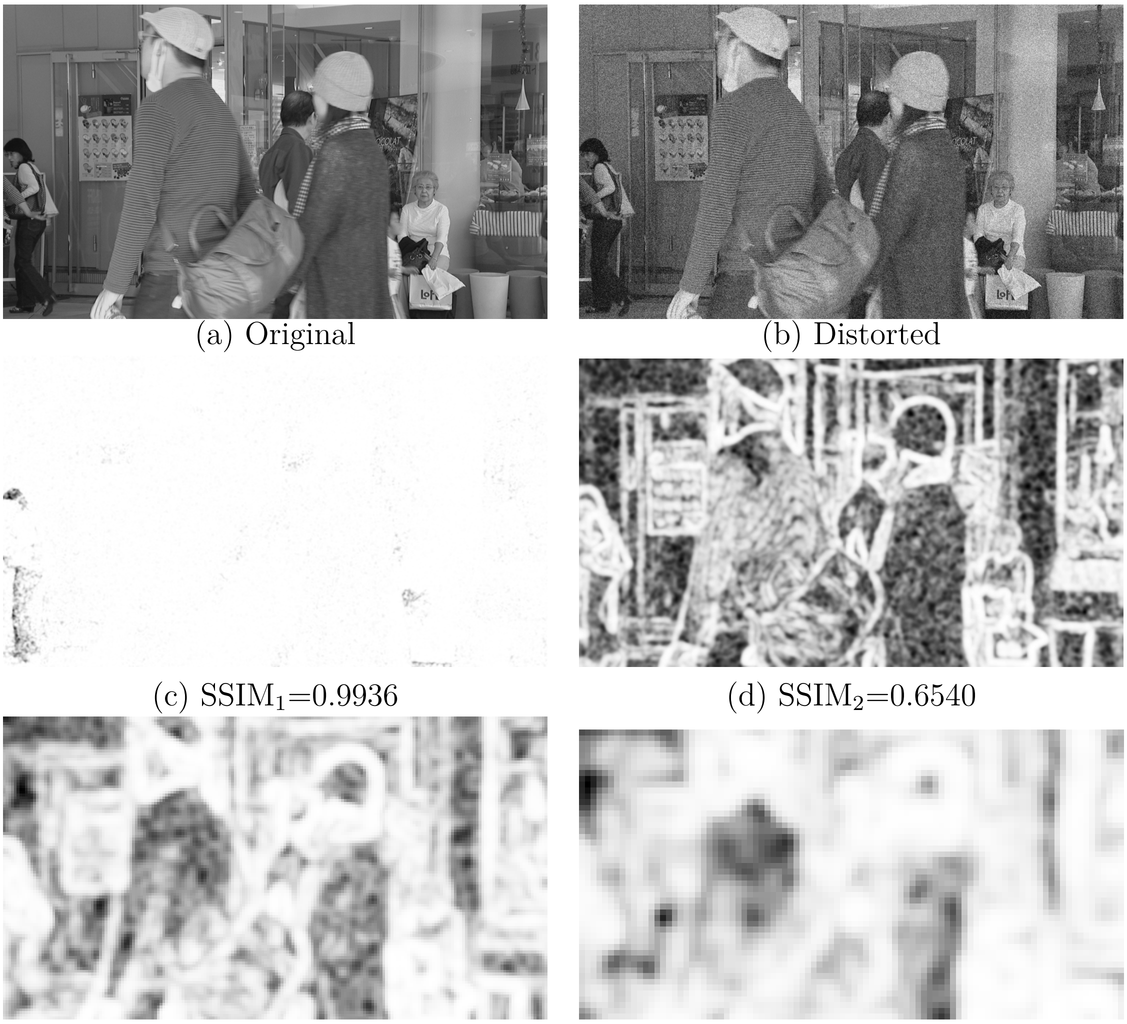
4.2.4 Gradient Magnitude Similarity Deviation

Gradient Magnitude Similarity Deviation (GMSD) is an IQA method based on the standard deviation of the gradient magnitude similarity (GMS) map [75]. The GMS map is computed as follows:

$$GMS(u,v) = \frac{2 \cdot m(u) \cdot m(v) + c}{m(u)^2 + m(v)^2 + c}, \tag{4.9}$$

where u is the SRC frame, v is the PVS frame, c is a positive constant that garantees numerical stability, and $m(z)$ is defined as:

$$m(z) = \sqrt{(z * G_2)^2 + (z * G_2^\top)^2}. \tag{4.10}$$



(a) Original

(b) Distorted

(c) SSIM$_1$=0.9936

(d) SSIM$_2$=0.6540

(e) SSIM$_3$=0.8074

(f) SSIM$_4$=0.8628

Figure 4.5: Multi-scale structural similarity maps.

In the above equation, $*$ denotes the convolution operation, G_2 represents the Prewitt filter along the vertical direction, which is defined as:

$$G_2 = \begin{bmatrix} \frac{1}{3} & 0 & -\frac{1}{3} \\ \frac{1}{3} & 0 & -\frac{1}{3} \\ \frac{1}{3} & 0 & -\frac{1}{3} \end{bmatrix}, \tag{4.11}$$

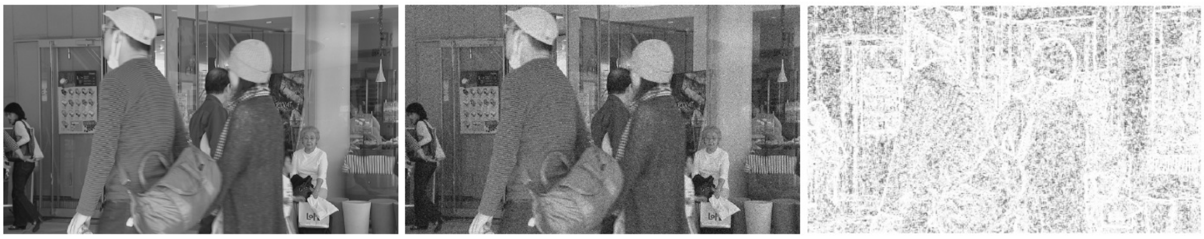
and $G_2^\top$ is the transpose of G_2, which corresponds to the Prewitt filter along the horizontal direction.

Fig. 4.6-(c) depicts the GMS map, which serves as a local quality map of the distorted frame. The GMSD index is computed as follows:

$$GMSD(u,v) = \sqrt{\frac{1}{NM}\sum_{i,j}\left(GMS(u,v) - \overline{GMS(u,v)}\right)^2}, \tag{4.12}$$

where $\overline{GMS(u,v)}$ is the gradient magnitude similarity mean, computed as follows:

$$\overline{GMS(u,v)} = \frac{1}{NM}\sum_{i,j} GMS(u,v). \tag{4.13}$$



(a) Original (b) Distorted (c) GMS map

Figure 4.6: Gradient magnitude similarity map.

4.2.5 Riesz Pyramids Similarity Deviation

Wadhwa *et al.* proposed a technique technique to represent images, which is called Riesz pyramids [145]. Their work was inspired by the work of Simoncelli and Freeman [146]. The Riesz pyramids make use of a highpass filter $h_H[n]$ and a lowpass filter $h_L[n]$. First, the frame is highpassed to generate the top level of the pyramid. Next, the frame is lowpassed and downsampled. This process is recursively applied to the downsampled image to generate the pyramid representation, as illustrated in Fig. 4.7.

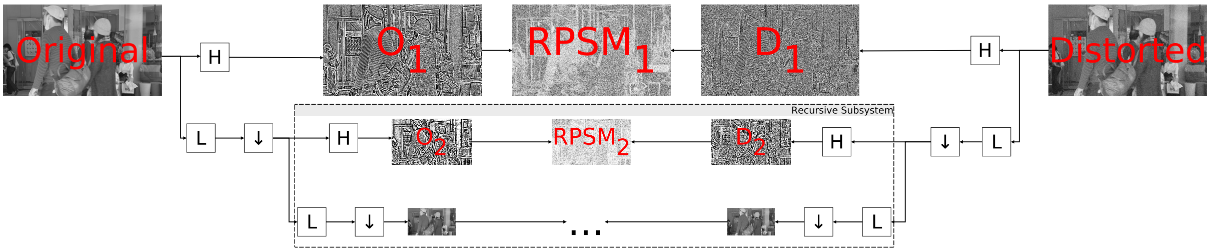


Figure 4.7: Diagram of RPSD pyramid construction of both original and distorted frames. The lowpass and highpass filters can be recursively used with subsampling to produce a sequence of critically bandpassed frames. The blocks ↓ denote downsampling by a factor of 2. L and H denote linear shift invariant lowpass and highpass filters, respectively.

To generate the features using the Riesz pyramids, we compare the highpassed v frames with the highpassed u frames at each pyramid level. More specifically, for each level i, we compute the highpassed version of the u (u_i^h) and v (v_i^h) frames. Using u_i^h and v_i^h, we generate the i-th Riesz pyramid similarity map (RPSM$_i$), as follows:

$$\mathrm{RPSM}_i(u,v) = \frac{2 \cdot n(u_i^h) \cdot n(v_i^h) + c}{n(u_i^h)^2 + n(v_i^h)^2 + c}, \tag{4.14}$$

where

$$n(z) = \sqrt{(z * G_3)^2 + (z * G_3^\top)^2} \tag{4.15}$$

and

$$G_3 = \begin{bmatrix} \frac{1}{2} & 0 & -\frac{1}{2} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ \frac{1}{2} & 0 & -\frac{1}{2} \end{bmatrix}. \tag{4.16}$$

From the the *i-th* Riesz pyramid similarity map, we compute the Riesz Pyramids Similarity Deviation (RPSD), as follows:

$$\mathrm{RPSD}_i(u,v) = \sqrt{\frac{1}{NM}\sum_{i,j}\left(\mathrm{RPSM}_i(u,v) - \overline{\mathrm{RPSM}_i(u,v)}\right)^2}, \tag{4.17}$$

where $\overline{\mathrm{RPSM}_i(u,v)}$ is the mean of *i-th* RPSM map, computed as follows:

$$\overline{\mathrm{RPSM}_i(u,v)} = \frac{1}{NM}\sum_{i,j}\mathrm{RPSM}_i(u,v). \tag{4.18}$$

Finally, the RPSD feature set is composed by all *i-th* RPSD values, i.e., $\mathrm{RPSD}(u,v) = \{\mathrm{RPSM}_1(u,v), \mathrm{RPSM}_2(u,v), \cdots\}$.

4.2.6 Temporal Distortion Measures

Temporal distortion measures (TDM) are based on the spatio-temporal texture representation proposed by Derpanis & Wilders [139]. The texture representation is computed using a bank of spatio-temporal 3-rd derivative Gaussian filters, defined as follows:

$$\mathcal{G}_{3_\theta} = \frac{\partial^3}{\partial\theta^3} k e^{-(x^2+y^2+t^2)}, \tag{4.19}$$

where θ is the unit vector that corresponds to the spatio-temporal direction of the filter and k is a normalization factor [147].

First, we compute the sum of the pointwise squared response of the filter output over a space-time region, Ω, producing the following energy measurement:

$$\mathcal{E}_{3_\theta}(x,y,t) = \sum_{x,y,t\in\Omega} (\mathcal{G}_{3_\theta} * V(x,y,t))^2, \tag{4.20}$$

where V is the input spatio-temporal signal (video).

Let $\mathcal{E}^v_{k_\theta}$ and $\mathcal{E}^u_{k_\theta}$ be the energy measurements, along the direction k_θ of u and v, respectively. The temporal distortion (TD) measure at (x,y,t) is obtained by computing the distance between the two corresponding energy distributions in u and v:

$$\mathrm{TD}(x,y,t) = \sqrt{\sum_k (\mathcal{E}^v_{k_\theta}(x,y,t) - \mathcal{E}^u_{k_\theta}(x,y,t))^2}. \tag{4.21}$$

Finally, the TDM is computed along time:

$$\mathrm{TDM}(t) = \sum_{x,y} \mathrm{TD}(x,y,t). \tag{4.22}$$

4.2.7 Feature Pooling

The formulation presented in the previous sections describes the generation of a set of values corresponding to each feature. To convert these sets of values into a single score per feature, we use a feature pooling strategy based on the Minkowski norm. The Minkowski norm is computed as follows:

$$Q_k = \sqrt[4]{\sum_t f_k(t)^4}, \tag{4.23}$$

where $f_k(t)$ corresponds to the k-th feature at its t-th value.

Next, the pooled features $Q_1, Q_2, \cdots Q_n$ are treated as inputs to a random forest regression (RFR) algorithm, which gives an estimated video quality score. We choose the RFR method, rather than the popular support vector regression (SVR), because RFR does not require the hyper-parameter tuning. Furthermore, RFR has been successfully used in several pattern recognition applications [133].

4.3 Experimental Setup and Protocol

There are a number of existing databases created for the evaluation of video quality. In this work, we use the following databases:

- Computational and Subjective Image Quality (CSIQ) Video Database [148]: The database contains 12 high-quality reference videos and 216 distorted videos from 6 different types of distortion. All videos are in raw YUV420 format with a resolution of 832x480 pixels, a duration of 10 seconds at 24, 25, 30, 50, or 60 fps. The distortion types consist of 4 compression-based distortion types including H.264 compression (H.264), HEVC/H.265 compression (HEVC), Motion JPEG compression (MJPEG), and Wavelet-based compression using the Snow codec (SNOW). The database also include 2 transmission-based distortion types, namely wireless transmission loss (WIRELESS), and additive white noise (WN).

- Image and Video Processing Laboratory (IVPL) Database [149]: The database contains 10 reference videos and 128 distorted videos from 4 different types of distortion. All videos are in raw YUV420 format with a resolution of 1920x1088 (progressive) at 25 fps. The distortion types consist of 3 compression-based distortion types including H.264 compression (H.264), Dirac coding (DIRAC), and MPEG2. The database also include 1 transmission-based distortion (IP).

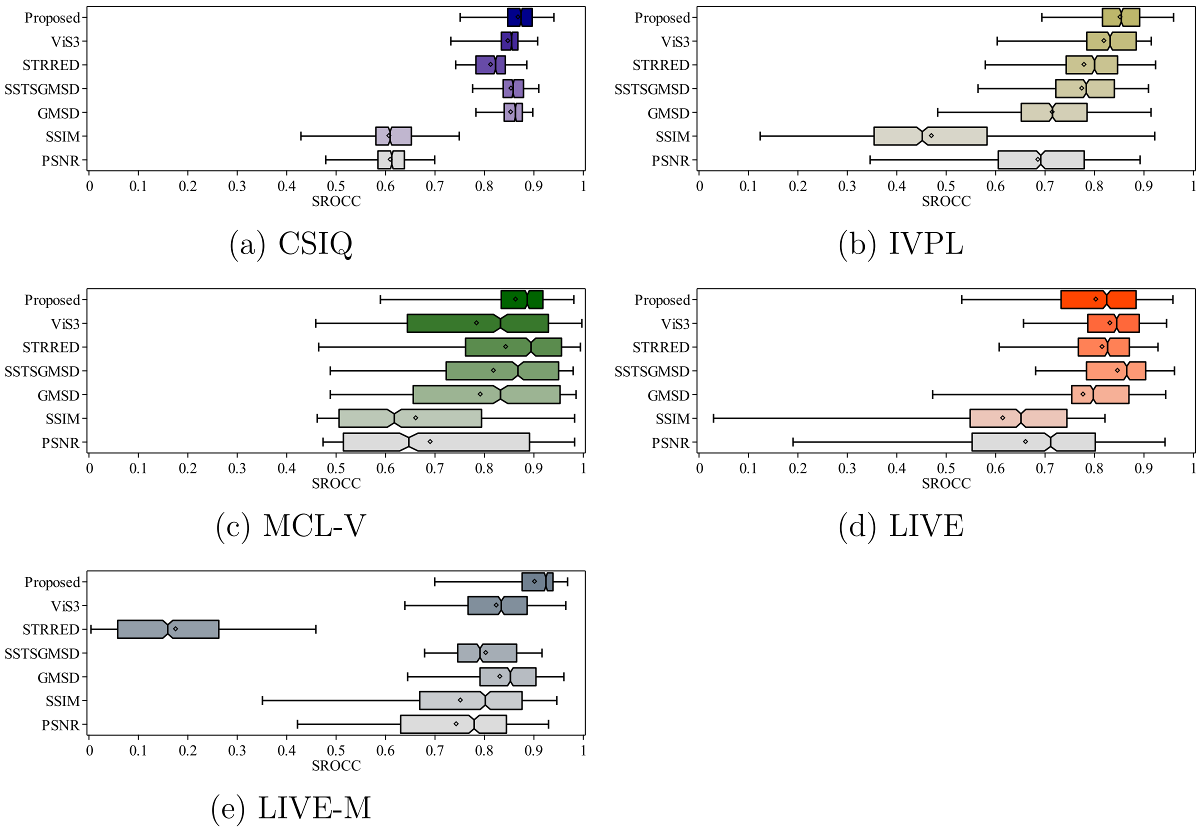
- MediaCommLab Video (MCL-V) Database [150]: The database contains 12 uncompressed source video clips with HD resolution (1080p). The database captures two typical video distortion types in video streaming services, including compression (H.264) and image size scaling (SD H.264). Four distortion levels are adopted for each distortion type. There are 96 distorted video clips in total.

- Laboratory for Image & Video Engineering (LIVE) Video Database [151]: The database contains 10 high-quality reference videos with 15 distorted videos per reference in a total of 150 test videos. The videos files have planar YUV420 format with spatial resolution of 768x432 pixels. The distortion types consist of 2 compression-based distortion types, including H.264 compression (H.264) and MPEG2. The database also include 2 transmission-based distortion, including simulated transmission of H.264 compressed bitstreams through (1) error-prone IP networks and (2) error-prone wireless networks. These two distortions are grouped into a single category, named “transmission errors” (TE).

- LIVE Public-Domain Subjective Mobile Video Quality Database (LIVE-M) [152, 153]: The database consists of 10 raw HD reference videos and 200 distorted videos (4 compression + 4 wireless packet-loss + 4 frame-freezes + 3 rate-adapted + 5 temporal dynamics per reference), each of resolution 1280x720 at a frame rate of 30 fps, and of duration 15 seconds each. For testing purposes, we excluded the frame-freezes distortions.

In our test methodology, we compare the proposed method with a set of publicly available standard-of-the-art VQA methods. The chosen VQA methods are SSTSGMSD [154], STRRED [32], and ViS3 [155]. Additionally, we also compared the proposed algorithm with three well-established IQA metrics, namely PSNR, SSIM [74], GMSD [75]

To estimate the performance of the tested methods, we calculate the Spearman's rank correlation coefficient (SROCC), which gives a measure of how well the subjective scores and the scores produced from the video quality methods are correlated. In addition to SROCC, we also calculated Pearson and Kendall correlation coefficients. But, since results for these metrics were very similar to what was obtained with SROCC, we only report the SROCC results.



(a) CSIQ

(b) IVPL

(c) MCL-V

(d) LIVE

(e) LIVE-M

Figure 4.8: Distribution of average SROCC after 1000 simulations.

4.4 Experimental Results

Fig. 4.8-(a) presents the SROCC results for the CSIQ dataset. Notice that the proposed method shows the highest SROCC value, when compared to the state-of-the-art metrics, followed by SSTSGMSD, GMSD, ViS3, and STRRED. Since CSIQ contains two transmission-based distortions, it is expected that IQA methods present a worse performance, which explains the differences between the PSNR and SSIM results when compared with other methods. Surprisingly, GMSD presents a competitive performance, having a performance similar to its video-based version, SSTSGMSD.

Among the different tested quality metrics, the proposed approach has the highest SROCC in the IVPL and LIVE-M datasets, as displayed in Figs. 4.8-(b) and (e). Notice that both mean and median correlation values are higher for the proposed method. Taking a closer look at the interquartile range of the methods, we notice that the proposed method also presents a smaller dispersion, what indicates that it is more stable along multiple simulations. Therefore, the proposed method is significantly better than all tested methods on IVPL and LIVE-M datasets.

Fig. 4.8-(c) shows the SROCC values for the MCL-V dataset. Notice that the performance of the proposed method is among the best performances, although its SROCC median value overlaps with the SROCC median value for STRRED and SSTSGMSD. However, observing again the interquartile range, we notice that the proposed method has a narrow range and, therefore, represents a smaller dispersion. Taking this into consideration, the performance of the proposed method is slightly better than the performance of STRRED and SSTSGMSD and significantly better than the performance of ViS3, GMSD, SSIM, and PSNR.

Fig. 4.8-(d) shows the SROCC results for the LIVE dataset. In this case, the proposed method has one of the highest median SROCC values. While this performance is not significantly better than the performance of ViS3, STRRED and SSTSGMSD, it outperforms other state-of-the-art methods.

Finally, Fig. 4.8-(e) depicts the SROCC boxplot for the LIVE-M dataset. Notice that, in this database, the proposed method represents a substantial prediction improvement in relation to the other metrics. In addition to a higher median value, the proposed method has a smaller confidence interval. More specifically, the interquartile range indicates that the proposed method has a narrower spread of the correlation values.

Table 4.1 depicts the average SROCC results, separated for the different distortions in each database. In this table, each row of the first column specifies the database, while the second column lists the distortion of the given database. The distortion named as 'ALL' corresponds to the general case that includes PVS videos with all types of distortions, i.e. it corresponds to the results presented in Fig. 4.8. In each line, the highlighted values (in bold) represent the best average SROCC values for each distortion.

Notice that the proposed method is among the top best three metrics. It is worth mentioning that, among all subsets, the proposed methods is the only one that has average SROCC values greater than 0.8 for almost all distortions. The only exception is the 'Temporal Dynamics' distortion of the LIVE-M dataset. These results indicate that the proposed method is the most adequate for practical multimedia scenarios, in which several types of distortions are present (besides compression and transmission distortions). Furthermore, the proposed method presents the best results for 20 out of the 24 cases (83.3%), what makes it the method with the highest prediction accuracy.

Table 4.1: Average SROCC of 1,000 runs of simulations on tested video databases

Database	Distortion	PSNR	SSIM	GMSD	SSTSGMSD	STRRED	ViS3	PROPOSED
CSIQ	H.264	0.8023	0.8440	0.9436	0.9137	**0.9768**	0.9194	0.9419
	HEVC	0.7846	0.8136	0.9418	0.9292	0.9135	0.9173	**0.9631**
	MJPEG	0.5086	0.7969	0.8842	0.8803	0.7289	0.7348	**0.9066**
	SNOW	0.7586	0.7539	0.8659	0.8584	**0.9459**	0.8998	0.9071
	WIRELESS	0.8512	0.8317	**0.8680**	0.8187	0.8476	0.8442	0.8122
	WN	0.9063	0.9300	0.9094	0.8818	0.9305	0.9202	**0.9492**
	ALL	0.6104	0.6076	0.8539	0.8546	0.8133	0.8482	**0.8688**
IVPL	DIRAC	0.8532	0.7846	0.8229	0.8140	0.8527	0.9132	**0.9289**
	H.264	0.8154	0.6636	0.8690	0.8435	0.8614	0.8425	**0.8781**
	MPEG2	0.6974	0.5884	0.8318	0.7864	0.6774	0.7939	**0.9165**
	IP	0.6283	0.0481	0.7093	0.7695	0.6650	0.7246	**0.8692**
	ALL	0.6863	0.4710	0.7153	0.7751	0.7796	0.8198	**0.8525**
MCL-V	H.264	0.4215	0.3545	0.6419	0.6946	0.7716	0.5868	**0.8671**
	SD H.264	0.4925	0.4400	0.6376	0.6817	0.7040	0.6890	**0.8741**
	ALL	0.6911	0.6619	0.7924	0.8191	0.8438	0.7848	**0.8638**
LIVE	H.264	0.4729	0.6561	0.6471	0.7938	0.8193	0.7685	**0.8809**
	MPEG2	0.3830	0.5609	0.6915	0.8123	0.7193	0.7362	**0.8819**
	TE	0.5798	0.5151	0.7457	0.8157	0.7934	0.8372	**0.8721**
	ALL	0.6614	0.6152	0.7776	**0.8473**	0.8162	0.8320	0.8035
LIVE-M	Compression	0.8270	0.7172	0.8662	0.8713	0.0881	0.8607	**0.9524**
	Rate Adaptation	0.6353	0.6014	0.7312	0.7666	0.0629	0.7550	**0.9429**
	Temporal Dynamics	0.2917	0.2850	0.3649	0.3752	0.1979	0.3746	**0.3901**
	Wireless PL	0.7897	0.6929	0.8468	0.8286	0.1014	0.8394	**0.9661**
	ALL	0.7437	0.7522	0.8320	0.8035	0.1762	0.8248	**0.9023**

4.5 Summary and Conclusions

In this chapter, we introduced a new method to assess the quality of a given video using a full reference approach. The proposed video quality metric is based on multiple spatio-temporal features. A random forest regression algorithm is used to map these features into the subjective scores. Based on the calculated Spearman correlation values, the proposed approach outperforms the state-of-the-art video quality metrics in most datasets and distortion types. In the cases where the proposed approach is not the best video quality metric, it still stands among the top approaches, providing competitive prediction performance. In future works, we will investigate how to adapt the feature sets to perform reduced and no-reference quality assessment.

Chapter 5

Conclusions

In this thesis, our goal was to investigate how to estimate digital image and video quality for real-time applications using both blind and full-reference objective metrics, following a machine learning approach. In such approach, visual quality methods were generated using texture features. We presented two methods, one for predicting image quality with no reference and other to predict video quality with full reference. First, we presented a general framework to predict image quality using texture operators. Specifically, the chosen texture generator was the Local Binary Pattern (LBP) and some of its extensions. Then, we adapted texture features for video quality assessment.

In Chapter 2, we presented algorithms for generating texture features using the local binary pattern operator and some of its extensions. Among these algorithms are some previously proposed in the literature, but we also propose some quality-aware extensions.

Using the algorithms described in Chapter 2, we have performed several simulations to determine which texture operators are suitable for measure image quality with no references. In Chapter 3, we presented a general framework to construct blind image quality metrics using these operators. In this chapter, we also presented the results of a experimental evaluation of these operators. The analysis of these results shows that some proposed operators present the best performance on quality predictions of images, notably the proposed Multiscale Salient Local Binary Patterns (MSLBP) and Orthogonal Color Planes Patterns (OCPP) operators.

Chapter 4 describes a full-reference video quality assessment method based on MSLBP operator. Among all the operators described in the Chapter 2 and tested on Chapter 3, the MSLBP operator was chosen because it incorporates a visual attention model. As depicted by Akamine and Farias [130], the addition of saliency maps improves the performance quality metrics, especially spatial metrics. Further to MSLBP, other sets of features were also incorporated into the proposed metric, including spatial activity, structural similarity, Riesz pyramids, and temporal distortion measures. These features combined with a random forest regression generate a model that present substantial prediction performance, outperforming several of the best state-of-the-art video metrics.

5.1 Summary of the Contributions

The main contributions of this thesis are

- Development of a set of extensions of LBP operator for generating quality-aware features to predict visual quality.
- Development of a machine learning framework for using LBP-based operators to predict image quality assessment with no reference.
- Development of a full-reference video quality assessment method.

5.2 Future Work

Extensions of LBP are usually proposed in order to improve the prediction performance of some problem, such as image face recognition [99], iris recognition [108], text recognition [126], etc. However, further to the extensive review of the state-of-the-art LBP extensions, we also proposed new LBP variants in Chapter 2. Although the proposed extensions were designed for quality prediction tasks, we also can investigate their usage in other applications.

Among the possible applications, the use of LBP extensions to create visual attention models is particularly interesting. As shown in Chapters 3 and 4, the incorporation of visual saliency improves significantly the visual quality prediction. Therefore, by using LBP-based information to model both visual saliency and texture statistics in a manner similar to that adopted in MSLBP operator, we can produce a simpler visual quality model.

In Chapter 4, we presented a new full-reference video quality metric. Although the obtained results overcome other state-of-the-art methods, the use of a reference video is a severe drawback for many applications. Therefore, we have to investigate strategies to convert the proposed full-reference method into a no-reference method. A possible approach to achieve this aim is to use a bag-of-features for creating a visual vocabulary [156]. The idea is to replace the comparison between reference and test video by a collections of local features (visual vocabulary). The quality assessment is performed by comparing the features of assessed video with the clustered features of the visual vocabulary.

Another possible future work is to improve the computational performance of the proposed metric using parallel computing techniques. Most of the algorithms presented in

Chapter 2 are pixelwise operators and, therefore, highly parallelizable. Implementations of LBP-based operators using CUDA [157] or OpenMP [158] technologies can benefit several real-world application involving images and videos that demands real-time processing.

References

[1] Cisco, Cisco Visual Networking Index: *Global mobile data traffic forecast update, 2013–2018.* white paper, 2014. 1

[2] ITU-R: *Bt.500-13, methodology for the subjective assessment of the quality of television pictures.* 2012. 1

[3] REC, ITUT: *P. 800.1, mean opinion score (mos) terminology*, 2006. 2

[4] Shahid, Muhammad, Andreas Rossholm, Benny Lövström, and Hans Jürgen Zepernick: *No-reference image and video quality assessment: a classification and review of recent approaches.* EURASIP Journal on Image and Video Processing, 2014(1):1–32, 2014. xiii, 2, 5, 6

[5] Farias, Mylène CQ, Marco Carli, and Sanjit K Mitra: *Objective video quality metric based on data hiding.* Consumer Electronics, IEEE Transactions on, 51(3):983–992, 2005. 2

[6] Chikkerur, Shyamprasad, Vijay Sundaram, Martin Reisslein, and Lina J Karam: *Objective video quality assessment methods: A classification, review, and performance comparison.* IEEE transactions on broadcasting, 57(2):165–182, 2011. 2, 3

[7] Ebrahimi Moghadam, Abbas, Pedram Mohammadi, and Shahram Shirani: *Subjective and objective quality assessment of image: A survey.* Majlesi Journal of Electrcial Engineering, 9, 2015. 2

[8] Farias, Mylene Christine Queiroz de: *No-reference and reduced reference video quality metrics: new contributions.* University of California, Santa Barbara, 2004. 3, 4, 5, 6

[9] Guo, Linfeng and Yan Meng: *What is wrong and right with mse.* In *Eighth IASTED International Conference on Signal and Image Processing*, pages 212–215, 2006. 3

[10] Kunt, M and C van den Branden Lambrecht: *Special issue on image and video quality metrics.* Signal Processing, 70(3), 1998. 3

[11] Lubin, J: *A visual discrimination mode for image system design and evaluation.* Visual Models for Target Detection and Recognition (E. Peli, ed.) O, 2, 1993. 3

[12] Daly, Scott: *Digital images and human vision.* In Watson, Andrew B. (editor): *Digital images and human vision*, chapter The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity, pages 179–206. MIT Press, Cambridge, MA, USA, 1993, ISBN 0-262-23171-9. `http://dl.acm.org/citation.cfm?id=197765.197783`. 3

[13] Watson, Andrew B, James Hu, and John F McGowan: *Digital video quality metric based on human vision.* Journal of Electronic imaging, 10(1):20–29, 2001. 3

[14] Teo, Patrick C and David J Heeger: *Perceptual image distortion.* In *IS&T/SPIE 1994 International Symposium on Electronic Imaging: Science and Technology*, pages 127–141. International Society for Optics and Photonics, 1994. 3

[15] Watson, Andrew B: *Perceptual-components architecture for digital video.* JOSA A, 7(10):1943–1954, 1990. 3

[16] Winkler, Stefan: *Perceptual distortion metric for digital color video.* In *Electronic Imaging'99*, pages 175–184. International Society for Optics and Photonics, 1999. 3

[17] Watson, Andrew B: *Towards a visual quality metric for digital video.* In *Proc. European Signal Processing Conference*, volume 2, 1998. 3

[18] Branden Lambrecht, Christian J Van den and Murat Kunt: *Characterization of human visual sensitivity for video imaging applications.* Signal Processing, 67(3):255–269, 1998. 3

[19] Algazi, V Ralph and Norimichi Hiwasa: *Perceptual criteria and design alternatives for low bit rate video coding.* In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 831–835. IEEE, 1993. 3

[20] Pessoa, A, A Falcao, R Nishihara, A Silva, and R Lotufo: *Video quality assessment using objective parameters based on image segmentation.* SMPTE journal, 108(12):865–872, 1999. 3

[21] Wolf, Stephen and Margaret H Pinson: *Spatial-temporal distortion metric for in-service quality monitoring of any digital video system.* In *Photonics East'99*, pages 266–277. International Society for Optics and Photonics, 1999. 3

[22] Winkler, Stefan: *Issues in vision modeling for perceptual video quality assessment.* Signal Processing, 78(2):231–252, 1999. 3

[23] Wang, Zhou, Alan C Bovik, and Ligang Lu: *Why is image quality assessment so difficult?* In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 4, pages IV–3313. IEEE, 2002.

[24] Lu, Ligang, Zhou Wang, Alan C Bovik, and Jack Kouloheris: *Full-reference video quality assessment considering structural distortion and no-reference quality evaluation of mpeg video.* In *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*, volume 1, pages 61–64. IEEE, 2002. 3

[25] Webster, Arthur A, Coleen T Jones, Margaret H Pinson, Stephen D Voran, and Stephen Wolf: *Objective video quality assessment system based on human perception.* In *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, pages 15–26. International Society for Optics and Photonics, 1993. 4

[26] Wang, Zhou and Alan C Bovik: *Reduced-and no-reference image quality assessment.* Signal Processing Magazine, IEEE, 28(6):29–40, 2011. 4, 5

[27] Bhateja, Vikrant, Aastha Srivastava, and Aseem Kalsi: *Fast ssim index for color images employing reduced-reference evaluation.* In *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013*, pages 451–458. Springer, 2014. 4

[28] Ma, Lin, Songnan Li, and King Ngi Ngan: *Reduced-reference image quality assessment in reorganized dct domain.* Signal Processing: Image Communication, 28(8):884–902, 2013. 4

[29] Wolf, Stephen and Margaret H Pinson: *Low bandwidth reduced reference video quality monitoring system.* In *First International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, pages 23–25, 2005. 4

[30] Gunawan, Irwan P and Mohammed Ghanbari: *Reduced-reference video quality assessment using discriminative local harmonic strength with motion consideration.* Circuits and Systems for Video Technology, IEEE Transactions on, 18(1):71–83, 2008. 4

[31] Le Callet, Patrick, Christian Viard-Gaudin, Stéphane Péchard, and Emilie Caillault: *No reference and reduced reference video quality metrics for end to end qos monitoring.* IEICE transactions on communications, 89(2):289–296, 2006. 4

[32] Soundararajan, Ravi and Alan C Bovik: *Video quality assessment by reduced reference spatio-temporal entropic differencing.* Circuits and Systems for Video Technology, IEEE Transactions on, 23(4):684–694, 2013. 4, 64

[33] Li, Qingling, Jing Yang, Liang He, and Shaofen Fan: *Reduced-reference video quality assessment based on bpneural network model for packet networks.* Energy Procedia, 13:8056–8062, 2011. 4

[34] Caviedes, Jorge and Joel Jung: *No-reference metric for a video quality control loop.* In *Proc. Int. Conf. on Information Systems, Analysis and Synthesis*, volume 13, 2001. 4

[35] Wang, Zhou, Alan C Bovik, and BL Evan: *Blind measurement of blocking artifacts in images.* In *Image Processing, 2000. Proceedings. 2000 International Conference on*, volume 3, pages 981–984. Ieee, 2000. 4

[36] Gastaldo, Paolo, Stefano Rovetta, and Rodolfo Zunino: *Objective assessment of mpeg-video quality: a neural-network approach.* In *Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on*, volume 2, pages 1432–1437. IEEE, 2001. 4

[37] Naccari, Matteo, Marco Tagliasacchi, and Stefano Tubaro: *No-reference video quality monitoring for h. 264/avc coded video.* Multimedia, IEEE Transactions on, 11(5):932–946, 2009. 4

[38] Farias, Mylene CQ and Sanjit K Mitra: *No-reference video quality metric based on artifact measurements.* In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 3, pages III–141. IEEE, 2005. 4

[39] Keimel, Christian, Manuel Klimpke, Julian Habigt, and Klaus Diepold: *No-reference video quality metric for hdtv based on h. 264/avc bitstream features.* In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 3325–3328. IEEE, 2011. 4

[40] Izumi, Kiyotaka, Kei Kawamura, Tomonobu Yoshino, and Sei Naito: *No reference video quality assessment based on parametric analysis of hevc bitstream.* In *Quality of Multimedia Experience (QoMEX), 2014 Sixth International Workshop on*, pages 49–50. IEEE, 2014. 4

[41] Jia, Lixiu, Xuefei Zhong, Yan Tu, and Wenjuan Niu: *A no-reference video quality assessment metric based on roi.* In *IS&T/SPIE Electronic Imaging*, pages 93960Z–93960Z. International Society for Optics and Photonics, 2015. 4

[42] Zhu, Kongfeng, Chengqing Li, Vijayan Asari, and Dietmar Saupe: *No-reference video quality assessment based on artifact measurement and statistical analysis.* Circuits and Systems for Video Technology, IEEE Transactions on, 25(4):533–546, 2015. 4

[43] Battisti, Federica, Marco Carli, Yiwei Liu, Alessandro Neri, and Pradip Paudyal: *Distortion-based no-reference quality metric for video transmission over ip.* In *Signals, Circuits and Systems (ISSCS), 2015 International Symposium on*, pages 1–4. IEEE, 2015. 4

[44] Eskicioglu, Ahmet M and Paul S Fisher: *Image quality measures and their performance.* Communications, IEEE Transactions on, 43(12):2959–2965, 1995. 4

[45] Group, Video Quality Experts *et al.*: *Final report from the video quality experts group on the validation of objective models of video quality assessment, phase ii (fr_tv2).* ftp://ftp. its. bldrdoc. gov/dist/ituvidq/Boulder_VQEG_jan_04/VQEG_PhaseII_FRTV_Final_Report_SG9060E. doc, 2003, 2003. 4

[46] Hemami, Sheila S and Amy R Reibman: *No-reference image and video quality estimation: Applications and human-motivated design.* Signal processing: Image communication, 25(7):469–481, 2010. 4

[47] Chen, Guan Hao, Chun Ling Yang, Lai Man Po, and Sheng Li Xie: *Edge-based structural similarity for image quality assessment.* In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 2, pages II–II. IEEE, 2006. 4

[48] Li, Qiaohong, Weisi Lin, and Yuming Fang: *Bsd: Blind image quality assessment based on structural degradation.* Neurocomputing, 236:93–103, 2017. 4

[49] Wu, Qingbo, Hongliang Li, and King N Ngan: *Gip: Generic image prior for no reference image quality assessment.* In *Pacific Rim Conference on Multimedia*, pages 600–608. Springer, 2016. 4

[50] Sheikh, Hamid R, Alan C Bovik, and Lawrence Cormack: *No-reference quality assessment using natural scene statistics: Jpeg2000.* IEEE Transactions on Image Processing, 14(11):1918–1927, 2005. 4

[51] Moorthy, Anush Krishna and Alan Conrad Bovik: *Blind image quality assessment: From natural scene statistics to perceptual quality.* IEEE transactions on Image Processing, 20(12):3350–3364, 2011. 4, 6

[52] Fang, Yuming, Kede Ma, Zhou Wang, Weisi Lin, Zhijun Fang, and Guangtao Zhai: *No-reference quality assessment of contrast-distorted images based on natural scene statistics.* IEEE Signal Processing Letters, 22(7):838–842, 2015. 4

[53] Zhang, Yazhong, Jinjian Wu, Xuemei Xie, Leida Li, and Guangming Shi: *Blind image quality assessment with improved natural scene statistics model.* Digital Signal Processing, 57:56–65, 2016. 4

[54] Appina, Balasubramanyam, Sameeulla Khan, and Sumohana S Channappayya: *No-reference stereoscopic image quality assessment using natural scene statistics.* Signal Processing: Image Communication, 43:1–14, 2016. 4

[55] Liu, Lixiong, Hongping Dong, Hua Huang, and Alan C Bovik: *No-reference image quality assessment in curvelet domain.* Signal Processing: Image Communication, 29(4):494–505, 2014. 4, 37, 39

[56] Liu, Lixiong, Bao Liu, Hua Huang, and Alan Conrad Bovik: *No-reference image quality assessment based on spatial and spectral entropies.* Signal Processing: Image Communication, 29(8):856–863, 2014. 4, 37, 39

[57] Hadizadeh, Hadi and Ivan V Bajić: *No-reference image quality assessment using statistical wavelet-packet features.* Pattern Recognition Letters, 80:144–149, 2016. 4

[58] Zhao, Maozheng, Qin Tu, Yanping Lu, Yongyu Chang, Bo Yang, *et al.*: *No-reference image quality assessment based on phase congruency and spectral entropies.* In *Picture Coding Symposium (PCS), 2015*, pages 302–306. IEEE, 2015. 4

[59] Liu, Bao, Li Xiong Liu, Hong Ping Dong, and Yong Gang Lin: *Blind Image Quality Assessment Based on Mutual Information*, pages 127–136. Springer Singapore, Singapore, 2016, ISBN 978-981-10-0740-8. `http://dx.doi.org/10.1007/978-981-10-0740-8_15`. 4

[60] Ye, Peng, Jayant Kumar, Le Kang, and David Doermann: *Unsupervised feature learning framework for no-reference image quality assessment.* In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1098–1105. IEEE, 2012. 4, 5, 6, 37, 39

[61] Ye, Peng and David Doermann: *No-reference image quality assessment using visual codebooks*. IEEE Transactions on Image Processing, 21(7):3129–3138, 2012. 4, 5, 6

[62] Xu, Jingtao, Peng Ye, Qiaohong Li, Yong Liu, and David Doermann: *No-reference document image quality assessment based on high order image statistics*. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3289–3293. IEEE, 2016. 4, 5

[63] Xu, Jingtao, Qiaohong Li, Peng Ye, Haiqing Du, and Yong Liu: *Local feature aggregation for blind image quality assessment*. In *Visual Communications and Image Processing (VCIP), 2015*, pages 1–4. IEEE, 2015. 4

[64] Farias, Mylène CQ and Sanjit K Mitra: *Perceptual contributions of blocky, blurry, noisy, and ringing synthetic artifacts to overall annoyance*. Journal of Electronic Imaging, 21(4):043013–043013, 2012. 4, 6

[65] Zhu, Kongfeng, Keigo Hirakawa, Vijayan Asari, and Dietmar Saupe: *A no-reference video quality assessment based on laplacian pyramids*. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 49–53. IEEE, 2013. 4

[66] Martinez, Helard Becerra and Mylène CQ Farias: *Full-reference audio-visual video quality metric*. Journal of Electronic Imaging, 23(6):061108–061108, 2014. 4

[67] Silva, Alexandre F, Mylene CQ Farias, and Judith A Redi: *Annoyance models for videos with spatio-temporal artifacts*. In *Quality of Multimedia Experience (QoMEX), 2016 Eighth International Conference on*, pages 1–6. IEEE, 2016. 4

[68] Moorthy, Anush Krishna and Alan Conrad Bovik: *Visual importance pooling for image quality assessment*. IEEE journal of selected topics in signal processing, 3(2):193–201, 2009. 5

[69] Narwaria, Manish, Weisi Lin, and Anmin Liu: *Low-complexity video quality assessment using temporal quality variations*. IEEE Transactions on Multimedia, 14(3):525–535, 2012. 5

[70] Charrier, Christophe, Abdelhakim Saadane, and Christine Fernandez-Maloigne: *Dimension reduction-based attributes selection in no-reference learning-based image quality algorithms*. Electronic Imaging, 2017(12):15–20, 2017. 5

[71] Zhu, Kongfeng, Marcus Barkowsky, Minmin Shen, Patrick Le Callet, and Dietmar Saupe: *Optimizing feature pooling and prediction models of vqa algorithms*. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 541–545. IEEE, 2014. 5

[72] Seshadrinathan, Kalpana and Alan C Bovik: *Temporal hysteresis model of time varying subjective video quality*. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 1153–1156. IEEE, 2011. 5

[73] Xu, Jingtao, Peng Ye, Yong Liu, and David Doermann: *No-reference video quality assessment via feature learning*. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 491–495. IEEE, 2014. 5

[74] Wang, Zhou, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli: *Image quality assessment: from error visibility to structural similarity.* IEEE transactions on image processing, 13(4):600–612, 2004. 5, 54, 58, 64

[75] Xue, Wufeng, Lei Zhang, Xuanqin Mou, and Alan C Bovik: *Gradient magnitude similarity deviation: A highly efficient perceptual image quality index.* IEEE Transactions on Image Processing, 23(2):684–695, 2014. 5, 54, 58, 64

[76] Lin, Weisi and C C Jay Kuo: *Perceptual visual quality metrics: A survey.* Journal of Visual Communication and Image Representation, 22(4):297–312, 2011. 5

[77] Farias, Mylene CQ and Sanjit K Mitra: *A methodology for designing no-reference video quality metrics.* In *Fourth International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, pages 1–6, 2009. 5

[78] Vranješ, Mario, Snježana Rimac-Drlje, and Krešimir Grgić: *Review of objective video quality metrics and performance comparison using different databases.* Signal Processing: Image Communication, 28(1):1–19, 2013. 5

[79] Soundararajan, Rajiv and Alan C Bovik: *Survey of information theory in visual quality assessment.* Signal, Image and Video Processing, 7(3):391–401, 2013. 5

[80] Reibman, Amy R, Subhabrata Sen, and Jacobus Van der Merwe: *Analyzing the spatial quality of internet streaming video.* In *International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, 2005. 5

[81] Farias, Mylène CQ, John M Foley, and Sanjit K Mitra: *Detectability and annoyance of synthetic blocky, blurry, noisy, and ringing artifacts.* Signal Processing, IEEE Transactions on, 55(6):2954–2964, 2007. 6

[82] Ramponi, Giovanni and Leonardo Abate: *Robust measurement of the blocking artefact.* In *IS&T/SPIE Electronic Imaging*, pages 72450K–72450K. International Society for Optics and Photonics, 2009. 6

[83] Zhang, Min, Chisako Muramatsu, Xiaoxin Zhou, Tenshi Hara, and Hideaki Fujita: *Blind image quality assessment using the joint statistics of generalized local binary pattern.* Signal Processing Letters, IEEE, 22(2):207–210, 2015. 6

[84] Zhang, Min, Jin Xie, Xiangrong Zhou, and Hideaki Fujita: *No reference image quality assessment based on local binary pattern statistics.* In *Visual Communications and Image Processing (VCIP), 2013*, pages 1–6. IEEE, 2013. 6

[85] Moorthy, Anush Krishna and Alan Conrad Bovik: *A two-step framework for constructing blind image quality indices.* Signal Processing Letters, IEEE, 17(5):513–516, 2010. 6

[86] Saad, Michele, Alan C Bovik, Christophe Charrier, *et al.*: *A dct statistics-based blind image quality index.* Signal Processing Letters, IEEE, 17(6):583–586, 2010. 6

[87] Cheng, Guangquan, Jincai Huang, Zhong Liu, and Cheng Lizhi: *Image quality assessment using natural image statistics in gradient domain.* AEU-International Journal of Electronics and Communications, 65(5):392–397, 2011. 6

[88] Li, Chaofeng, Alan Conrad Bovik, and Xiaojun Wu: *Blind image quality assessment using a general regression neural network.* Neural Networks, IEEE Transactions on, 22(5):793–799, 2011. 6

[89] Tang, Huixuan, Neel Joshi, and Ashish Kapoor: *Learning a blind measure of perceptual image quality.* In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 305–312. IEEE, 2011. 6

[90] Gastaldo, Paolo, Rodolfo Zunino, and Judith Redi: *Supporting visual quality assessment with machine learning.* EURASIP Journal on Image and Video Processing, 2013(1):1–15, 2013. 7

[91] Petrou, Maria and Pedro Garcia Sevilla: *Image Processing: Dealing with Texture.* John Wiley and Sons, 2006. 8

[92] Davies, Emlyn Roy: *Introduction to texture analysis.* In Mirmehdi, Majid, Xianghua Xie, and Jasjit Suri (editors): *Handbook of Texture Analysis*, chapter 1, pages 1–31. Imperial College Press, 2008. 8

[93] Galloway, Mary M: *Texture analysis using gray level run lengths.* Computer graphics and image processing, 4(2):172–179, 1975. 8

[94] Soh, L K and Costas Tsatsoulis: *Texture analysis of sar sea ice imagery using gray level co-occurrence matrices.* IEEE Transactions on geoscience and remote sensing, 37(2):780–795, 1999. 8

[95] He, Dong Chen and Li Wang: *Texture unit, texture spectrum, and texture analysis.* IEEE transactions on Geoscience and Remote Sensing, 28(4):509–512, 1990. 8

[96] Julesz, Bela: *Textons, the elements of texture perception, and their interactions.* Nature, 290(5802):91–97, 1981. 8

[97] Ojala, Timo, Matti Pietikäinen, and Topi Mäenpää: *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24(7):971–987, 2002. 8, 9, 11

[98] Ahonen, Timo, Abdenour Hadid, and Matti Pietikäinen: *Face recognition with local binary patterns.* Computer vision-eccv 2004, pages 469–481, 2004. 8

[99] Yoo, Cheol Hwan, Seung Wook Kim, June Young Jung, and Sung Jea Ko: *High-dimensional feature extraction using bit-plane decomposition of local binary patterns for robust face recognition.* Journal of Visual Communication and Image Representation, 45:11–19, 2017. 8, 68

[100] Hadid, Abdenour, Juha Ylioinas, Messaoud Bengherabi, Mohammad Ghahramani, and Abdelmalik Taleb-Ahmed: *Gender and texture classification: A comparative analysis using 13 variants of local binary patterns.* Pattern Recognition Letters, 68:231–238, 2015. 8, 9, 15

[101] Ahonen, Timo and Matti Pietikäinen: *Image description using joint distribution of filter bank responses.* Pattern Recognition Letters, 30(4):368–376, 2009. 9

[102] Mäenpää, Topi and Matti Pietikäinen: *Texture analysis with local binary patterns.* Handbook of Pattern Recognition and Computer Vision, 3:197–216, 2005. xiii, 9

[103] Brahnam, Sheryl, Lakhmi C. Jain, Alessandra Lumini, and Loris Nanni: *Introduction to local binary patterns: New variants and applications.* In Brahnam, Sheryl, Lakhmi C. Jain, Loris Nanni, and Alessandra Lumini (editors): *Local Binary Patterns*, volume 506 of *Studies in Computational Intelligence*, pages 1–13. Springer, 2013. 9, 12

[104] Pietikäinen, Matti, Timo Ojala, and Zelin Xu: *Rotation-invariant texture classification using feature distributions.* Pattern Recognition, 33(1):43–52, 2000. 11

[105] Ojansivu, Ville and Janne Heikkilä: *Blur insensitive texture classification using local phase quantization.* In *International conference on image and signal processing*, pages 236–243. Springer, 2008. 14

[106] Kannala, Juho and Esa Rahtu: *Bsif: Binarized statistical image features.* In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 1363–1366. IEEE, 2012. 15

[107] Arashloo, Shervin Rahimzadeh and Josef Kittler: *Dynamic texture recognition using multiscale binarized statistical image features.* IEEE Transactions on Multimedia, 16(8):2099–2109, 2014. 17

[108] Raja, Kiran B, Ramachandra Raghavendra, and Christoph Busch: *Binarized statistical features for improved iris and periocular recognition in visible spectrum.* In *Biometrics and Forensics (IWBF), 2014 International Workshop on*, pages 1–6. IEEE, 2014. 17, 68

[109] Arashloo, Shervin Rahimzadeh, Josef Kittler, and William Christmas: *Face spoofing detection based on multiple descriptor fusion using multiscale dynamic binarized statistical image features.* IEEE Transactions on Information Forensics and Security, 10(11):2396–2407, 2015. 17

[110] Raghavendra, R and Christoph Busch: *Robust scheme for iris presentation attack detection using multiscale binarized statistical image features.* IEEE Transactions on Information Forensics and Security, 10(4):703–715, 2015. 17

[111] Mehta, Rakesh and Karen O Egiazarian: *Rotated local binary pattern (rlbp)-rotation invariant texture descriptor.* In *ICPRAM*, pages 497–502, 2013. 17

[112] Mehta, Rakesh and Karen Egiazarian: *Dominant rotated local binary patterns (drlbp) for texture classification.* Pattern Recognition Letters, 71:16–22, 2016. xiii, 18

[113] Kullback, Solomon and Richard A Leibler: *On information and sufficiency.* The annals of mathematical statistics, 22(1):79–86, 1951. 19

[114] Briët, Jop and Peter Harremoës: *Properties of classical and quantum jensen-shannon divergence.* Physical review A, 79(5):052311, 2009. 19

[115] Ye, Nong, Connie M Borror, and Darshit Parmar: *Scalable chi-square distance versus conventional statistical distance for process monitoring with uncorrelated data variables.* Quality and Reliability Engineering International, 19(6):505–515, 2003. 19

[116] Guo, Zhenhua, Lei Zhang, and David Zhang: *A completed modeling of local binary pattern operator for texture classification.* IEEE Transactions on Image Processing, 19(6):1657–1663, 2010. xiii, 19

[117] Yimo Guo, Guoying Zhao and Matti Pietikäinen: *Texture classification using a linear configuration model based descriptor.* In *Proceedings of the British Machine Vision Conference*, pages 119.1–119.10. BMVA Press, 2011, ISBN 1-901725-43-X. http://dx.doi.org/10.5244/C.25.119. 20

[118] Maenpaa, Topi, Matti Pietikainen, and Jaakko Viertola: *Separating color and pattern information for color texture discrimination.* In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 668–671. IEEE, 2002. 22

[119] Mäenpää, Topi: *The local binary pattern approach to texture analysis: extensions and applications.* Oulun yliopisto, 2003. 22

[120] Jain, Amit and Glenn Healey: *A multiscale representation including opponent color features for texture recognition.* IEEE Transactions on Image Processing, 7(1):124–128, 1998. 22

[121] Wolf, Lior, Tal Hassner, and Yaniv Taigman: *Descriptor Based Methods in the Wild.* In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, Marseille, France, October 2008. Erik Learned-Miller and Andras Ferencz and Frédéric Jurie. `https://hal.inria.fr/inria-00326729`. xiv, 24, 25

[122] Chang, Dar Jen, Ahmed H Desoky, Ming Ouyang, and Eric C Rouchka: *Compute pairwise manhattan distance and pearson correlation coefficient of data points with gpu.* In *Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, 2009. SNPD'09. 10th ACIS International Conference on*, pages 501–506. IEEE, 2009. 24

[123] De Maesschalck, Roy, Delphine Jouan-Rimbaud, and Désiré L Massart: *The mahalanobis distance.* Chemometrics and intelligent laboratory systems, 50(1):1–18, 2000. 24

[124] Merigó, José M and Montserrat Casanovas: *A new minkowski distance based on induced aggregation operators.* International Journal of Computational Intelligence Systems, 4(2):123–133, 2011. 24

[125] Freitas, Pedro Garcia, Welington YL Akamine, and Mylène CQ Farias: *Blind image quality assessment using multiscale local binary patterns.* Journal of Imaging Science and Technology, 60(6):60405–1, 2016. 25, 42, 46

[126] Opitz, Michael, Markus Diem, Stefan Fiel, Florian Kleber, and Robert Sablatnig: *End-to-end text recognition using local ternary patterns, mser and deep convolutional nets.* In *Document Analysis Systems (DAS), 2014 11th IAPR International Workshop on*, pages 186–190. IEEE, 2014. 28, 68

[127] Anthimopoulos, Marios, Basilis Gatos, and Ioannis Pratikakis: *Detection of artificial and scene text in images and video frames.* Pattern Analysis and Applications, 16(3):431–446, 2013. 28

[128] Freitas, Pedro Garcia, Welington YL Akamine, and Mylene CQ Farias: *No-reference image quality assessment based on statistics of local ternary pattern.* In *Quality of Multimedia Experience (QoMEX), 2016 Eighth International Conference on*, pages 1–6. IEEE, 2016. 28, 39

[129] Farias, Mylène CQ and Welington YL Akamine: *On performance of image quality metrics enhanced with visual attention computational models.* Electronics letters, 48(11):631–633, 2012. 33

[130] Akamine, Welington YL and Mylène CQ Farias: *Video quality assessment using visual attention computational models.* Journal of Electronic Imaging, 23(6):061107–061107, 2014. 33, 67

[131] Zhang, Jianming and Stan Sclaroff: *Exploiting surroundedness for saliency detection: a boolean map approach.* IEEE transactions on pattern analysis and machine intelligence, 38(5):889–902, 2016. 33

[132] Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay: *Scikit-learn: Machine learning in Python.* Journal of Machine Learning Research, 12:2825–2830, 2011. 37, 39

[133] Liu, Miao, Mingjun Wang, Jun Wang, and Duo Li: *Comparison of random forest, support vector machine and back propagation neural network for electronic tongue data classification: Application to the recognition of orange beverage and chinese vinegar.* Sensors and Actuators B: Chemical, 177:970–980, 2013. 37, 62

[134] Mittal, Anish, Anush Krishna Moorthy, and Alan Conrad Bovik: *No-reference image quality assessment in the spatial domain.* Image Processing, IEEE Transactions on, 21(12):4695–4708, 2012. 37, 39

[135] Rodriguez-Galiano, V, M Sanchez-Castillo, M Chica-Olmo, and M Chica-Rivas: *Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines.* Ore Geology Reviews, 71:804–818, 2015. 37

[136] Sheikh, Hamid R, Zhou Wang, Lawrence Cormack, and Alan C Bovik: *LIVE image quality assessment database release 2.* http://live.ece.utexas.edu/research/quality, 2005. Accessed: 2016-09-30. 39

[137] Larson, Eric C and DM Chandler: *Categorical image quality (CSIQ) database.* Online, http://vision.okstate.edu/csiq, 2010. Accessed: 2016-09-30. 39

[138] Ponomarenko, Nikolay, Lina Jin, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, Jaakko Astola, Benoit Vozel, Kacem Chehdi, Marco Carli, Federica Battisti, *et al.*: *Image database tid2013: Peculiarities, results and perspectives.* Signal Processing: Image Communication, 30:57–77, 2015. 39

[139] Derpanis, Konstantinos GP and Richard Wildes: *Spacetime texture representation and recognition based on a spatiotemporal orientation analysis.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(6):1193–1205, 2012. 54, 61

[140] Sobel, Irvin: *An isotropic 3× 3 image gradient operator.* Machine Vision for three-demensional Sciences, 1990. 55

[141] Fuglede, Bent and Flemming Topsoe: *Jensen-shannon divergence and hilbert space embedding.* In *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, page 31. IEEE, 2004. 57

[142] Joyce, James M: *Kullback-leibler divergence.* In *International Encyclopedia of Statistical Science*, pages 720–722. Springer, 2011. 57

[143] Thomas, Joy A and Thomas M Cover: *Elements of information theory.* John Wiley & Sons, 2006. 57

[144] Wang, Zhou, Eero P Simoncelli, and Alan C Bovik: *Multiscale structural similarity for image quality assessment.* In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, volume 2, pages 1398–1402. IEEE, 2003. 58

[145] Wadhwa, Neal, Michael Rubinstein, Frédo Durand, and William T Freeman: *Riesz pyramids for fast phase-based video magnification.* In *Computational Photography (ICCP), 2014 IEEE International Conference on*, pages 1–10. IEEE, 2014. 60

[146] Simoncelli, Eero P and William T Freeman: *The steerable pyramid: A flexible architecture for multi-scale derivative computation.* In *Image Processing, 1995. Proceedings., International Conference on*, volume 3, pages 444–447. IEEE, 1995. 60

[147] Derpanis, Konstantinos G and Jacob M Gryn: *Three-dimensional nth derivative of gaussian separable steerable filters.* In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 3, pages III–553. IEEE, 2005. 61

[148] *Csiq video quality database.* http://vision.eng.shizuoka.ac.jp/mod/page/view.php?id=24. Accessed: 2017-05-09. 62

[149] *Ivp subjective quality video database.* http://ivp.ee.cuhk.edu.hk/research/database/subjective/. Accessed: 2017-05-09. 63

[150] Lin, Joe Yuchieh, Rui Song, Chi Hao Wu, TsungJung Liu, Haiqiang Wang, and C C Jay Kuo: *Mcl-v: A streaming video quality assessment database.* Journal of Visual Communication and Image Representation, 30:1–9, 2015. 63

[151] Seshadrinathan, Kalpana, Rajiv Soundararajan, Alan Conrad Bovik, and Lawrence K Cormack: *Study of subjective and objective quality assessment of video.* IEEE transactions on Image Processing, 19(6):1427–1441, 2010. 63

[152] Moorthy, Anush Krishna, Lark Kwon Choi, Alan Conrad Bovik, and Gustavo De Veciana: *Video quality assessment on mobile devices: Subjective, behavioral and objective studies.* IEEE Journal of Selected Topics in Signal Processing, 6(6):652–671, 2012. 63

[153] Moorthy, Anush K, Lark K Choi, Gustavo De Veciana, and Alan C Bovik: *Subjective analysis of video quality on mobile devices.* In *Sixth International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM), Scottsdale, Arizona.* Citeseer, 2012. 63

[154] Yan, Peng, Xuanqin Mou, and Wufeng Xue: *Video quality assessment via gradient magnitude similarity deviation of spatial and spatiotemporal slices.* In *SPIE/IS&T Electronic Imaging*, pages 94110M–94110M. International Society for Optics and Photonics, 2015. 63

[155] Vu, Phong V and Damon M Chandler: *Vis3: an algorithm for video quality assessment via analysis of spatial and spatiotemporal slices.* Journal of Electronic Imaging, 23(1):013016–013016, 2014. 64

[156] Ravinder, M and T Venugopal: *Content-based cricket video shot classification using bag-of-visual-features.* In *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, pages 599–606. Springer, 2016. 68

[157] Di Salvo, Roberto and Carmelo Pino: *Image and video processing on cuda: state of the art and future directions.* MACMESE, 11:60–66, 2011. 69

[158] Blume, Holger, Jörg von Livonius, Lisa Rotenberg, Tobias G Noll, Harald Bothe, and Jörg Brakensiek: *Openmp-based parallelization on an mpcore multiprocessor platform–a performance and power analysis.* Journal of Systems Architecture, 54(11):1019–1029, 2008. 69

Appendix A

Awards

- **A1:** Honorable mention award in the main track for the work *"Embedding Color Watermarks into Halftoning Images using Minimum-Distance Binary Patterns"* presented in the 28th Conference on Graphics, Patterns and Images (SIBGRAPI), Salvador, Brazil, on August 26–29, 2015.

- **A2:** Best student paper award in the Image Quality and System Performance track for the work *"Blind Image Quality Assessment Using Multiscale Local Binary Patterns"* presented in the International Symposium on Electroning Imaging, Burlingame, California, on February 29, 2017.

SIBGRAPI 2015

XXVIII Conference on Graphics, Patterns and Images

Honorable Mention

This is to certify that **Pedro Garcia Freitas, Mylene Farias and Aletéia Araújo** received honorable mention award in the **main track** for the work **“Embedding Color Watermarks into Halftoning Images using Minimum-Distance Binary Patterns”** presented in the 28th Conference on Graphics, Patterns and Images (SIBGRAPI), which took place in Salvador, Brazil, on August 26–29, 2015.

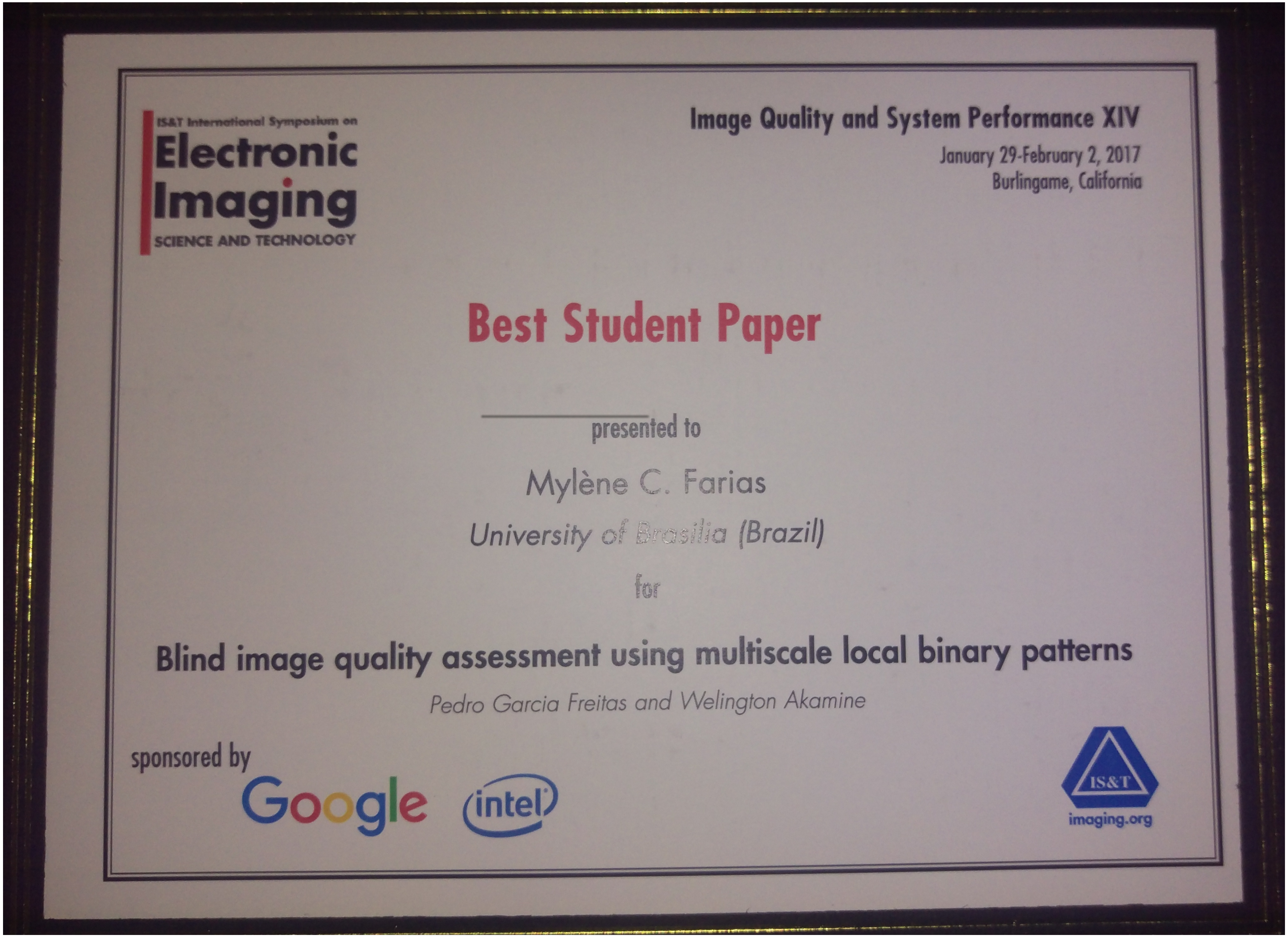
Oliveira

Luciano R. Oliveira
General Chair

Antônio L. Apolinário Jr.
General Chair

Rubisley P. Lemes
General Chair

SBC
Sociedade Brasileira
de Computação



IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Image Quality and System Performance XIV

January 29-February 2, 2017
Burlingame, California

Best Student Paper

presented to

Mylène C. Farias

University of Brasilia (Brazil)

for

Blind image quality assessment using multiscale local binary patterns

Pedro Garcia Freitas and Welington Akamine

sponsored by

Google intel

IS&T
imaging.org

Appendix B

Papers Resulting From This Thesis

B.1 Conference Papers

- **C1:** *A Parallel Framework for Video Super-Resolution* - SIBGRAPI - 2014 - Qualis **B1**

- **C2:** *Tampering Detection of Audio-Visual Content Using Encrypted Watermarks* - SIBGRAPI - 2014 - Qualis **B1**

- **C3:** *Embedding Color Watermarks Into Halftoning Images Using Minimum-Distance Binary Patterns* - SIBGRAPI - 2015 - Qualis **B1**

- **C4:** *Improved performance of inverse halftoning algorithms via coupled dictionaries* - ICME - 2015 - Qualis **A1**

- **C5:** *Video quality ruler: A new experimental methodology for assessing video quality* - QoMEX - 2015 - Qualis not included

- **C6:** *No-Reference Image Quality Assessment Using Texture Information Banks* - BRACIS - 2016 - Qualis not included

- **C7:** *No-reference image quality assessment based on statistics of Local Ternary Pattern* - QoMEX - 2016 - Qualis not included

- **C8:** *Blind Image Quality Assessment Using Local Variant Patterns* - BRACIS - 2017 - Qualis not included

- **C9:** *On the Performance of Visual Semantics for Improving Texture-based Blind Image Quality Assessment* - SIBGRAPI - 2017 - Qualis **B1**

B.2 Journal Papers

- **J1:** *A Parallel Framework for Video Super-resolution* - Electronic Letters on Computer Vision and Image Analysis (ELCVIA) - 2014 - Qualis **B2**

- **J2:** *Detecting tampering in audio-visual content using QIM watermarking* - Information Sciences - 2016 - Qualis **A1**

- **J3:** *Enhancing Inverse Halftoning via Coupled Dictionary Training* - Signal Processing: Image Communication - 2016 - Qualis **A1**

- **J4:** *Secure self-recovery watermarking scheme for error concealment and tampering detection* - Journal of The Brazilian Computer Society - 2016 - Qualis **B1**

- **J5:** *Hiding color watermarks in halftone images using maximum-similarity binary patterns* - Signal Processing: Image Communication - 2016 - Qualis **A1**

- **J6:** *Blind Image Quality Assessment Using Multiscale Local Binary Patterns* - Journal of Imaging Science and Technology - 2017 - Qualis **B4**

- **J7:** *Fast Video Artistic Transfer via Motion Compensation* - The International journal of Multimedia & Its Applications - 2017 - Qualis not included

B.3 First Page of Papers

2014 27th SIBGRAPI Conference on Graphics, Patterns and Images

A Parallel Framework for Video Super-Resolution

Pedro Garcia Freitas
Department of Computer Science
University of Brasília (UnB)
Brasília, Brazil
http://www.sawp.com.br

Mylène C.Q. Farias
Department of Electrical Engineering
University of Brasília (UnB)
Brasília, Brazil
http://www.pgea.unb.br/ mylene/

Aletéia P. F. de Araújo
Department of Computer Science
University of Brasília (UnB)
Brasília, Brazil

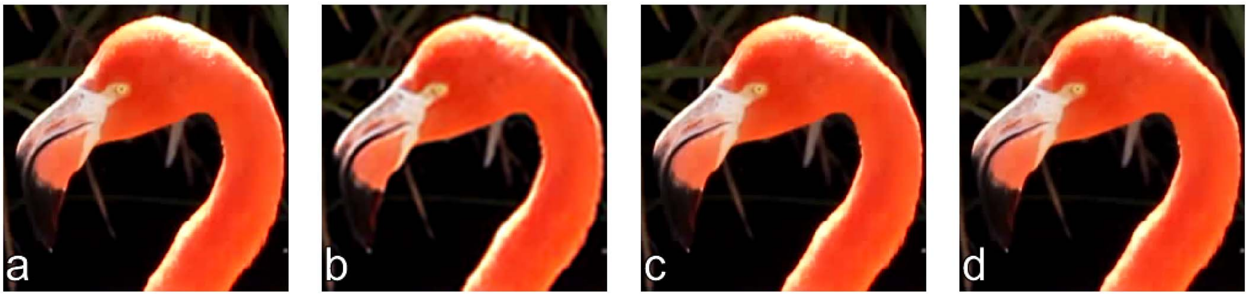


Fig. 1. Sample results of the proposed method: (a) reference frame, (b) enlarged using bicubic interpolation, (c) enlarged using SR for whole frame, and (d) enlarged using proposed method.

***Abstract*—In this paper, we propose a framework for acquiring super-resolution videos from low-resolution originals. Given that super-resolution conversion algorithms require a large amount of data processing, the proposed framework uses a set of strategies to improve performance and computational efficiency. The strategies consists of a combination of data simplification and parallel processing techniques. The simplification strategies are used to decrease the amount of data to process and, consequently, the required processing time. The parallel processing techniques are designed so that major modifications of the super-resolution algorithms are not required. The framework is fast and makes the video resolution increase timely.**

***Keywords*-Video super-resolution; parallel computing; high performance computing network;**

I. Introduction

The techniques used to increase the spatial resolution of videos consist of resizing the individual frames to higher spatial dimensions, using digital signal processing algorithms [1]. The most simple way of obtaining this magnification is using interpolation techniques. However, interpolation techniques are limited and introduce spatial and temporal artifacts in the magnified videos. The most common artifacts in interpolated images are blurring, aliasing, and edge halo.

To overcome these limitations, several techniques have been proposed to increase spatial resolution of images and videos. One of the first works on this topic was the technique proposed by Tsai and Huang [2], which considers the problem of increasing the spatial resolution of still images using a sequence of low-resolution images. Around 1990, the term "super-resolution" was incorporated in the literature by Irani and Peleg [3]. Since then, many super-resolution algorithms have been proposed, using different approaches [4], [5], [6], [7].

Some of the proposed super-resolution methods operate in frequency domain. These frequency-based methods use the shift and scale properties of the transform to obtain a higher resolution image. Tekalp *et al.* [4] use a multivariate statistical technique known as correspondence analysis [8]. Still exploring the frequency domain, Kim *et al.* [5] propose an approach that uses a recursive least-squares method.

There is also a class of super-resolution methods that operate on the spatial (pixel) domain. In this class of methods, different approaches may be used, such as reconstruction of non-uniformly spaced samples [9], backprojection [10], [7], [11], and stochastic models [12], [6], [13].

Image super-resolution algorithms are, generally, computationally expensive, since they involve performing several operations over a large amount of data. When these algorithms are adapted for video signals, the computational complexity is further increased. Therefore, approaches that reduce the processing time of video super-resolution algorithms are necessary.

In this paper, we use a selective processing strategy to reduce the processing time of super-resolution algorithms. In other words, the approach selects a subset of the video frame pixels to be processed by the super-resolution algorithms. Combined with this selective processing technique, we propose a specific parallel data processing approach. The combination of these two approaches allows us to build an efficient strategy to increase video resolution. This proposed strategy is described as a framework because the selective

1530-1834/14 $31.00 © 2014 IEEE
DOI 10.1109/SIBGRAPI.2014.15

2014 27th SIBGRAPI Conference on Graphics, Patterns and Images

Tampering Detection of Audio-Visual Content using Encrypted Watermarks

Ronaldo Rigoni*, Pedro Garcia Freitas*, and Mylène C.Q. Farias*+
*Department of Computer Science,
+Department of Electrical Engineering,
University of Brasília (UnB),
Campus Universitário Darcy Ribeiro, 70919-970 Brasília, DF - Brazil
Emails: rrigoni@gmail.com, sawp@sawp.com.br, mylene@ieee.org

***Abstract*—In this paper, we present a framework for detecting tampered information in digital videos. Using the proposed technique is possible to detect several types of tampering with a pixel granularity. The framework uses a combination of temporal and spatial watermarks that do not decrease the perceived quality of the host videos. We use a modified version of Quantization Index Modulation (QIM) algorithm to store the watermarks. Since QIM is a fragile watermarking scheme, it is possible to detect local, global, and temporal tampers and also estimate the attack type. The framework is fast, robust, and accurate.**

***Keywords*-Quantization Index Modulation, Watermarking, Video Tampering Detection**

I. Introduction

The flexibility of digital images and videos is both a blessing and a curse. Digital technologies make it possible to create high quality pictures, animations, games, and special effects with an amazing realism. Digital pictures (images and videos) can be enhanced, compressed, transmitted, translated across different standards, and displayed in a variety of devices. Then, because of the significant advances in compression and transmission techniques, it is possible to deliver high quality visual content to the end user in many different ways. As a consequence, a variety of delivery services have been created in the last years, such as direct TV broadcast satellite, digital broadcast television, and IP-based video streaming.

A very important concern for video distribution applications is content protection, which involves tampering detection [1]. Several softwares are currently available for video processing, making it easy to alter (tamper) visual digital content without leaving any clear sign of these modifications. These softwares allow unauthorized users to change and illegally distribute digital video content, causing monetary and personal losses [2], [3]. As a consequence, automatic methods for checking the authenticity and integrity of digital images and videos are, undoubtedly, very important.

Several techniques have been proposed with the goal of detecting tampering of digital content [1]. These techniques can be divided in approaches that do not require the original (no-reference) and approaches that do require the reference (full reference). Since in most transmission applications the original is not available, no reference approaches are the most adequate ones for these applications.

Most of the no-reference tampering detection techniques are specialized in detecting a single type of tamper [4], [5]. Because of this limitation, it is not possible to create a software that easily detects multiple types of tamper attacks. A compromise between the no-reference and full-reference is the reduced-reference approach. This approach embeds an "invisible" information (mark) into the content using a watermarking technique. To verify if the original content was tampered, the embedded information is extracted and its integrity is verified. In this approach, the *fragility* of the embedded mark is a key element that determines the amount of tampering which the algorithm is able to detect.

There are several reduced-reference tampering detection algorithms [6], [7], [8], [9]. The work of Amerini *et al.* is based on watermarking techniques [6]. Hou *et al.* [10] proposed a spatial tampering detection technique that consists of storing a verification bit in DCT 4×4 blocks. The method is robust and causes low degradation, but it can only detect tamper in 4×4 blocks (spatial resolution) and it does not detect temporal attacks. On the other hand, Lin *et al.* [7] uses spatial and temporal redundancy to detect tampers in videos and static image sequences. Their method inserts a mark in high correlation blocks, making it robust to geometric attacks, such as rotation and scaling. Subramanyam and Emmanuel [8] and Wang and Farid [9] also propose methods for temporal tampering detection.

In this paper, we present a framework with the goal of protecting and detecting tampered information in digital videos. The proposed framework is based on a reduced-reference watermarking technique that is able to detect temporal, local, and global attacks. The algorithm is divided in two main stages. The first stage consists of a *tampering protector* that generates a mark and embeds it into the host content. The second stage is a *tampering detector* that detects tampers using the extracted mark.

The paper is divided as follows. In Section II, we review the types of tampers that the framework is able to detect. In Section III, we present the suite of techniques used to generate the mark and embed it. In Section IV, we describe

1530-1834/14 $31.00 © 2014 IEEE
DOI 10.1109/SIBGRAPI.2014.50

2015 28th SIBGRAPI Conference on Graphics, Patterns and Images

Embedding Color Watermarks into Halftoning Images using Minimum-Distance Binary Patterns

Pedro Garcia Freitas*, Mylène C.Q. Farias† and Aletéia P. F. de Araújo‡
*‡Department of Computer Science,
†Department of Electrical Engineering,
University of Brasília, Brazil
Email: *sawp@sawp.com.br, †mylene@ieee.org, ‡aleteia@cic.unb.br

***Abstract*—This paper presents a halftoning-based watermarking method. This method enables the embedding of a color image into a binary black-and-white halftone, while maintaining the image quality. The proposed technique is capable of embedding watermarks of three color channels into a binary halftone. To achieve high quality halftones, the method maps colors to halftone channels with homogeneous dot patterns which in turn use different binary texture orientations to carry the watermark. They are obtained by solving a minimization problem in which the objective function is the binary distance between the original binary halftone and the available patterns. To restore the color information, we scan the printed halftone image and compute the inverse information (considering the dot pattern). Using the mapped information, we restore the original color channels from the halftone images using a high-quality inverse halftoning algorithm. Experimental results show that the method produces restorations with a superior quality than other methods found in the literature and increases the embedding capacity.**

***Keywords*-Embedding, Halftone, Color Restoration, Watermark, Inverse Halftoning.**

I. Introduction

Transmitting side information using printed media is a challenge due to the distortions introduced by the print-and-scan (PS) process [1]. Some of these distortions occur because the displayed digital color may differ from its printed representation. One of the causes for these differences is the fact that the digital image is converted to a halftone representation before being printed [2]. This halftone representation is generated from a mathematical model that produces colors using a combination of colored dot patterns [3]. The halftone images are perceived as continuous tone images when viewed from a distance due to a low-pass property of the Human Visual System (HVS). Many different halftoning methods have been developed, including Direct Binary Search (DBS) [4, 5], Ordered Dithering (OD) [6, 7], Error Diffusion (ED) [8]–[10], and Dot Diffusion (DD) [11]–[13]. Although there is a great diversity of image halftoning methods, most of them modify the coding information of the printing process.

The scanning process performs the inverse task of the printing process. Scanner devices read the printed halftone and restore a multi-level image via an inverse halftoning algorithm [14, 15]. Although the inverse halftoning algorithm recovers the distinct intensity levels of the original image, the reconstructed image may present distortions like noise [16] and blur [17]. Therefore, PS processes make hardcopy watermarking more challenging than digital watermarking.

Many works in the literature address the document hardcopy problem by trying to keep the hidden information on a PS channel more robust. Most prior work on image data hiding target color and grayscale images with a wide range of intensity values [18]–[20]. These methods differ from each other in terms of efficiency, capacity, and robustness. Brassil *et al.* [21] propose authentication methods based on shift coding. To increase the robustness, their methods require the use of uniformly spaced centroids, which are often difficult to obtain. Tan *et al.* [22] extended these methods using a directional modulation technique for watermarking of Chinese text images. More recently, other methods were proposed for specific applications [23]–[26].

Among the available methods, those that embed information into binary images are very promising because the pixel binarization is the last step process of the printing process. When the scanner reads the paper, the data is first collected as a binarized halftone which increases the robustness of PS process. It is worth pointing out that, since binary images have less capacity to hide information, embedding data in binary images is more difficult than in color or grayscale images [27]. Although more difficult, the demand for this kind of technique is increasing and several binary-image watermarking techniques have been developed [28]–[32]. However, as stated by Hou *et al.* [27], these methods have several limitations that include a limited data capacity and the presence of noticeable artifacts.

Some approaches have been proposed to increase the embedding capacity of binary-images embedding. Pan *et al.* propose a low-capacity watermarking scheme for halftone image authentication, exploiting an image hash as a fragile watermark [33]. Guo and Liu [34] developed a higher capacity watermarking technique that uses a block truncation code. Son and Choo [35] proposed a watermarking method for clustered halftone dots in which the embedded binary data is recovered using dictionary learning. Guo *et al.* [36] proposed a halftoning-based approach capable of embedding watermarks using direct binary search to encode the binary data. Guo and Liu [37] propose a method for embedding a

1530-1834/15 $31.00 © 2015 IEEE
DOI 10.1109/SIBGRAPI.2015.35

IMPROVED PERFORMANCE OF INVERSE HALFTONING ALGORITHMS VIA COUPLED DICTIONARIES

Pedro Garcia Freitas**, Mylène C.Q. Farias*$^{*+}$*, and Aletéia P. F. de Araújo**

*Department of Computer Science, $^{+}$Department of Electrical Engineering
University of Brasília, Brasília, Brazil

ABSTRACT

Inverse halftoning techniques are known to introduce visible distortions (typically, blurring or noise) into the reconstructed image. To reduce the severity of these distortions, we propose a novel training approach for inverse halftoning algorithms. The proposed technique uses a coupled dictionary (CD) to match distorted and original images via a sparse representation.This technique enforces similarities of sparse representations between distorted and non-distorted images. Results show that the proposed technique can improve the performance of different inverse halftone approaches. Images reconstructed with the proposed approach have a higher quality, showing less blur, noise, and chromatic aberrations.

Index Terms— Inverse Halftone, Dictionary Training, Image Restoration, Sparse Coding, Sparse Modeling.

1. INTRODUCTION

Most printed materials are produced using halftoning techniques. Halftoning is the technique of converting continuous-tone images into images with a limited number of color levels. The technique generates images that, although having a limited number of levels, convey the illusion of having a higher number of levels. Inverse halftoning is the process of generating continuous-tone images from their halftoned versions. The reconstruction of scanned images is an application of inverse halftoning that is very important for the publishing industry [1]. Other applications include the protection of digital documents against piracy [2], authentication of video content [3], compression of images [4], and error concealment for images and videos [5]. In all these applications, the quality of the reconstructed image using the inverse halftoning algorithm is crucial.

Generally, algorithms proposed for inverse halftoning focus on the restoration of a specific kind of halftoning algorithm. In the case of dithering, there are several techniques to generate the patterns that create the illusion of a continuous-tone image. The corresponding inverse halftoning algorithms use the appropriate pattern to optimize image reconstruction. For example, Saika *et. al* [6] and Freitas *et al.* [7] use stochastic models to restore continuous-tone image from Ordered Dithering (OD) halftones. But, when the halftoning technique is an error diffusion dithering technique, these approaches do not produce good results. This is a problem since error diffusion techniques [8] have a better performance than ordered dithering techniques [9, 10]

In this paper, we present a method to enhance the visual quality of images reconstructed using inverse halftoning techniques. We treat the inverse halftoned image as a distorted version of the original image and focus on recovering a non-distorted version. Although the technique presented here can be used to obtain the inverse halftone directly, the goal of the proposed technique is to detect and reduce distortions in reconstructed images generated by an inverse halftoning technique.

This paper is organized as follows. In Section 2, we describe the distortions associated with inverse halftoning algorithms. In Section 3, we discuss how to train a pair of dictionaries to match information between the original image and the reconstructed image. Section 4 details our strategy to improve inverse halftoning methods and Section 5 presents its results. Finally, in Section 6, we present our conclusions and discuss future works.

2. INVERSE HALFTONING

In this work, halftoning is the process of generating a binary (2 levels) image I_b from a continuous-tone (255-levels) grayscale image I_g (or a channel of a colored image). Particularly,

$$I_b = \mathbf{H} \cdot I_g, \tag{1}$$

where $\mathbf{H}$ is an operator representing the halftoning process that transforms I_g into the binary image (pixels values equal to '1' or '0'). Although $\mathbf{H}$ can be viewed as a simple operation, there are several ways to model it. As presented by Ulichney [11], different models correspond to different dithering patterns, leading to different ways to correlate the information between I_b and I_g.

Inverse halftoning is the process of reconstructing I_g from I_b. Since to generate I_b a considerable amount of information is discarded, the inverse halftoning algorithm can only generate an approximation of I_g. In other words, the reconstructed

Video Quality Ruler: A New Experimental Methodology for Assessing Video Quality

Pedro Garcia Freitas*, Judith A. Redi†, Mylène C.Q. Farias‡ and Alexandre F. Silva§
*§Department of Computer Science, University of Brasília, Brazil
†Department of Intelligent Systems, Delft University of Technology, Netherlands
‡Department of Electrical Engineering, University of Brasília, Brazil

Abstract—**In this paper, we propose a subjective video quality assessment method called video quality ruler (VQR) that can be employed to determine the perceived quality of video sequences. The described method is an extension of the ISO 20462, which is a method to assess image quality. The VQR method provides an interface with a set of pictures. The subjects assess the video using these pictures as a scale and compare the subjective perceived video quality with their perceived quality. The pictures are calibrated to form a numerical scale in units of just noticeable differences (JNDs), which allows to analyze and compare both subjective video and image stimuli. To evaluate the effectiveness of the proposed method, we compare the VQR method with a well-used single stimulus (SS) method. The results show that proposed method can be used to quantify the overall video quality with higher efficiency and with a less biased results than the SS method.**

I. Introduction

Subjective visual quality assessments are crucial for designing reliable objective quality metrics. Subjective experiments are necessary to (1) observe perceptual and annoyance mechanisms in users when exposed to an impaired stimulus, to be modeled in objective metrics and (2) collecting data (subjective quality scores) to be used as a benchmark to test the accuracy of these metrics [1]. The reliability of subjective quality data is, therefore, a major precondition for the development of effective quality metrics. To collect subjective quality assessments, psychometric experiments are typically performed, often involving a set of participants (subjects) which are asked to judge the quality of a set of stimuli using a rating scale [2].

When concerned with measuring the quality of video material, several subjective video quality assessment methodologies are available [2, 3]. A main characteristic of subjective methodologies relates to the way in which stimuli are presented to the subjects. In Single Stimulus (SS) methodologies, subjects rate the quality of just one video clip (the test video), without having a reference. In Double Stimulus (DS) methodologies, subjects rate the quality or difference in quality between two or more videos presented simultaneously or closely spaced in time. Methodologies also differ with respect to the type of scale on which the stimulus is rated. Rating scales can be discrete or continuous, labeled or unlabeled, or with numbered rating points or categories [4]. Each methodology type has advantages and disadvantages. As stated by Engeldrum [5], it is practically impossible to cover all factors affecting the results of a scaling task and provide specific recommendations for each of them. There are common pitfalls in standardized quality assessment methodologies, such as the dependency of the scores on the range of quality spanned by the test samples [6] and the difficulty os subjects is to give a numerical (or categorical) value for quality [5], that can lead to imprecision in measurements and subject bias [1, 7]. Imprecision manifests itself as wide confidence intervals that cause problems in the discriminability of pairs of stimuli. Therefore, it is preferable to choose an experimental methodology that minimizes inter-subject variability of scores, hence maximizing confidence.

It has been shown that SS methodologies (e.g. ACR) and DS methodologies (e.g. DSIS) [2] yield similar confidence levels in Mean Opinion Scores (MOS). On the other hand, for image quality assessment, it has been shown that the Quality Ruler (QR) [3] methodology has advantages in this respect. The image Quality Ruler method is based on the use of a set of reference images that are evenly distributed along a pre-calibrated quality scale (the Standard Quality Scale - SQS). The task of the subjects is to find the image in the ruler whose quality matches that of the test image. The position of the matching ruler image on the SQS gives the quality score of the test image. The task of the subject is therefore reduced to a visual comparison (subjects decide whether the qualities of the ruler image and the test image match), which is simpler than giving a quality score [8]. As a result, the image Quality Ruler retains the advantages of methodologies purely based on visual comparison (such as Paired comparison [2]), but is less time-consuming since the set of comparisons to be performed per test stimulus is limited to the number of reference images in the ruler. The Quality Ruler makes it possible to estimate the quality of images within a large quality range [9] with higher confidence than SS methodologies [10]. In addition, the method has been shown to be less prone to context effects [11].

Considering these important advantages, we investigated the opportunity to extend the image Quality Ruler for *video* quality assessment. The main challenge to tackle here is how to allow the comparison of a video, which is dynamic, with a set of still images. Comparing pairs of images is straightforward since they are static and no details are missed when moving the focus of attention from one to the other, which cannot be recuperated by focusing back on the first image. This is not necessarily the case for video. Specifically, questions arise whether (1) subjects can match the quality of an image with that of a video and (2) the use of a set of images for comparison distracts the subject's attention from the video. In this paper, we report how we addressed this challenge and implemented a 'Video Quality Ruler' (VQR). To validate the method, we

978-1-4799-8958-4/15/$31.00 ©2015 IEEE

2016 5th Brazilian Conference on Intelligent Systems

No-Reference Image Quality Assessment Using Texture Information Banks

Pedro Garcia Freitas*, Welington Y.L. Akamine[†] and Mylène C.Q. Farias[‡]
*Department of Computer Science, [†‡]Department of Electrical Engineering,
University of Brasília, Brazil
Email: *sawp@sawp.com.br, [†]welingtonylakamine@gmail.com, [‡]mylene@ieee.org

***Abstract*—In this paper, we propose a new no-reference quality assessment method which uses a machine learning technique based on texture analysis. The proposed method compares test images with texture images of a public database. Local Binary Patterns (LBPs) are used as local texture feature descriptors. With a Csiszár-Morimoto divergence measure, the histograms of the LBPs of the test images are compared with the histograms of the LBPs of the database texture images, generating a set of difference measures. These difference measures are used to blindly predict the quality of an image. Experimental results show that the proposed method is fast and has a good quality prediction power, outperforming other no-reference image quality assessment methods.**

***Keywords*—Machine Learning, Computer Vision, Texture Analysis, No-Reference Image Quality Assessment, Texture Information Banks**

I. Introduction

Humans perform many tasks that remain difficult for computers, like for example in pattern recognition tasks. Another example of a task in which humans outperform computers is the quality assessment of a visual content (an image or a video). Quality assessment is becoming increasingly important because of its crucial role in various image processing applications [1], such as compression techniques [2], transmission processes, displays, restoration algorithms [3], or photo enlargement techniques [4, 5].

The most robust method for assessing the quality of images is to use a pool of human observers to evaluate the quality of a given visual content. This process of using humans for assessing the visual quality of images is called subjective quality assessment. In other words, subjective quality assessment methods consist of psychophysical experiments in which human subjects estimate the quality of a series of visual stimuli. Subjective quality assessment methods provide a Mean Opinion Score (MOS) for each visual stimuli, which are the average of the individual scores given by subjects for this stimuli [6]. Psychophysical experiments are expensive, laborious, time-consuming, and, therefore, hard to incorporate into an automatic quality assessment system.

In order to make the process of assessing image quality simpler, many researchers have been developing algorithms that use computers to perform quality assessment tasks. These algorithms are defined as objective image quality assessment (IQA) methods. IQA methods make it possible to implement fast and cheap procedures that can monitor and control the final image quality in several image processing applications. Although a big effort has been dedicated to create efficient algorithms, the development of IQA methods is still a challenging area [1].

Objective image quality assessment methods can be classified in three categories, according to the amount of reference (original) required by the algorithm. Full reference (FR) methods estimate the quality of a test image performing some type of comparison with the reference. Reduced reference (RR) methods use only partial information about the reference image. Since requiring the reference image or even partial reference information is an obstacle for many multimedia applications, the solution is to use no-reference (NR) methods that do not require any information about the reference image.

The development of no-reference image quality assessment (NR-IQA) methods is an even more challenging [7]. Among the the challenges faced by NR-IQA methods, we can cite:

- *Masking models:* The development of accurate masking models are central to determine which image distortions are noticeable and, therefore, which distortions may affect quality.
- *Suprathreshold distortions:* While masking models aim to determine whether distortions are noticeable, they are not suitable for distortions which are beyond the threshold of visibility. For these cases, different perceptual models need to be developed and incorporated into the image quality assessment method.
- *Content effects:* As distortions are superimposed with image content, they can become more or less noticeable depending on the type of visual content. This interaction between the distortion and the content require that IQA methods takes into the consideration the content characteristics.
- *Multiple distortions:* Image processing operations (compression, enhancement, or transmission) can simultaneously insert multiple forms of distortions. Although there are methods for assessing the quality of images subject to a single distortion, the combination of multiple distortions is still an open question.
- *Computational performance:* Although a great effort has been devoted to improve prediction accuracy, state-of-the-art algorithms still present high computational complexity

978-1-5090-3566-3/16 $31.00 © 2016 IEEE
DOI 10.1109/BRACIS.2016.23

No-Reference Image Quality Assessment Based on Statistics of Local Ternary Pattern

Pedro Garcia Freitas*, Welington Y.L. Akamine† and Mylène C.Q. Farias‡
*Department of Computer Science,
†‡Department of Electrical Engineering,
University of Brasília, Brazil
Email: *sawp@sawp.com.br, †welingtonylakamine@gmail.com, ‡mylene@ieee.org

Abstract—**In this paper, we propose a new no-reference image quality assessment (NR-IQA) method that uses a machine learning technique based on Local Ternary Pattern (LTP) descriptors. LTP descriptors are a generalization of Local Binary Pattern (LBP) texture descriptors that provide a significant performance improvement when compared to LBP. More specifically, LTP is less susceptible to noise in uniform regions, but no longer rigidly invariant to gray-level transformation. Due to its insensitivity to noise, LTP descriptors are not able to detect milder image degradation. To tackle this issue, we propose a strategy that uses multiple LTP channels to extract texture information. The prediction algorithm uses the histograms of these LTP channels as features for the training procedure. The proposed method is able to blindly predict image quality, i.e., the method is no-reference (NR). Results show that the proposed method is considerably faster than other state-of-the-art no-reference methods, while maintaining a competitive image quality prediction accuracy.**

I. Introduction

In recent years, multimedia applications, including image and video services, have become widely popular. As a consequence, the interest in objective methods that are able to estimate the perceived quality of a multimedia content has increased considerably, both in academia and industry. However, modeling a user reaction to a multimedia content is still a challenging problem. Among the conceivable user reactions, one of the most important issues concerns the image quality assessment (IQA).

Objective image quality assessment (IQA) methods can be classified in three categories, according to the amount of the reference image required by the algorithm. Full reference (FR) methods estimate the quality of a test image performing some type of comparison with the reference. Reduced reference (RR) methods use only partial information about the reference image to estimate quality. Since requiring the reference image or even partial reference information is an obstacle for many multimedia applications, the solution is to use no-reference (NR) methods that do not require any information about the reference image.

Although a lot has been done in the area of multimedia quality assessment, most of the achievements have been in the development of FR methods and there is, still, much to be done in no-reference image quality assessment (NR-IQA) methods [1, 2]. A popular NR-IQA approach consists of estimating the strength of the most relevant image distortions and, then, predicting image quality using a combination of these distortion measures. This approach is known as distortion-specific (DS) because it requires the knowledge of one or more types of distortions. Among the DS-NR-IQA methods, we can cite the works of Chabard *et al.* [3], Li *et al.* [4], Wang *et al.* [5], and Manap & Shao [6].

978-1-5090-0354-9/16/$31.00 ©2016 IEEE

Methods that do not require a prior knowledge of image distortions are described as non-distortion-specific (NDS). Although more complex, NDS methods are more adequate for diverse multimedia scenarios where several different types of distortions are present. The most common NDS approaches either use the statistics of natural images [7, 8] or machine learning techniques. Among the several NR-IQA methods based on machine learning, we can cite the works of Ye *et al.* [9], Zhang *et al.* [2], and Liu *et al.* [10]. It is worth pointing out that, although machine learning techniques show promising results, they may present limitations in terms of computational complexity and prediction performance.

In this paper, we present a method that tackles the aforementioned limitations. The proposed method is a NDS-NR-IQA method that uses machine learning techniques. For training the machine learning algorithm, the proposed method uses the histograms of the local ternary pattern (LTP) as features [11]. This approach enables to blindly predict the image quality, without making any assumptions about the type of distortions that the image may contain. The main advantages of the proposed method are: (1) a high computational efficiency and (2) a quality prediction performance that is comparable to the performance of state-of-the-art NR-IQA methods.

The paper is organized as follows. Section II summarizes the basic concepts of the use of LTP operators for feature extraction. Section III describes the proposed NR-IQA method. Sections IV and V present the experimental setup and results, respectively. Finally, in Section VI we present our conclusions.

II. Brief Review of Local Ternary Pattern

The Local Ternary Pattern (LTP) operator is a generalization of the Local Binary Pattern (LBP) [12] patchwise texture feature extractor. The LBP operator is formally defined as:

$$LBP_{R,P}(t_c) = \sum_{p=0}^{P-1} S(t_p - t_c)2^p, \tag{1}$$

2017 Brazilian Conference on Intelligent Systems

Blind Image Quality Assessment Using Local Variant Patterns

Pedro Garcia Freitas*, Welington Y.L. Akamine† and Mylène C.Q. Farias‡
*Department of Computer Science, †‡Department of Electrical Engineering,
University of Brasília, Brazil
Email: *sawp@sawp.com.br, †welingtonylakamine@gmail.com, ‡mylene@ieee.org

***Abstract*—This paper introduces a new blind image quality assessment (BIQA) metric using texture analysis. The method adopts two texture operators to select image texture information. The first operator is the Local Binary Pattern (LBP), an effective texture operator that is extensively adopted for texture analysis. The second operator is proposed as an extension of LBP. The proposed operator, the Local Variant Pattern (LVP), extracts local energy information. Energy information is particularly important for BIQA metrics because image distortions modify the energy of the textures. Histograms of the LBP and LVP outputs are used as features in a random forest regression algorithm. The proposed method surpass other state-of-the-art BIQA method, as results demonstrate.**

I. Introduction

Image quality assessment (IQA) methods is one of the most challenging applications of image analysis. This is crucial for several multimedia applications such as image acquisition, transmission, storage, compression, restoration, and enhancement. Because of that, research on IQA methods has drawn significant attention over the last years. This research are generally divided into two main approaches: subjective and objective. The subjective approach is based on human judgments to estimate image quality. Because of the human factor, subjective methodologies are time and labor-intensive, it is cumbersome, and, therefore, unsuitable for real-world applications. For that reason, efforts to develop objective IQA metrics have been increasing every year. Objective methods are computational techniques to provide quality assessment. If the outputs of these methods are well correlated to the human perception of quality, they can be successfully used in multimedia applications.

Objective IQA can be further classified according to the quantity of reference needed to estimate quality. Full reference (FR) methods estimate the quality of a test image by performing a comparison with the reference image. Reduced reference (RR) methods require only partial information about the reference image. Although several FR and RR methods has achieved satisfactorily effective levels of performance, as demonstrated by high correlations with human subjective perception, the scope of them is rather limited, because the original image is unavailable in many scenarios. For these cases, there is a need for no-reference (NR) methods that do not require any information about the reference image. Since the NR methods do not see the reference, they are often called blind image quality assessment (BIQA) methods.

The development of BIQA methods is still a challenging problem [1], [2]. A common approach is to estimate image quality using distortion-specific (DS) procedures. These DS procedures measure the strength of the most relevant image distortions. Among the state-of-the-art DS methods are the work of Hadizadeh & Bajic [3], Bahrami & Kot [4], Golestaneh & Chandler [5], and Li *et al.* [6], [7], [8]. The evident disadvantage of these metrics is their application specific to some extent.

It is beneficial to develop general-purpose (GP) and Non-distortion-specific (NDS) methods that can quantify the perceived quality of a distorted image without previous knowledge of the distortion types. These methods are more advisable for quality assessment in different multimedia scenarios. The idea is to make assumptions about the image characteristics instead of assume characteristics of specific distortions. A consummated example of NDS approach is the use of "natural scene statistics", where statistics of impaired and non-impaired images are compared [11], [12].

In this paper, we propose a NDS-BIQA method based on machine learning and texture analysis. This method is inspired on some prominent ML-based NDS-BIQA methods [13], [14], [15], [16], [9]. Among these methods, Wu *et al.* [17] propose a FR-IQA method that compares the structural similarity between reference and test texture information. Although Wu's method achieve good performance results, it is a FR method and hence presents all aforementioned drawbacks. The method proposed by Ye *et al.* [19] is a ML-NDS-BIQA method and it is trained on a codebook composed of complex Gabor-filter-based features. As a result, this method is computationally costly and present several limitations for real-world implementations. On the other hand, Freitas *et al.* [18] use the Local Ternary Pattern (LTP) operator to generate texture features. Although fast and reliable, this extension of the LBP operator produces poor results for some specific artifacts (e.g. color distortions).

This paper is influenced by the aforementioned methods because the same premise is used. The premise is that impairments degrade textures in the same way they affect quality. As shown by Wang *et al.* [20], different impairments, corresponding to similar mean squared error (MSE) values,

978-1-5386-2407-4/17 $31.00 © 2017 IEEE
DOI 10.1109/BRACIS.2017.16

On the Performance of Visual Semantics for Improving Texture-based Blind Image Quality Assessment

Pedro Garcia Freitas* and Mylène C.Q. Farias‡
*Department of Computer Science, ‡Department of Electrical Engineering,
University of Brasília, Brazil
Email: *sawp@sawp.com.br, ‡mylene@ieee.org

***Abstract*—Blind image quality assessment (BIQA) methods aim to estimate the quality of a given test image without referring to the corresponding reference (original) image. Most BIQA methods use visual sensitivity models, which take into consideration intrinsic image characteristics (e.g. contrast, luminance, and texture) to identify degradations and estimate quality. For example, texture-based BIQA methods are based on the assumption that visual impairments (degradations) alter the characteristics of the image textures and, therefore, their statistics. Although these methods have been are known to provide an acceptable performance, they do not take into account the semantic information of the image. In this paper, we propose a BIQA method that estimates quality using texture characteristics and semantic information. The texture characteristics are obtained using the Opponent Color Local Binary Pattern (OCL) operator. The semantic information is obtained by estimating the probability distribution of the scene characteristics. A random forest regression algorithm is used to map semantic and texture-based features into a quality score. Results obtained testing the proposed BIQA method on several public databases show the method has a good accuracy on quality prediction.**

I. Introduction

Image quality assessment (IQA) is a research area that has achieved a great importance in the last years, mostly due to the exponential growth of the popularity of digital visual information (images). Given this high volume of visual information, the task of accurately assessing the quality of an image has become crucial for several multimedia applications. More specifically, IQA methods are used to estimate the performance of compression algorithms [1], multimedia transmission [2], [3], display technologies, image enhancement and restoration algorithms [4].

Over the past decades, a lot of progress has been made in the area of image quality, with a large number of IQA methods being proposed. IQA methods can be classified into three types, according to the amount of information required to perform the assessment task. Full-reference (FR) methods [5] require the original image and are, usually, more precise. Reduced-reference (RR) methods require only part information (e.g. features) about the original image [6], [7]. Because needing even partial information of the reference image can be a hindrance for several multimedia applications, frequently, the most adequate solution is to use blind image quality assessment (BIQA) methods. BIQA methods [8], [9] blindly estimate the quality of a test image without requiring any information about its reference.

Many BIQA methods have been proposed [8]–[11]. Among the available approaches, methods based on texture analysis in combination with machine learning techniques have been very successful. As an example, we can cite the work of Peng Ye and Doermann [12], which uses local Gabor-filter features to build a visual codebook that is used to estimate quality. Recently, several BIQA methods based on a texture descriptor known as the Local Binary Pattern (LBP) operator [13] have been proposed. State-of-the-art LBP-based BIQA methods include the efforts of Freitas *et al.* [9], [14], Rezaie *et al.* [15], Li *et al.* [11], Zhang *et al.* [10], and Wu *et al.* [16].

Although the aforementioned methods achieve an acceptable prediction accuracy, some issues remain open. As stated by Chandler [17], so far, IQA developments focus on improving the prediction accuracy for popular distortions, such as JPEG, blurring, or noise. There are few methods that perform efficiently for multiple distortions. Therefore, there are very few general purpose BIQA methods. In this paper, we investigate if semantics can improve the lack of generality of BIQA methods.

Most IQA methods assume that the perceived quality depends exclusively on the sensitivity to impairments. In this paper, we study how image semantics can affect quality. Our work is inspired by the subjective study performed by Siahaan *et al.* [18], which demonstrated that visual quality is influenced by the semantic content. Moreover, Farias & Akamine [19] studied how to incorporate visual saliency into IQA methods, obtaining interesting results. Since saliency is an aspect of image semantics [20], we believe that image semantics can indeed be used to improve the accuracy performance of IQA methods.

Differently from Siahaan *et al.* [18], who performed an investigation using subjective experiments, we aim to incorporate semantic features into the design of a BIQA method. More specifically, we use a pre-trained deep convolutional neural network to generate semantic categories of an image. These

Electronic Letters on Computer Vision and Image Analysis 13(2):56-57, 2014

A Parallel Framework for Video Super-resolution

Pedro Garcia Freitas

Department of Computer Science, University of Brasília (UnB), Brasília, Brazil
Advisor: Mylène C.Q. Farias, University of Brasília (UnB), Brasília, Brazil
Date and location of M.Sc. defense: 19 February 2013, University of Brasília

Received 3^{rd} Feb 2014; Accepted 28^{th} May 2014

1 Background

Image super-resolution [1] algorithms are, generally, computationally expensive, since they involve a number of operations over a large data amount. When these algorithms are adapted to magnify video frames, this computational effort is further increased. Therefore, approaches that reduce the processing time of super-resolution algorithms are desirable. With this aim, we propose a strategy to reduce the processing time of super-resolution algorithms that uses a selective processing approach. In other words, the algorithm selects which data is to be processed exclusively by super-resolution algorithms. Combined with the selective processing technique, we also propose an approach to parallel video data processing. The combination of these two approaches allows us to build a framework to increase video resolution.

2 Proposed Framework

The framework consists of two major contributions. The first one we call "simplification" because it classifies the video data into complex or simple regions. Complex regions contain a higher percentage of visually significant information than simple regions. Complex regions are processed by algorithms that produce a better visual quality, which are also more computationally complex. The less expensive algorithms are used in simpler regions. As simpler regions require a smaller resource consumption, a simplification approach increases the number of these regions and decreases the number of complex regions. In this work, the proposed simplifications are significant information selection (SIS), contour-guided processing (CGP), and differential encoding (DC).

The SIS simplification strategy considers that the super-resolution algorithms are applied on the YUV color space. If the video is not encoded using YUV color space, the framework converts it to YUV and the super-resolution algorithm is applied to the luminance channel. Color channels are resized using simple interpolation techniques.

The CGP strategy uses the Canny edge detector [2] to segment the regions with more or less details. The regions containing edges are classified as "regions of interest" (ROI). Then, the super-resolution algorithm is applied only on the ROI. Other regions are resized using interpolation algorithms.

Correspondence to: <sawp@sawp.com.br>

Recommended for acceptance by <Alicia Fornés and Volkmar Friken>
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

Information Sciences 328 (2016) 127–143

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Detecting tampering in audio-visual content using QIM watermarking

Ronaldo Rigoni[a], Pedro Garcia Freitas[a,*], Mylène C.Q. Farias[b]

[a] Department of Computer Science, University of Brasília (UnB), Brasília, Brazil
[b] Department of Electrical Engineering, University of Brasília (UnB), Brasília, Brazil

ARTICLE INFO

Article history:
Received 2 July 2014
Revised 17 June 2015
Accepted 23 August 2015
Available online 28 August 2015

Keywords:
Quantization Index Modulation
Watermarking
Video tampering detection

ABSTRACT

This paper presents a framework for detecting tampered information in digital audio-visual content. The proposed framework uses a combination of temporal and spatial watermarks that do not decrease the quality of host videos. A modified version of the Quantization Index Modulation (QIM) algorithm is used to embed watermarks. The fragility of the QIM watermarking algorithm makes it possible to detect local, global, and temporal tampering attacks with pixel granularity. The technique is also able to identify the type of tampering attack. The framework is fast, robust, and accurate.

1. Introduction

Digital technologies make it possible to create high quality pictures, animations, games, and special effects with an amazing realism. Digital pictures (images and videos) can be enhanced, compressed, transmitted, converted across different formats, and displayed in a variety of devices. Also, given the significant advances in compression and transmission techniques, it is possible to deliver high quality content to the end user. As a consequence, a variety of delivery services have been created in the last few years, such as direct TV broadcast satellite, digital broadcast television, and IP-based video streaming.

This flexibility of the digital content is both a blessing and a curse. Using one of the several currently available video editing softwares, it is possible to alter (tamper) digital content without leaving any clear sign. Unauthorized users can modify and illegally distribute digital content, causing monetary and personal losses [10,12]. Therefore, a requirement for current video distribution applications is content protection, including tampering detection [22]. As a consequence, there is a great demand for automatic methods that analyze the authenticity and integrity of digital images and videos.

Several techniques have been proposed with the goal of detecting tampering in digital visual content [22]. These techniques can be divided in approaches that do not require the original (no-reference) and approaches that do require the original (full-reference). Since in most transmission applications the original is not available, the use of no-reference approaches is necessary. Unfortunately, most currently available no-reference tampering detection techniques identify a few types of tamper attacks. For example, the work of Ng et al. [16] uses higher order statistics to detect copy-and-paste (splicing) attacks, while the work of Peng and Wang detects only motion blur inconsistencies [20]. The work of Amerini et al. [1] uses a methodology based on scale invariant features transform (SIFT) to detect copy-and-paste attacks. The framework proposed by Wang et al. checks illumination angles and possible chromatic aberrations to identify traces of tampering [28]. Their method detects interpolation, double

[*] Corresponding author. at: University of Brasília (UnB), Department of Computer Science, Campus Universitário Darcy Ribeiro, 70910900 Brasília, Brazil, Tel.: +556186033557.

E-mail address: sawp@sawp.com.br (P.G. Freitas).

http://dx.doi.org/10.1016/j.ins.2015.08.040
0020-0255/© 2015 Elsevier Inc. All rights reserved.

Signal Processing: Image Communication 49 (2016) 1–8

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

Enhancing inverse halftoning via coupled dictionary training

Pedro G. Freitas[a,*], Mylène C.Q. Farias[b], Aletéia P.F. Araújo[a]

[a] *Department of Computer Science, University of Brasília (UnB), Brasília, Brazil*
[b] *Department of Electrical Engineering, University of Brasília (UnB), Brasília, Brazil*

A R T I C L E I N F O

Keywords:
Coupled dictionaries
Image restoration
Inverse halftoning
Enhancement
Training

A B S T R A C T

Inverse halftoning is a challenging problem in image processing. Traditionally, this operation is known to introduce visible distortions into reconstructed images. This paper presents a learning-based method that performs a quality enhancement procedure on images reconstructed using inverse halftoning algorithms. The proposed method is implemented using a coupled dictionary learning algorithm, which is based on a patchwise sparse representation. Specifically, the training is performed using image pairs composed by images restored using an inverse halftoning algorithm and their corresponding originals. The learning model, which is based on a sparse representation of these images, is used to construct two dictionaries. One of these dictionaries represents the original images and the other dictionary represents the distorted images. Using these dictionaries, the method generates images with a smaller number of distortions than what is produced by regular inverse halftone algorithms. Experimental results show that images generated by the proposed method have a high quality, with less chromatic aberrations, blur, and white noise distortions.

1. Introduction

Printing a digital image requires a conversion from a digital representation to an analog representation. This process is often linked with digital halftoning, which is the technique of converting continuous-tone images into images with a limited number of tones (known as halftones) [1,2]. The technique generates images that, although having a limited number of levels, convey the illusion of having a higher number of levels. Halftoning techniques can be applied both to grayscale and color images. On the other hand, inverse halftoning is the process of generating a reconstruction (or an approximation) of the original continuous-tone image from their halftoning versions. The inverse halftoning process is an important image restoration problem and is frequently associated with several other multimedia problems, such as content protection using watermarking [3], visual cyptography [4], compression of multimedia content [5], error concealment [6], and image quality assessment [7,8].

Since halftoning techniques discard a considerable amount of information from the original continuous-tone image, distortions are frequently introduced in halftone images. As a consequence, the design of inverse halftoning techniques is challenging and, when compared to the original image, restored images may contain distortions. Most common distortions include color distortions, noise, or blur. Over the years, several inverse halftoning methods have been proposed. Examples include the works of Freitas et al. [9] and Saika et al. [10], who propose inverse halftoning methods that restore continuous-tone images from ordered dithering (OD) halftones. Xiong et al. [11], Kite et al. [12], and Neelamani et al. [13] propose wavelet-based approaches that restore images from halftones generated using error diffusion algorithms [14].

The aforementioned inverse halftoning techniques restore continuous-tone images using the knowledge about the specific halftoning technique used to produce the halftone (i.e. ordered dithering, error diffusion, dot diffusion, etc). However, in recent years, a few inverse halftoning techniques that work for different halftoning techniques have been proposed. One example is the work of Wen et al. [15] that uses a template optimization method (based on an elitist genetic algorithm) to implement a lookup-table inverse halftoning technique. Their method is able to restore Floyd-Steinberg error diffusion, Jarvis-Judice error diffusion, cluster dither, Bayer disperse dither, and dot diffusion halftone images. Another example is the work of Guo et al. [16] that is based on a block truncation code (BTC). Finally, Gopale and Sarode [17] propose a descreening inverse halftoning technique that uses image redundancy and adaptive filtering and does not require information about the halftoning process. Although these methods are state-of-the-art techniques, the continuous-tone images restored with them still present visual distortions, like for example noise and blur.

In this paper, we propose a new technique for effectively enhancing fine textures and details of restored halftone images, by concealing noise and blurring effects. More than yet another inverse halftoning

[*] Corresponding author.
E-mail address: sawp@sawp.com.br (P.G. Freitas).

http://dx.doi.org/10.1016/j.image.2016.09.008
Received 6 May 2016; Received in revised form 27 September 2016; Accepted 27 September 2016
Available online 29 September 2016

Journal of the Brazilian Computer Society

RESEARCH Open Access

Secure self-recovery watermarking scheme for error concealment and tampering detection

Pedro Garcia Freitas*, Ronaldo Rigoni and Mylène C. Q. Farias

Abstract

Background: In this paper, we present a method for protecting and restoring lost or tampered information in images or videos using secure watermarks. The proposed method consists of a combination of techniques that are able to detect image and video manipulations. Contrary to most existing watermarking schemes, the method can identify the exact position of the tampered region. Furthermore, the method is capable of restoring the manipulated information and retrieve the original content. This set of capabilities make it possible to use the proposed method in error concealment and digital tampering applications.

Methods: The proposed method is employed as both an error concealment algorithm and a tampering detection algorithm. The proposed method is divided into two stages. At the encoder side, the method generates a binary version (watermark) of the original picture (image or video frame) using a halftoning technique. Then, a quantization index modulation technique is used to embed this watermark into the protected picture. At the decoder side, after the lost or tampered regions are identified, the original content is recovered by extracting the watermark corresponding to the affected areas. An inverse halftoning algorithm is used to convert the dithered version of the picture into a good-quality multi-level approximation of the original content.

Results: First, we test the method in error concealment applications, using a set of still images and H.264 videos. Then, we test the proposed method for tampering detection and content retrieval applications, again considering both images and videos. We compare the proposed method with several other several state-of-the-art algorithms. The results show that the proposed method is fast, robust, and accurate.

Conclusions: Our results show that we can use a single approach to tackle both error concealment and tampering detection problems. The proposed method provides high levels of security, high detection accuracy, and recovery capability, and it is robust to several types of attacks.

Keywords: Quantization index modulation, Watermarking, Error concealment, Tampering detection

Background

The flexibility of digital images and videos is both a blessing and a curse. Digital technologies make it possible to create high-quality pictures, animations, games, and special effects with an amazing realism. Digital pictures (images and videos) can be enhanced, compressed, transmitted, translated across different standards, and displayed in a variety of devices. Because of the significant advances in compression and transmission techniques, it is possible to deliver high-quality visual content to the end user in many different ways. As a consequence, a variety of delivery services have been created in the last years, such as direct TV broadcast satellite, digital broadcast television, and IP-based video streaming.

In content delivery applications, video and image signals are transmitted in a compressed format [1] and they are divided into packets before transmission. Unfortunately, during the transmission over wired and wireless channels, some packets may be lost or delayed. These transmission losses cause various types of visible degradations that may

*Correspondence: sawp@sawp.com.br
Department of Computer Science, University of Brasília, Campus Universitário Darcy Ribeiro, 70919-970 Brasília, DF, Brazil

Signal Processing: Image Communication 48 (2016) 1–11

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

Hiding color watermarks in halftone images using maximum-similarity binary patterns

Pedro Garcia Freitas [a],*, Mylène C.Q. Farias [b], Aletéia P.F. Araújo [a]

[a] *Department of Computer Science, University of Brasília (UnB), Brasília, Brazil*
[b] *Department of Electrical Engineering, University of Brasília (UnB), Brasília, Brazil*

ARTICLE INFO

Article history:
Received 3 June 2016
Received in revised form
25 August 2016
Accepted 25 August 2016
Available online 26 August 2016

Keywords:
Color embedding
Halftone
Color restoration
Watermarking
Enhancement

ABSTRACT

This paper presents a halftoning-based watermarking method that enables the embedding of a color image into binary black-and-white images. To maintain the quality of halftone images, the method maps watermarks to halftone channels using homogeneous dot patterns. These patterns use a different binary texture arrangement to embed the watermark. To prevent a degradation of the host image, a maximization problem is solved to reduce the associated noise. The objective function of this maximization problem is the binary similarity measure between the original binary halftone and a set of randomly generated patterns. This optimization problem needs to be solved for each dot pattern, resulting in processing overhead and a long running time. To overcome this restriction, parallel computing techniques are used to decrease the processing time. More specifically, the method is tested using a CUDA-based parallel implementation, running on GPUs. The proposed technique produces results with high visual quality and acceptable processing time.

1. Introduction

Printing an image consists of performing a conversion from digital to analog, while scanning an image involves a conversion from analog to digital. These two processes may add several types of distortions to the original content, which include geometric distortions (rotation, scaling, cropping, etc.), color distortions, and noise. These distortions are a consequence of several factors, like, for example, the process of conversion from digital images to halftone representations performed just before printing [1].

The halftone representation is generated using a mathematical model that produces the illusion of colors by using a combination of colored dot patterns [2]. Due to the low-pass property of the Human Visual System (HVS), halftone images are perceived as continuous tone images when viewed from a distance. Many different halftoning methods have been developed over the years, like for example Direct Binary Search (DBS) [3,4], Ordered Dithering (OD) [5,6], Error Diffusion (ED) [7–9], and Dot Diffusion (DD) [10–12]. Although there is a great diversity of image halftoning methods, these methods insert distortions during the quantization process that converts multi-level images (color or grayscale) into binary (halftone) images.

Scanner devices read the printed halftone and restore a multi-level image via an inverse halftoning algorithm [13,14]. Therefore, the scanning process corresponds to the inverse of the printing process. Although the inverse halftoning algorithm is able to recover an approximation of the intensity levels of the original image, the reconstructed image often presents distortions like noise [15] and blurring [16].

Although digital watermarking is a well-established area that mostly targets color and grayscale images (wide range of intensity levels) [17–19], *hardcopy* watermarking is still a challenging area. In particular, distortions introduced by the print-and-scan (PS) process make the task of transmitting data using hardcopy watermarking more difficult [20].

Most works in this area focus on making the embedded information more robust to distortions of the PS channel. Methods differ from each other in terms of efficiency, capacity, and robustness. For example, Brassil et al. [21] have proposed an authentication method that is based on shift coding. To increase robustness, their method requires the use of uniformly spaced centroids, which are often difficult to obtain. Tan et al. [22] extended this method using a directional modulation technique for watermarking Chinese text images. More recently, other methods have been proposed for specific applications [23–26].

Among hardcopy watermarking methods, those that embed information into binary images are particularly interesting because pixel binarization is the last step of the printing process. Also, during the scanning process, the data is first acquired as a halftone. Therefore, restoring the watermark from the halftoned

[*] Corresponding author.
E-mail address: sawp@sawp.com.br (P.G. Freitas).

http://dx.doi.org/10.1016/j.image.2016.08.007
0923-5965/© 2016 Elsevier B.V. All rights reserved.

Journal of Imaging Science and Technology® 60(6): 060405-1–060405-8, 2016.

Blind Image Quality Assessment Using Multiscale Local Binary Patterns

Pedro Garcia Freitas
Department of Computer Science, University of Brasília, Brasília, Brazil

Welington Y. L. Akamine
Department of Electrical Engineering, University of Brasília, Brasília, Brazil

Mylène C. Q. Farias▲
Department of Computer Science, University of Brasília, Brasília, Brazil
Department of Electrical Engineering, University of Brasília, Brasília, Brazil
E-mail: mylene@ieee.org

Abstract. *This article proposes a new no-reference image quality assessment method that is able to blindly predict the quality of an image. The method is based on a machine learning technique that uses texture descriptors. In the proposed method, texture features are computed by decomposing images into texture information using multiscale local binary pattern (MLBP) operators. In particular, the parameters of local binary pattern operators are varied, which generates MLBP operators. The features used for training the prediction algorithm are the histograms of these MLBP channels. The results show that, when compared with other state-of-the-art no-reference methods, the proposed method is competitive in terms of prediction precision and computational complexity. © 2016 Society for Imaging Science and Technology.*
[DOI: 10.2352/J.ImagingSci.Technol.2016.60.6.060405]

INTRODUCTION

Given the high volume of visual media available, the task of assessing the visual quality of a content is becoming increasingly important. In particular, image quality assessment (IQA) methods are often necessary to estimate the performance of compression techniques,[1] transmission processes, displays, enhancement and restoration algorithms,[2] or any type of image processing technique.

There are two types of IQA methods: subjective and objective.[3] Subjective quality assessment methods consist of psychophysical experiments in which human subjects estimate the quality of a series of stimuli.[4] These experiments are expensive, laborious, time-consuming, and, therefore, hard to incorporate into an automatic system. On the other hand, in objective quality assessment methods, computer algorithms substitute psychophysical experiments, making it possible to implement fast and cheap procedures for monitoring and controlling the quality of images.[5]

Objective image quality assessment methods are generally classified into three classes, according to the amount of reference information required by the algorithm.[6] If the full reference (original image) is required to estimate the image quality, the algorithm is classified as a full-reference (FR) method. When only partial information about the reference (e.g., features extracted from the original image) is used, the algorithm is classified as a reduced-reference (RR) method. Since requiring even partial reference information is an impediment for several multimedia applications, in most cases the most suitable solution is to use no-reference (NR) methods, which blindly estimate the quality of a test image without requiring any information about its reference.

Although a lot of work has been carried out in the area of IQA, most efforts have been in the development of FR methods, and there are still many open questions in the area of no-reference image quality assessment (NR-IQA).[7] The development of fast and accurate NR-IQA methods is still a challenging problem, with two popular approaches. The first approach is distortion-specific NR-IQA (DS-NR-IQA), which estimates the perceived quality of an image using specific distortion measures.[8–10] The second approach is non-distortion-specific NR-IQA (NDS-NR-IQA). NDS-NR-IQA methods are generally based on the assumption that natural images cover a small subset of all possible images (including distorted images), and, therefore, a statistical comparison between test images and the subset of natural images can be used to obtain a quality estimate.[11–13]

NDS-NR-IQA methods that perform a statistical comparison between impaired and non-impaired natural images are known as "natural scene statistic" (NSS)-based methods. As mentioned earlier, NSS methods are based on an analysis of the statistical regularities of non-distorted natural images, considering a set of features or artifacts. For example, Saad et al.[13] have successfully used Discrete Cosine Transform (DCT) domain statistics to determine the model parameters that are sensitive to most commonly perceived artifacts. Sheikh et al.[14] have developed an NR-IQA method using

▲ IS&T Member.

Received July 3, 2016; accepted for publication Oct. 21, 2016; published online Dec. 7, 2016. Associate Editor: Yeong-Ho Ha.

1062-3701/2016/60(6)/060405/8/$25.00

FAST VIDEO ARTISTIC TRANSFER VIA MOTION COMPENSATION

Pedro Garcia Freitas[1] and Mylène C.Q. Farias[2]

[1]Department of Computer Science, University of Brasília, Brasília, Brazil
[2]Department of Electrical Engineering, University of Brasília, Brasília, Brazil

ABSTRACT

Techniques for conversion of natural video scenes into drawing-style videos are frequently used to produce animated movies. In the past, the conversion was manually performed, what demanded a lot of time and a high production cost. Recently, with the advancement of computer vision techniques and the development of new deep learning algorithms, 'drawing' can be automatically performed. Nevertheless, current `drawing' algorithms are computationally expensive and require a high processing time. In this letter, we present a simple, but effective `drawing' algorithm that is capable of reducing the processing time.

KEYWORDS

Computer-guided Rotoscopy, Style Transfer, Motion Compensation, Video Processing

1. INTRODUCTION

Rotoscoping is a technique [1] that converts 'natural' video frames into cartoon or artistic animated frames. In the early days, rotoscoping was performed manually, demanding an enormous amount of time and a high number of artists, what naturally increased the cost. In recent years, algorithms that perform an automatic conversion of filmed scenes into artistic scenes have been developed to decrease costs. More recently, machine learning algorithms have been used to perform the conversion of natural images into artistic images. For example, Gatys et al. [2] developed an algorithm based on neural networks, which captures the style of paintings (or art works) and transfers it to natural images. Although effective, their approach uses a deep convolutional network to mimic the artistic style and transfer it to the natural content, what is computationally expensive. To increase computational performance, Johnson et al. [3] proposed an algorithm that uses perceptual loss functions based on high-level features from pre-trained neural networks.

Since Gatys' [2] and Johnson's [3] methods were developed for images, they produce flickering and discontinuities distortions when applied to videos. To reduce these distortions, Ruder et al. [4] developed an algorithm that preserves the smooth frame transitions by using temporal constraints that penalize discontinuities between two consecutive frames. Unfortunately, although this algorithm is able to reduce distortions, it requires a large processing time. In this letter, we propose an algorithm that uses motion compensation to eliminate frame discontinuities in converted artistic videos. The algorithm eliminates the unnecessary processing of redundant information in consecutive frames, what reduces the overall processing time. Visual results are good and comparable to previous works [2, 3, 4].

DOI : 10.5121/ijma.2017.9202