



CENTRO DE INFORMÁTICA

CIÊNCIA DA COMPUTAÇÃO

RAFAEL DE MELO FERREIRA MONTEIRO

Métricas de Nitidez de Imagens Aplicadas em Regiões de Saliência

TRABALHO DE CONCLUSÃO DE CURSO

Recife
5 de Dezembro de 2017

RAFAEL DE MELO FERREIRA MONTEIRO

Métricas de Nitidez de Imagens Aplicadas em Regiões de Saliência

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Tsang Ing Ren

Recife
5 de Dezembro de 2017

Rafael de Melo Ferreira Monteiro

Métricas de Nitidez de Imagens Aplicadas em Regiões de Saliência/ Rafael de Melo Ferreira Monteiro. – Recife, 5 de Dezembro de 2017-

49 p. : il. (algumas color.) ; 30 cm.

Orientador: Tsang Ing Ren

Trabalho de Conclusão de Curso – CENTRO DE INFORMÁTICA
CIÊNCIA DA COMPUTAÇÃO , 5 de Dezembro de 2017.

IMPORTANTE: ESSE É APENAS UM TEXTO DE EXEMPLO DE FICHA CATALOGRÁFICA. VOCÊ DEVERÁ SOLICITAR UMA FICHA CATALOGRÁFICA PARA SEU TRABALHO NA BIBLIOTECA DA SUA INSTITUIÇÃO (OU DEPARTAMENTO).

Agradecimentos

Agradeço à minha família, amigos e ao meu professor orientador, por toda a força, conselhos e incentivos que me deram.

Resumo

A avaliação de qualidade de imagens e vídeos possui um papel fundamental em muitas aplicações. As avaliações subjetivas têm sua aplicação impossibilitada devido ao custo e tempo para obter resultados satisfatórios. Faz-se necessário o uso de métricas objetivas, mas que mantenham boa correlação com valores subjetivos e por isso devem se basear em características e comportamentos do Sistema Visual Humano. Este trabalho propõe combinar métricas objetivas, já estabelecidas na literatura, com a detecção de regiões de maior atenção visual na imagem, regiões essas que tendem a captar a atenção de quem observa. Estudos levam a crer que essas áreas causam um maior impacto na percepção de qualidade. Três métricas diferentes serão comparadas entre a sua aplicação comum e sua aplicação sobre as áreas de saliência. A primeira é uma métrica com referência baseada na similaridade estrutural das imagens. A segunda é uma métrica com referência que avalia as variações de luminância. A terceira é uma métrica sem referência que avalia a nitidez ao redor das principais arestas presentes na imagem. A técnica de extração de saliência escolhida utiliza uma abordagem *bottom-up* baseada nos contrastes dos canais de cores da imagem. Os resultados mostram que a avaliação em regiões de saliência é bastante sensível à combinação entre métrica utilizada e distorção aplicada na imagem avaliada

Palavras-chave: Avaliação de qualidade de imagem, Métricas objetivas, Detecção de saliência em imagens

Abstract

The quality assessment of images and videos has a key role in many applications. Subjective assessments are not an option due to high costs and implementation time to obtain some satisfactory results. Objective metrics are required, but should maintain a good correlation with subjective results and therefore they should be based on features and behaviours of Human Visual System. This paper proposes to combine objective metrics, already established in literature, with the detection of regions of greater visual attention, regions that tend to catch the attention of those who observe. Studies suggest that these regions have greater impact on quality perception. Three different metrics will be compared between their simple application and in salient regions. The first metric is a full reference metric based on structural similarity of the images. The second one is a full reference metric that measures luminance variations. The last metric is a no reference metric that measures sharpness around main edges of the image. The saliency extraction technique uses a bottom-up approach based on contrast of image's color channels. The results show that quality assessment on salient regions is very sensitive to the combination of the objective metric and the distortion applied on the images.

Key-words: Image Quality Assessment, Objective Metrics, Image Saliency Detection

Lista de ilustrações

Figura 1 – Imagem de referência (mais à esquerda) e duas versões processadas com filtros Gaussianos.	14
Figura 2 – Exemplo das etapas de processamento para obtenção do mapa de saliência, na extrema direita temos o <i>ground truth</i> para efeito de comparação.	16
Figura 3 – Fluxo do algoritmo <i>Perceptual Blur</i>	23
Figura 4 – Comparação entre a técnica utilizada (coluna mais à direita) e outras técnicas de detecção de saliência	25
Figura 5 – Comparações entre as avaliações subjetivas e objetivas. Os três gráficos mais à esquerda são resultado do SSIM puro e à direita, do SSIM-Sal	28
Figura 6 – Exemplo de imagem retirada do D2. A imagem obteve SSIM = 0,72739 e SSIM-Sal = 0,46049	29
Figura 7 – Comparações entre PB e PB-Sal com os resultados subjetivos dos 3 bancos de imagens	30
Figura 8 – Comparação das métricas PSI e PSI-Sal com os valores subjetivos dos <i>datasets</i>	31
Figura 9 – Duas imagens do mesmo <i>dataset</i> com compressões em taxas diferentes. O efeito de <i>blockiness</i> pode ser notado na imagem à esquerda.	32

Lista de tabelas

Tabela 1 – Comparação do desempenho entre SSIM e SSIM-Sal.	27
Tabela 2 – Desempenho de PB e PB-Sal	29
Tabela 3 – Desempenho de PSI e PSI-Sal	31

Lista de abreviaturas e siglas

D1	Dataset1
D2	Dataset2
D3	Dataset3
DMOS	Differential Mean Opinios Score
FR	Full Reference
JNB	Just Noticeable Blur
LIVE	Laboratory for Image & Video Engineering
MOS	Mean Opinion Score
NR	No Reference
PB	Perceptual Blur
PLCC	Pearson Linear Coeficients Correlation
PSI	Perceptual Sharpness Index
SROCC	Spearman's Rank Correlation Coefficient
SSIM	Structural Similarity
SVH	Sistema Visual Humano

Sumário

1	Introdução	10
1.1	Objetivo	11
1.2	Estrutura	11
2	Conceitos Básicos	12
2.1	Avaliação Subjetiva	12
2.2	Avaliação Objetiva	13
2.2.1	Métricas <i>Full Reference</i>	13
2.2.2	Métricas <i>No Reference</i>	14
2.3	Saliência	15
3	Materiais e Métodos	17
3.1	Bases de imagens	17
3.2	Medidas de desempenho	17
3.3	Métricas utilizadas	18
3.3.1	<i>Perceptual Sharpness Index (PSI)</i>	19
3.3.2	<i>Structural Similarity</i>	20
3.3.3	<i>Perceptual Blur</i>	22
3.3.4	Extração de saliência	24
3.4	Metodologia	26
4	Resultados e Discussão	27
4.1	SSIM	27
4.2	PB	29
4.3	PSI	30
5	Conclusão	33
	Referências Bibliográficas	34
	APÊNDICES	37
	APÊNDICE A – Perceptual Blur	38
	APÊNDICE B – SSIM	39
	APÊNDICE C – Perceptual Sharpness Index	45

1 Introdução

Como resultado dos avanços em técnicas de compressão e transmissão, os conteúdos multimídia se tornaram meios de comunicação e entretenimento muito populares. Atualmente o compartilhamento de vídeos e imagens permeia nossa rotina, estando presente na maioria das aplicações. Os serviços de *streaming* de vídeos também crescem a cada dia, chegando a faturar mais do que canais de TV consolidados, seja transmitindo vídeos e séries, ou conteúdo produzido pelos próprios usuários. Parte desse panorama só é possível graças à qualidade que o usuário experimenta nessas aplicações, não apenas com a usabilidade, mas também com a qualidade das imagens proporcionadas.

Devido às etapas de processamento, compressão e transmissão, algumas distorções podem ser inseridas na imagem. Portanto nos contextos em que a qualidade é um fator crítico no sucesso da aplicação, ela precisa ser mensurada. A avaliação de qualidade mais confiável ainda é a fornecida por nós mesmos, seres humanos. De tal forma que as métricas de avaliação objetivas são analisadas de acordo com a correlação que obtém com um conjunto de avaliações subjetivas.

As métricas de avaliação objetivas costumam se basear no comportamento do sistema visual humano (SVH), para se aproximar cada vez mais da sua percepção. Conforme avança o entendimento sobre o SVH, melhores serão as métricas desenvolvidas.

As métricas de qualidade objetivas podem ser divididas de acordo com a quantidade de informação que dispõem da imagem analisada. As com referência ou *Full Reference* (FR), são aquelas que possuem acesso à outro estado da imagem, geralmente anterior ao seu processamento. Métricas sem referência ou *No Reference* (NR), se baseiam apenas em características da imagem avaliada e costumam se adequar melhor às situações práticas, pois nem sempre haverá outra imagem para comparação [1]. Por não ter outra imagem para referência, as métricas NR tendem a se basear mais em aspectos do SVH.

Mas ao receber estímulos visuais a partir de uma imagem, o SVH não processa, ao mesmo tempo, todos os elementos contidos nela. Esse processamento é guiado por uma série de fatores aos quais se dá o nome de atenção visual, que é a capacidade do SVH de focar em certos elementos do campo visual, e esses elementos tendem a se localizar nas chamadas regiões salientes [3]. Regiões salientes da imagem

são as áreas da imagem que retêm a atenção de quem as observa, seja pela sua posição, contraste ou outra característica.

Neste trabalho serão propostos métodos de avaliação da qualidade de imagem, baseados em regiões de saliência.

1.1 Objetivo

O objetivo deste trabalho é aproximar as métricas objetivas da percepção do SVH, propondo a aplicação de métricas de nitidez apenas nas regiões de saliência. O intuito principal é evitar o efeito de profundidade de campo (*Depth of Field*), quando um objeto é focado e o fundo da imagem fica borrado, que pode acabar influenciando negativamente as avaliações de nitidez quando o *background* é muito maior que a região em foco.

1.2 Estrutura

Este trabalho está dividido da seguinte forma. O Capítulo 2 apresenta conceitos básicos para melhor entendimento da área de estudo, dos tipos de métricas e do trabalho em questão. No Capítulo 3 serão descritas as métricas utilizadas e a suas aplicações. O Capítulo 4 abordará os experimentos realizados e resultados obtidos. Por fim, o Capítulo 5 apresentará as conclusões e propostas de trabalhos futuros.

2 Conceitos Básicos

Nesse capítulo serão abordados conceitos relevantes relacionados às métricas de avaliação de qualidade.

2.1 Avaliação Subjetiva

Avaliação subjetiva é aquela que é diretamente correlacionada com a percepção humana de qualidade [8]. Essa avaliação é feita por um conjunto de voluntários, escolhidos como amostragem do público em geral, de forma que o resultado possa ser generalizado. Para obter resultados confiáveis e possíveis de serem reproduzidos posteriormente, essas avaliações são padronizadas com diversos parâmetros, como o ambiente de teste, instruções para avaliação, equipamento utilizado, entre outros [7].

Os resultados são obtidos submetendo imagens de um banco de imagens, dispostas de forma aleatória, à avaliação dos voluntários. O banco de imagens ou *dataset* é composto por imagens de referência e imagens geradas a partir de algum processamento sobre elas. A avaliação mais comum é feita numa escala linear dividida em 5 regiões de igual amplitude e a cada região é atribuída um dos seguintes adjetivos: *Bad*, *Poor*, *Fair*, *Good*, e *Excellent*. A escala normalmente varia de 0 a 10 ou a 100.

Existem vários métodos de realizar a avaliação e eles diferem entre si em questões como: o voluntário ter ou não o conhecimento de quais são as imagens de referência, se as imagens de referência serão mostradas e como elas serão mostradas [9]. O tipo de avaliação depende do estudo que está sendo realizado e do que se espera obter com ele.

Embora esses resultados sejam precisos é necessário muito tempo e recursos para planejar e executar esse tipo de estratégia, impossibilitando sua aplicação em um contexto prático.

O principal resultado obtido nessas abordagens é o *Mean Opinion Score* (MOS), que é a pontuação média obtida por cada imagem ao fim da avaliação. Outro *score* resultante dessas avaliações é o *Differential Mean Opinion Score* (DMOS). Enquanto o MOS representa a avaliação média sobre uma determinada imagem de teste, o DMOS é obtido através da diferença entre o *score* subjetivo da imagem original e da imagem processada, ou seja, quanto maior o DMOS, pior foi a nota daquela imagem.

Por levar em consideração a imagem de referência o DMOS é mais utilizado para avaliar métricas objetivas FR, enquanto o MOS é mais indicado para métricas NR

2.2 Avaliação Objetiva

Avaliações objetivas buscam realizar a mesma tarefa que as subjetivas só que de forma automática, sempre buscando manter uma boa correlação com a percepção humana.

2.2.1 Métricas *Full Reference*

As métricas *full reference* só podem ser utilizadas em contextos que uma imagem de referência esteja disponível. Esse tipo de métrica gera um *score* para cada imagem a partir de comparações entre características da imagem avaliada e da referência.

Métricas mais simples como *Mean Square Error* (MSE) e *Peak Signal-to-Noise Ratio* (PSNR), avaliam apenas a luminância das imagens envolvidas. Apesar de se correlacionarem bem com resultados subjetivos ao avaliar certos tipos de distorções, como o blockiness resultante de uma descompressão. Essas métricas não avaliam outros aspectos que influenciam a percepção do SVH e por isso são usados apenas para comparações entre métricas.

Algumas outras métricas avaliam as imagens através do borramento ou *blur*. O *blur* é caracterizado pela suavização das arestas e a perda de detalhes na imagem, resultado da atenuação das altas frequências [10]. Esse efeito afeta diretamente os valores de luminância, provocando a convergência de valores entres pixels próximos, como é possível ver na Figura 1. A luminância é o valor RGB de um pixel em uma imagem em tons de cinza.

A métrica descrita em Frederique Crete *et al.* [7] avalia as variações de intensidades dos pixels vizinhos e normaliza a diferença dessa variação das duas imagens para obter uma estimativa do *blur*. Já uma das métricas em Marziliano *et al.* [10], propõe a avaliação do *blur* apenas em regiões próximas às arestas contidas na imagem.

Figura 1 – Imagem de referência (mais à esquerda) e duas versões processadas com filtros Gaussianos.



LIVE Image Quality Assessment Database release 2 - Gaussian Blur

Outras métricas mais complexas, como as descritas em Z. Wang and A. C. Bovik [11] e Z. Wang *et al.*[12], se utilizam do cálculo do contraste e da avaliação estrutural da imagem, além da luminância, para avaliar a qualidade entre imagens.

2.2.2 Métricas *No Reference*

As métricas NR são aplicáveis nos cenários em que a imagem de referência não está à disposição. Para se adaptar à falta de uma referência, essas métricas precisam se basear no SVH para se aproximar das avaliações subjetivas. Outro ponto importante é que na ausência de uma imagem para comparação, esse tipo de métrica costuma requerer treinamento.

Assim como ocorre com as FR algumas métricas NR também avaliam o *blur* da imagem, pois o *blur* é o inverso da nitidez, que é um fator importante na percepção de qualidade.

Em Ferzli e Karam [5] é introduzido o conceito de *Just Noticeable Blur* (JNB) baseado na característica do SVH de mascarar o efeito de *blur* quando este ocorre abaixo de um certo limiar. A métrica proposta calcula a propabilidade de detectar *blur* ao redor das arestas a partir do JNB, um método similar é proposto em D., Niranjan *et al.* [13].

A métrica em Feichtenhofer *et al.* [1] também se utiliza do JNB para calcular o gradiente local nas principais arestas, além de incluir outros conceitos, como o da

atenção espacial, levando em consideração a posição da aresta na imagem para determinar sua influência na percepção de nitidez.

2.3 Saliência

Com o aumento do entendimento sobre o SVH as métricas mais recentes vêm sendo desenvolvidas levando em conta a atenção visual.

A atenção visual é um conjunto de aspectos e comportamentos do SVH que influenciam na percepção de qualidade, como por exemplo a movimentação dos olhos pela cena [22], dando mais importância à certas regiões da imagem e desprezando outras. Essa atenção seletiva ocorre devido à imensa quantidade de informação que o SVH recebe e não é capaz de processar simultaneamente [14].

Apesar de não ser completamente compreendida ainda, a atenção visual pode ser descrita por dois processos fisiológicos que combinados resultam na seleção de áreas de interesse em uma imagem. Esses processos são a seleção *Bottom-up* e a *Top-down* [15].

O mecanismo de seleção *bottom-up* funciona mais rapidamente e é guiado apenas por estímulos visuais presentes na imagem. Entretanto o mecanismo *top-down* é regido por preferências individuais, como memórias ou interesses. Por ser extremamente complexo de ser reproduzido algorítmicamente, o processo *top-down* não é levado em conta na maioria das técnicas de detecção de saliência.

As abordagens *bottom-up* costumam levar em consideração características como: contraste, intensidade, cores, orientação e movimento [16]. Essas características, quando extraídas, formam os *feature maps* que serão combinados para gerar um mapa de saliências para a imagem.

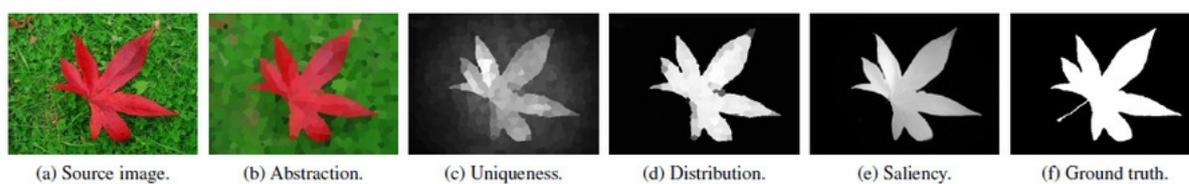
Zhang e Slaroff [18] consideram a movimentação dos olhos e a divisão entre *background* e *foreground* para detectar as regiões de atenção. O artigo propõe a extração de diversos *feature maps* para obter *boolean maps*. Esses mapas booleanos são versões binarizadas, apenas preto e branco, dos mapas de *features*. Por fim ele gera mapas de atenção visual para cada mapa booleano, utilizando conceitos psicológicos descritos por V. Mazza *et al.*[19] e E. Rubin. [20]. Os mapas de saliência podem ser gerados através dos mapas de atenção resultantes.

Em Ming-Ming Cheng *et al.* [17] são demonstradas duas técnicas de detecção de saliência, ambas baseadas em contraste. A primeira utiliza o histograma de cores da imagem para detectar o contraste global entre os pixels. A segunda avalia o contraste

de forma local, pois tendemos a prestar mais atenção em regiões com maior contraste em volta [21].

Perazzi, F. *et al.* [16] também utiliza o contraste para detectar as regiões de saliência da imagem. A partir do agrupamento ou *clustering* de pixels com características semelhantes é possível definir a singularidade, ou raridade, dos pixels na imagem e também a sua distribuição. A união dessas características gera a região de saliência, como se pode ver na Figura ??.

Figura 2 – Exemplo das etapas de processamento para obtenção do mapa de saliência, na extrema direita temos o *ground truth* para efeito de comparação.



Perazzi, F. et al (2012)

3 Materiais e Métodos

Neste capítulo serão descritos as métricas, bases de imagens e metodologias utilizadas visando atingir o objetivo proposto no início, de aplicar métricas de nitidez em regiões de saliência.

3.1 Bases de imagens

Os três *datasets* (D1, D2 e D3) utilizados para validação das métricas foram fornecidos pelo *Laboratory for Image & Video Engineering* (LIVE) em dois releases diferentes. Todas as bases foram geradas a partir das mesmas vinte e nove imagens de referência, variando apenas a distorção aplicada.

No D1 as imagens foram comprimidas utilizando JPEG com diferentes taxas de compressão, escolhidas aleatoriamente, de forma que cada imagem gerasse outras três totalizando cento e dezesseis imagens. O MOS fornecido foi obtido a partir de um estudo com vinte voluntários em que as imagens de referência também foram inseridas na avaliação, porém de maneira aleatória e sem o conhecimento dos voluntários.

O D2 e D3 fazem parte do segundo *release* junto com outros três *datasets*. No D2 as imagens foram comprimidas com JPEG2000, também variando de forma aleatória a compressão e gerando outras três imagens a partir da referência. O D3 foi criado a partir da aplicação de um filtro Gaussiano 2-D (circular-simétrico), de forma a obter cinco imagens a partir da original. Assim como no D1, as imagens originais também foram submetidas à avaliação, porém tanto no D2 como no D3, o *score* fornecido não foi o MOS, mas o DMOS, ou seja, quanto menor o *score* melhor foi a avaliação da imagem.

3.2 Medidas de desempenho

Para comparar e medir o desempenho das avaliações objetivas com relação às avaliações subjetivas das bases de imagens, foram utilizadas duas medidas de correlação para analisar a monotonicidade e a acurácia dos resultados.

A acurácia será medida pelos coeficientes lineares de correlação de Pearson (PLCC em inglês). Nesta medida os resultados variam numa escala de -1 a 1, onde 1 significa uma correlação direta entre os valores avaliados e -1 mostra que os valores são inversamente correlacionados. O PLCC é calculado a partir da Equação 3.1.

$$PLCC = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (3.1)$$

Onde n é o tamanho dos conjuntos de dados que estão sendo correlacionados, x_i e y_i são, respectivamente, valores dos *scores* subjetivos e objetivos e \bar{x} e \bar{y} são as médias desses conjuntos de valores. A fórmula pode ser simplificada para a Equação 3.2.

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \quad (3.2)$$

Onde $cov(X,Y)$ é a covariância dos conjuntos e σ_X e σ_Y são os desvios padrão.

A outra medida utilizada para medir o desempenho foi a dos coeficientes de correlação de postos de Spearman (SROCC em inglês). Essa correlção é calculada a partir da posição ou ranqueamento dos valores dentro dos seus conjuntos. Se nos conjuntos houverem valores repetidos, com o mesmo rank, deve-se utilizar a Equação 3.3. Para conjuntos com todos os valores distintos pode-se usar a Equação 3.4.

$$r_s = \frac{cov(r_{gX}, r_{gY})}{\sigma_{r_{gX}} \sigma_{r_{gY}}} \quad (3.3)$$

$$r_s = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (3.4)$$

Onde r_{gX} e r_{gY} são os ranks dos valores subjetivos e objetivos em seus respectivos conjuntos e d_i é a diferença entre os ranks das duas avaliações associados à mesma imagem.

3.3 Métricas utilizadas

Para obter as avaliações objetivas sobre os bancos de imagens foram utilizadas as métricas propostas em Feichtenhofer *et al.* [1], Frederique *et al.* [7] e Z. Wang *et al.* [12], sendo a primeira uma métrica NR e as duas últimas FR. Para detecção das regiões salientes utilizou-se a técnica descrita em Zhang *et al.* [2]. As métricas serão descritas em detalhes nesta sessão.

3.3.1 Perceptual Sharpness Index (PSI)

O PSI é a métrica proposta no artigo de Feichtenhofer *et al.* [1]. Nesse artigo os autores descrevem uma métrica NR que não necessita de treinamento e é baseada no domínio espacial das imagens.

Ela visa atribuir um índice que represente a nitidez percebida na imagem e foca sua análise no contraste local ao redor das arestas mais relevantes da imagem, pois o SVH tende a focar em locais de alto contraste [23]. O processo é dividido em quatro etapas.

A primeira etapa é gerar um mapa de arestas através de uma seleção adaptativa. Nesse mapa estarão representadas apenas as arestas mais relevantes da imagem. Para esse fim, um filtro de Sobel é aplicado sobre a luminância, ou seja, a imagem convertida em tons de cinza, para destacar suas arestas.

A aplicação do filtro é seguida por um processo de binarização utilizando um *threshold* adaptativo calculado na Equação 3.5.

$$T = \alpha \bar{G} \quad (3.5)$$

Onde \bar{G} representa a média dos valores de luminância após o filtro de Sobel e α é uma constante determinada experimentalmente, na maioria dos casos testados no artigo $\alpha = 5.6$.

Após a binarização, a imagem resultante passa por um processo de afinamento.

A segunda etapa é a medição da largura das arestas, mas apenas arestas de gradiente aproximadamente vertical são levadas em conta nesse processo. Os autores concluíram através de experimentos que, levar em conta arestas de gradiente horizontal não apresenta melhora na performance e arestas diagonais resultam em perda de performance [1].

As larguras das arestas são computadas seguindo a Equação 3.6.

$$w(X) = \frac{w_{up}(X) + w_{down}(X)}{\cos(\Delta\phi(X))} \quad (3.6)$$

Onde $w_{up}(X)$ e $w_{down}(X)$ são, respectivamente, as distâncias entre o pixel X da aresta e os pixels locais de maior $I_{max}(X)$ e menor $I_{min}(X)$ luminân-

cia. O denominador representa o cosseno da diferença entre a direção do gradiente $\phi(X) = \tan^{-1}(I_y(X)/I_x(X))$ e a direção da aresta.

A terceira etapa do processo busca modelar a percepção humana da acutância, motivados pelo fato que arestas de alto contraste são percebidas como mais nítidas. A acutância descreve o quão rapidamente há transições de informação entre regiões da imagem [24]. A medida da acutância é dada pela Equação 3.7.

$$m(X) = \frac{I_{max}(X) - I_{min}(X)}{w(X)} \quad (3.7)$$

Com isso as arestas medidas serão decrementadas pela sua acutância dependendo das suas larguras, para que arestas menos nítidas percam importância.

$$w_{PSI}(X) = \begin{cases} w(X) - m(X), & w(X) \geq w_{JNB} \\ w(X) & otherwise. \end{cases} \quad (3.8)$$

Na Equação 3.8 $w_{PSI}(X)$ é o PSI atribuído à aresta X e w_{JNB} é um valor associado às larguras de arestas que possuam o JNB. No artigo e nos experimentos conduzidos neste documento $w_{JNB} = 3$.

Por fim, na última etapa são realizadas estimativas do PSI local e global da imagem. Para estimar o PSI local, a imagem é dividida em blocos de 32x32 pixels e o PSI de cada bloco é dado pelo $w_{PSI}(X)$ médio das arestas do bloco.

O PSI global é obtido pela média dos blocos com maior PSI local.

3.3.2 Structural Similarity

A ideia principal da métrica FR do artigo de Z. Wang *et al* [12] é que os pixels de uma imagem carregam informações importantes sobre os objetos e formas presentes nela, especialmente quando estes pixels estão mais próximos entre si. O *Structural Similarity index* (SSIM) é uma evolução da métrica proposta por Z. Wang and A. C. Bovik [11].

Além da informação estrutural, os autores consideram que a luminância também desempenha importante função na percepção de qualidade. Porém, como a estrutura na imagem não depende da luminância, eles optam por calcular esses fatores de forma separada. O contraste também é levado em conta pelo SSIM pois também

influencia na nossa avaliação subjetiva, mas, assim como a luminância, ele é calculado separadamente.

As formulas para luminância e contraste entre imagens correspondem às Equações 3.9 e 3.10 respectivamente.

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (3.9)$$

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (3.10)$$

Na equação da luminância $l(\mathbf{x}, \mathbf{y})$, μ_x , μ_y representam, respectivamente, a média dos valores de luminância em \mathbf{x} e \mathbf{y} . O C_1 é uma constante que corresponde a $(K_1L)^2$, em que K_1 é uma constante que foi definida através de experimentos e L é a faixa de valores de luminância, numa imagem com 8 bits/pixel $L = 255$.

O contraste $c(\mathbf{x}, \mathbf{y})$ é calculado em termos dos desvios-padrão de \mathbf{x} e \mathbf{y} , representados respectivamente por σ_x , σ_y . C_2 é uma constante obtida de forma similar a C_1 e seu valor é expresso por $(K_2L)^2$. Os valores padrão para K_1 e K_2 são 0,01 e 0,03.

Por fim a comparação estrutural entre \mathbf{x} e \mathbf{y} é dada pela Equação 3.11.

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (3.11)$$

Onde $C_3 = C_2/2$. Todas as constantes C_1, C_2, C_3 só foram incluídas nas fórmulas após os resultados em [11] demonstrarem valores muito instáveis quando os denominadores se aproximavam de zero.

Definidos esses três fatores, o valor final do SSIM é dado por:

$$SSIM(\mathbf{x}, \mathbf{y}) = [l(\mathbf{x}, \mathbf{y})]^\alpha \cdot [c(\mathbf{x}, \mathbf{y})]^\beta \cdot [s(\mathbf{x}, \mathbf{y})]^\gamma \quad (3.12)$$

Onde α, β, γ são parâmetros de ajuste da importância de cada elemento. No artigo $\alpha = \beta = \gamma = 1$, com esses valores as fórmulas podem ser combinadas na

Equação 3.13.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (3.13)$$

3.3.3 Perceptual Blur

Ao contrário do PSI o artigo escrito por Frederique *et al.* [7], propõe avaliar o oposto da nitidez das imagens, o seu borramento. O *Perceptual Blur* (PB) é uma métrica FR e sua ideia principal é analisar a diferença entre a luminância dos pixels, dado que em imagens mais borradas a intensidade dos pixels tende a convergir ao mesmo valor.

O efeito de *blur* não ocorre, necessariamente, de forma homogênea na imagem, mas por menor que tenha sido a área afetada ele também não afeta um único pixel. Levando essas características em consideração, os autores propõem avaliar as variações de luminância apenas em pixels vizinhos, podendo comparar essa variação com a imagem original e determinar se houve borramento. As etapas do processo estão esquematizadas na Figura 3 .

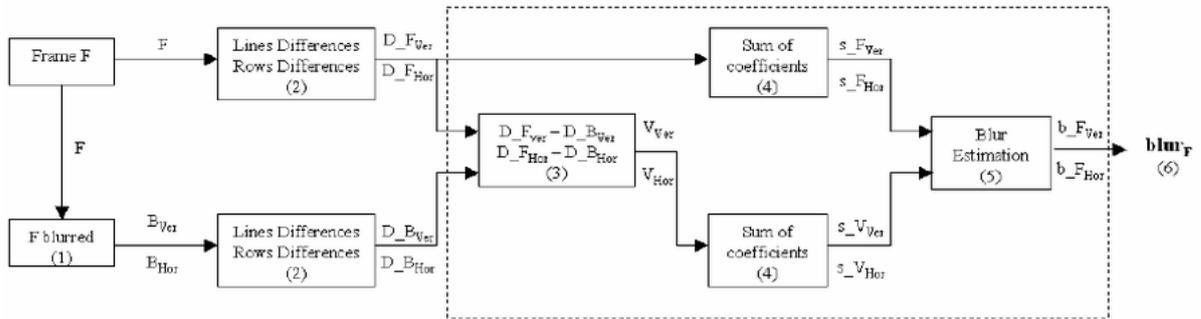
Para avaliar as variações de intensidade em pixels vizinhos, o algoritmo gera duas imagens a partir da diferença absoluta entre a luminância de um pixel e seus vizinhos verticais e horizontais, como mostrado nas Equações 3.14 e 3.15.

$$D_{F_{ver}}(i, j) = Abs(F(i, j) - F(i - 1, j)), i = 1 \rightarrow m - 1, j = 0 \rightarrow n - 1 \quad (3.14)$$

$$D_{F_{hor}}(i, j) = Abs(F(i, j) - F(i, j - 1)), i = 1 \rightarrow m - 1, j = 0 \rightarrow n - 1 \quad (3.15)$$

Onde $D_{F_{ver}}(i, j)$ é a diferença absoluta vertical do pixel (i, j) , $F(i, j)$ é o valor de intensidade do pixel (i, j) , m e n são respectivamente a largura e altura da imagem e Abs retorna o valor absoluto da diferença. Esse cálculo é feito em ambas as imagens, a de referência e a, possivelmente, borrada.

Figura 3 – Fluxo do algoritmo *Perceptual Blur*



Frederique et al. (2007)

Após calcular imagens das diferenças absolutas a métrica analisa a diferença desses valores. Quanto maior a diferença, maior foi o borramento. Essa variação é calculada apenas para pixels cuja variação absoluta tenha diminuído.

$$V_{Ver} = \text{Max}(0, D_{F_{ver}}(i, j) - D_{B_{ver}}(i, j)) \quad \text{for } i = 1 \rightarrow m-1, \quad j = 1 \rightarrow n-1 \quad (3.16)$$

$$V_{Hor} = \text{Max}(0, D_{F_{hor}}(i, j) - D_{B_{hor}}(i, j)) \quad \text{for } i = 1 \rightarrow m-1, \quad j = 1 \rightarrow n-1 \quad (3.17)$$

V_{Ver} é a variação vertical entre as diferenças absolutas da imagem de referência F e a imagem B . V_{Hor} é, de maneira análoga, a variação horizontal.

Para comparar as variações obtidas em 3.16 e 3.17, com as diferenças absolutas da imagem original, são calculados os seguintes coeficientes:

$$s_{F_{ver}} = \sum_{i,j=1}^{m-1,n-1} D_{F_{ver}}(i, j) \quad s_{F_{hor}} = \sum_{i,j=1}^{m-1,n-1} D_{F_{hor}}(i, j) \quad (3.18)$$

$$s_{V_{ver}} = \sum_{i,j=1}^{m-1,n-1} V_{ver}(i, j) \quad s_{V_{hor}} = \sum_{i,j=1}^{m-1,n-1} V_{hor}(i, j) \quad (3.19)$$

Por fim o *blur* horizontal e vertical é calculado de forma normalizada segundo as Equações em 3.20.

$$b_{F_{ver}} = \frac{s_{F_{ver}} - s_{V_{ver}}}{s_{F_{ver}}} \quad b_{F_{hor}} = \frac{s_{F_{hor}} - s_{V_{hor}}}{s_{F_{hor}}} \quad (3.20)$$

O valor de *blur* final atribuído à imagem será aquele de maior valor entre o horizontal e vertical.

3.3.4 Extração de saliência

O método de detecção de saliência utilizado foi proposto por Zhang *et al* [2]. No artigo os autores demonstram uma técnica *bottom-up* baseada no contraste dos canais de cores da imagem, combinada com informações sobre a atenção espacial. A atenção espacial se refere ao comportamento do SVH de dar mais importância às áreas mais centrais da imagem [2]. A Figura 4 mostra alguns resultados desta técnica comparados à outras do estado da arte.

O algoritmo proposto no artigo começa pela escolha adaptativa do canal de informação mais relevante na imagem. Baseados no funcionamento dos fotosensores presentes em nossa retina, os autores extraem de cada imagem canais de cores formadas por um único componente de cor, estes são: R (vermelho), G (verde), B(azul) e Y(amarelo). O amarelo não é extraído puro da imagem, mas estudos feitos por Gelasca *et al* [25], mostram que ele desempenha um papel importante para o SVH.

Os canais são gerados segundo as equações a seguir.

$$\tilde{R} = | R - (G + B)/2 | \quad (3.21)$$

$$\tilde{G} = | G - (R + B)/2 | \quad (3.22)$$

$$\tilde{B} = | B - (R + G)/2 | \quad (3.23)$$

$$\tilde{Y} = | (R + G)/2 - | R - G | /2 - B | \quad (3.24)$$

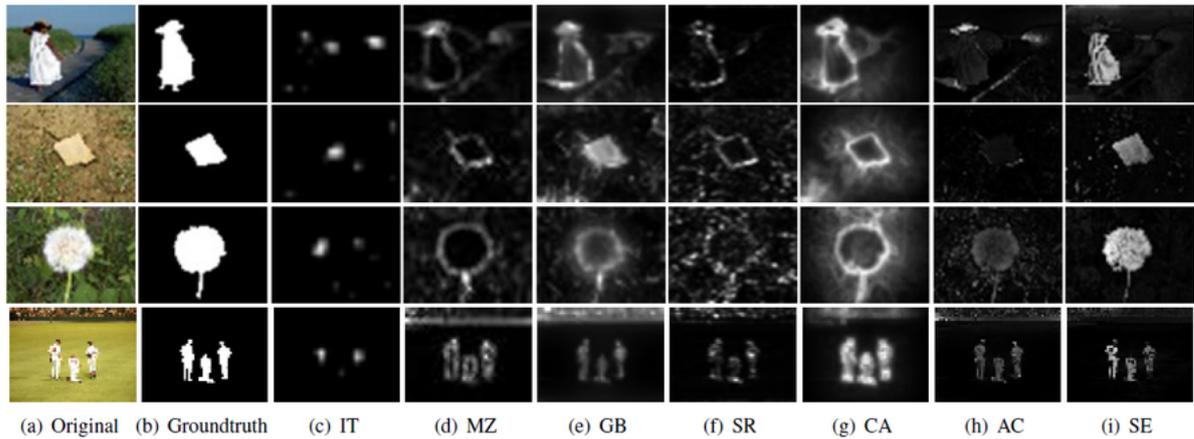
Onde R, G, B são os valores no espaço de cores RGB da imagem e $\tilde{R}, \tilde{G}, \tilde{B}, \tilde{Y}$ são os canais de cores que representam a relevância de um componente comparado aos demais.

A escolha do canal de cor é feita baseada no contraste de cada um dos canais, segundo a Equação 3.25.

$$\sigma_k = \frac{1}{W \times H} \sum_i \sum_j (I_k(i, j) - \bar{I}_k)^2 \quad (3.25)$$

Em que σ_k é a variância das intensidades dos pixels no canal de cor k , $I_k(i, j)$ é a intensidade do pixel (i, j) no canal k , \bar{I}_k é a média das intensidades dos pixels do canal k e W e H são respectivamente a largura e altura da imagem. O canal que tiver a maior variância será escolhido para gerar o mapa de saliência.

Figura 4 – Comparação entre a técnica utilizada (coluna mais à direita) e outras técnicas de detecção de saliência



Zhang et al. (2012)

Definido o canal c que será utilizado, o mapa de saliência pode ser gerado pela Equação 3.26 e posteriormente normalizado pela Equação 3.27 .

$$s(i, j) = |I_c(i, j) - \bar{I}_c| \quad (3.26)$$

$$s'(i, j) = 255 \times \frac{s(i, j) - s_{min}}{s_{max} - s_{min}} \quad (3.27)$$

Onde $s(i, j)$, $s'(i, j)$ são, respectivamente o mapa de saliência e o mapa de saliência normalizado e s_{max} , s_{min} são os valores máximo e mínimo presentes no mapa.

Para refletir a tendência do SVH em focar mais as regiões centrais da imagem, os autores definem uma função de atenção espacial dada pela Equação 3.28

$$f(i, j) = \frac{1}{1 + d(i, j)/L} \quad (3.28)$$

$d(i, j)$ denota a distância entre a localização do pixel e o centro da imagem e L é a metade da diagonal da imagem. Com essa ponderação baseada na localização, o mapa de saliência final é definido por:

$$S(i, j) = s'(i, j) \times f(i, j) \quad (3.29)$$

3.4 Metodologia

A metodologia utilizada neste estudo consiste basicamente em duas etapas. A primeira é obter avaliações objetivas aplicando as métricas descritas neste capítulo nos bancos de imagens citados anteriormente. A segunda é gerar três novos bancos de imagens formados pelas regiões de saliência das imagens dos *datasets* originais.

Nesta segunda etapa foi necessário aplicar a extração de saliência nos três *datasets* e realizar um pós-processamento nos mapas de saliência gerados, pois o intuito do estudo é propor uma métrica de nitidez aplicada apenas na região de saliência.

Para essa finalidade, os mapas de saliência foram binarizados, segundo a binarização adaptativa de Otsu, em seguida aplicou-se um filtro de abertura, para eliminar pequenas regiões isoladas. Por fim, com um mapa de saliência binarizado, foi possível definir uma subimagem que contivesse apenas a região mais saliente. Caso a subimagem representasse mais de 85% da imagem original, a imagem original era mantida.

Também vale ressaltar que a extração de saliência e a definição da subimagem foram feitas, em todos os casos, apenas nas imagens mais nítidas. Portanto as imagens borradas ou distorcidas pela compressão, foram reduzidas à mesma região definida como saliente nas imagens de referência.

De posse dos resultados objetivos aplicados nos *datasets* originais e nos *datasets* de saliência, foi possível comparar os resultados com as medidas de desempenho descritas na seção 3.2. Os resultados do estudo serão abordados no próximo capítulo.

4 Resultados e Discussão

Neste capítulo serão apresentados os resultados da aplicação das métricas SSIM, PB e PSI nas versões originais e salientes dos *datasets* fornecidos pelo LIVE. Também serão mostradas e discutidas a correlação entre os resultados aplicando as medidas PLCC e SROCC.

Os códigos utilizados para as avaliações objetivas se encontram nos apêndices A, B e C.

4.1 SSIM

Nesta sessão, os resultados apresentados como SSIM serão referentes à métrica original aplicada nos *datasets* (D1, D2, D3) da forma que foram fornecidos pelo LIVE. Os resultados SSIM-Sal farão referência à métrica aplicada sobre as subimagens dos *datasets* selecionadas como regiões de saliência.

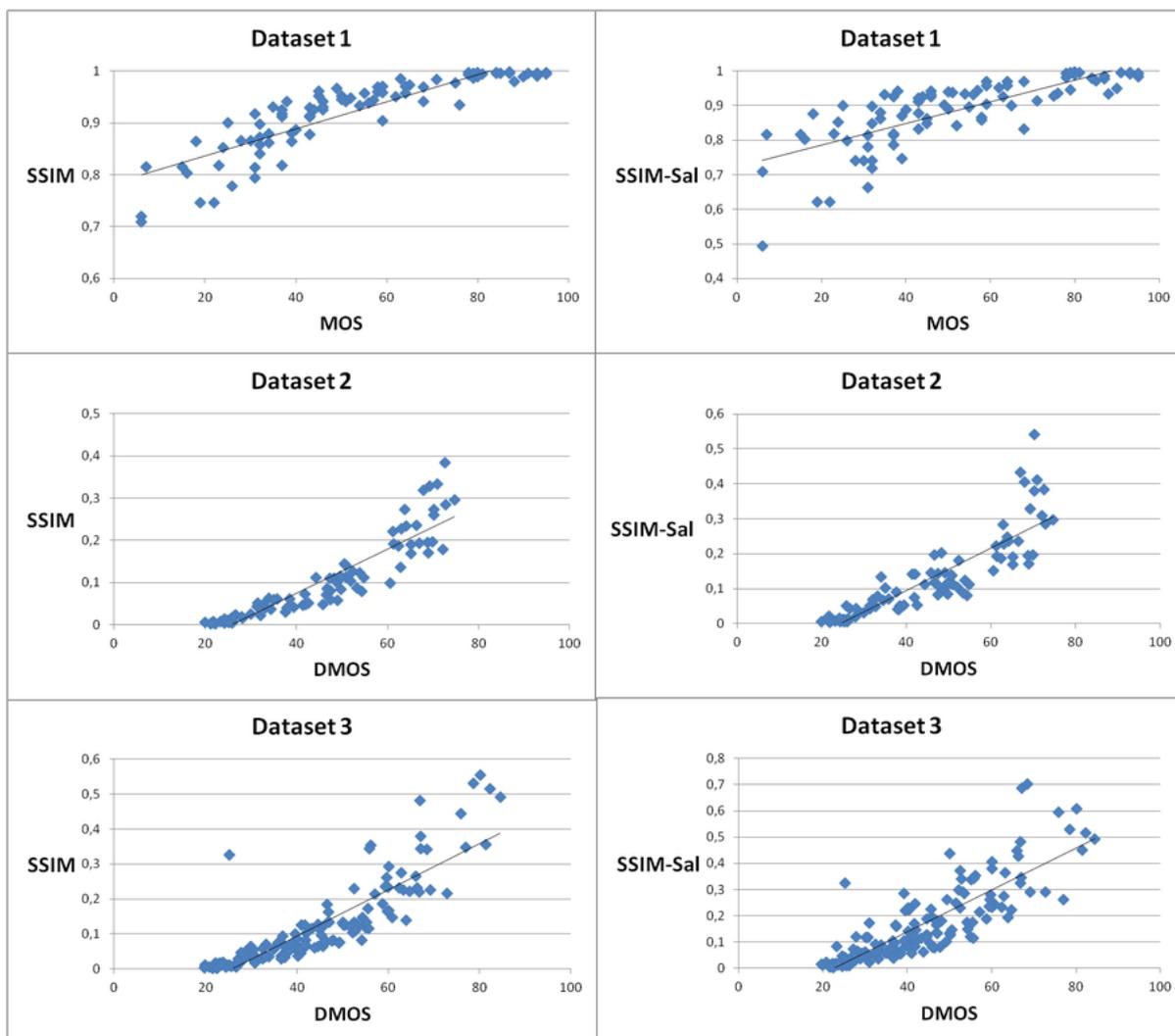
A Figura 5 mostra a correlação entre os *scores* objetivos e subjetivos após a aplicação de SSIM e SSIM-Sal. Na primeira coluna de gráficos estão as comparações entre o SSIM, aplicado aos *datasets* originais, e as avaliações subjetivas. Na segunda coluna, estão os resultados da aplicação apenas em regiões de saliência do *dataset*. Também estão representadas a função de regressão linear para cada gráfico. Os PLCCs e SROCCs podem ser vistos na Tabela 1.

Tabela 1 – Comparação do desempenho entre SSIM e SSIM-Sal.

	SSIM			SSIM-Sal		
	D1	D2	D3	D1	D2	D3
PLCC	0,868888	0,912559	0,863010	0,765836	0,866243	0,836929

	SSIM			SSIM-Sal		
SROCC	0,932954	0,962841	0,927408	0,850278	0,921089	0,877862

Figura 5 – Comparações entre as avaliações subjetivas e objetivas. Os três gráficos mais à esquerda são resultado do SSIM puro e à direita, do SSIM-Sal



Na Tabela 1 é possível ver que a métrica pura aplicada nos *datasets* se saiu melhor em todos os casos. Isso se deve ao fato de que imagens bem avaliadas pelo SSIM, obtiveram piores resultados quando avaliadas apenas sobre a região de saliência. Um exemplo é mostrado na Figura 6.

Figura 6 – Exemplo de imagem retirada do D2. A imagem obteve SSIM = 0,72739 e SSIM-Sal = 0,46049



LIVE database release 2 - JPEG2000

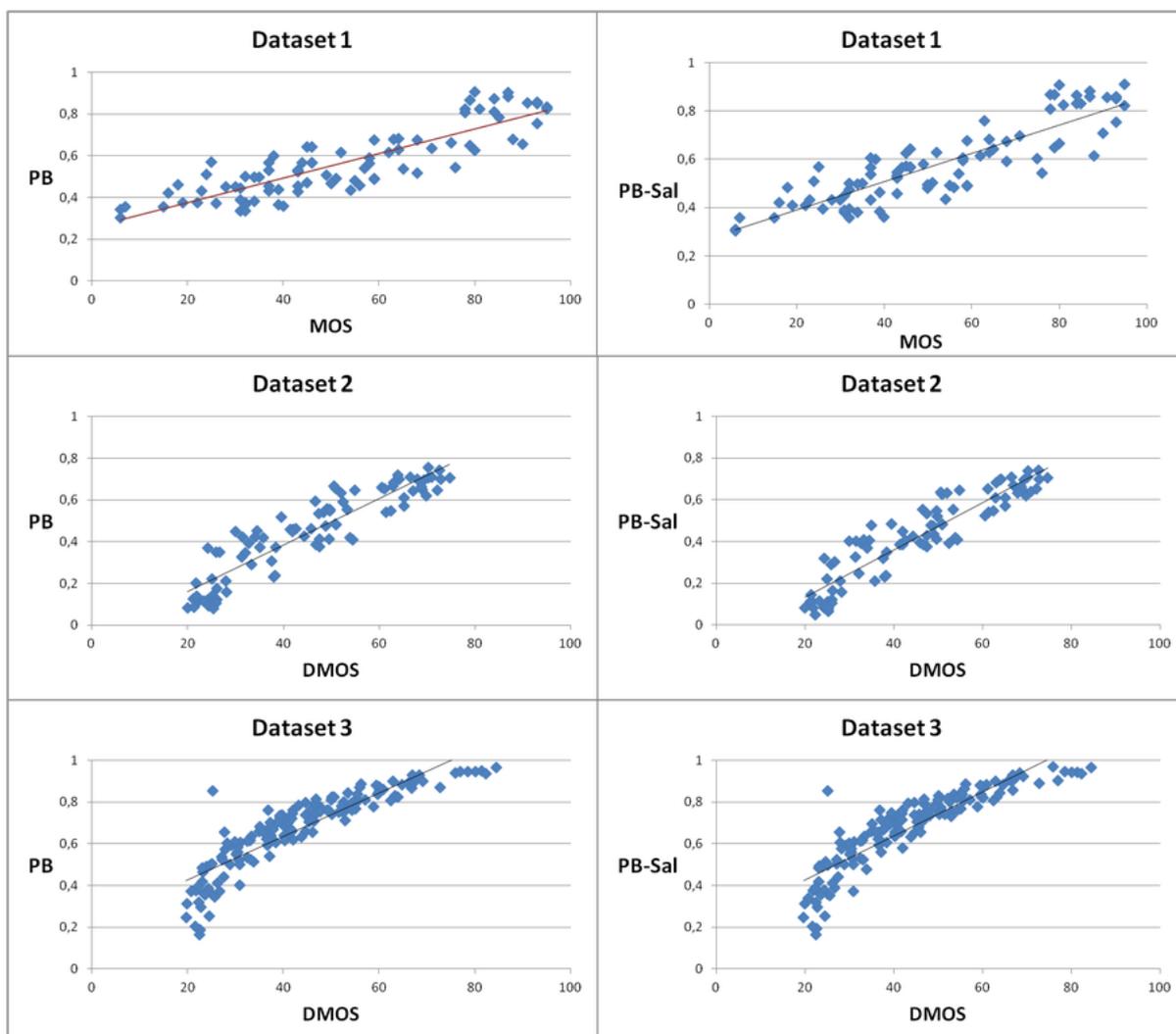
4.2 PB

Seguindo o padrão estabelecido na sessão anterior, PB irá se referir à aplicação da métrica nas imagens originalmente fornecidas e PB-Sal às imagens das regiões de saliência.

A comparação das avaliações de PB e PB-Sal com os valores subjetivos, são mostrados na Figura 7 e o desempenho das métricas está relacionado na Tabela 2.

Tabela 2 – Desempenho de PB e PB-Sal

	PB			PB-Sal		
	D1	D2	D3	D1	D2	D3
PLCC	0,86467	0,91148	0,88638	0,87141	0,91947	0,88537
SROCC	0,84869	0,91816	0,94149	0,85466	0,92585	0,93904

Figura 7 – Comparações entre PB e PB-Sal com os resultados subjetivos dos 3 bancos de imagens

Ao contrário do SSIM-Sal a métrica PB-Sal se mostrou melhor nos *datasets* em que as imagens foram geradas com compressão à diferentes taxas, mas teve um desempenho pior no D3, onde foi utilizado um filtro passa-baixa para borrar as imagens.

Isso se deve à métrica PB medir o *blur* na imagem através das diferenças de intensidade dos pixels. Apesar da imagem receber o filtro por igual em toda sua área, um filtro de borrimento provoca uma convergência muito grande de valores em regiões próximas.

4.3 PSI

Nesta sessão mostraremos os resultados da aplicação da última métrica apresentada neste estudo. Os resultados da aplicação do PSI e PSI-Sal são apresentados na Figura 8 e seus desempenhos comparados na Tabela 3.

Figura 8 – Comparação das métricas PSI e PSI-Sal com os valores subjetivos dos datasets

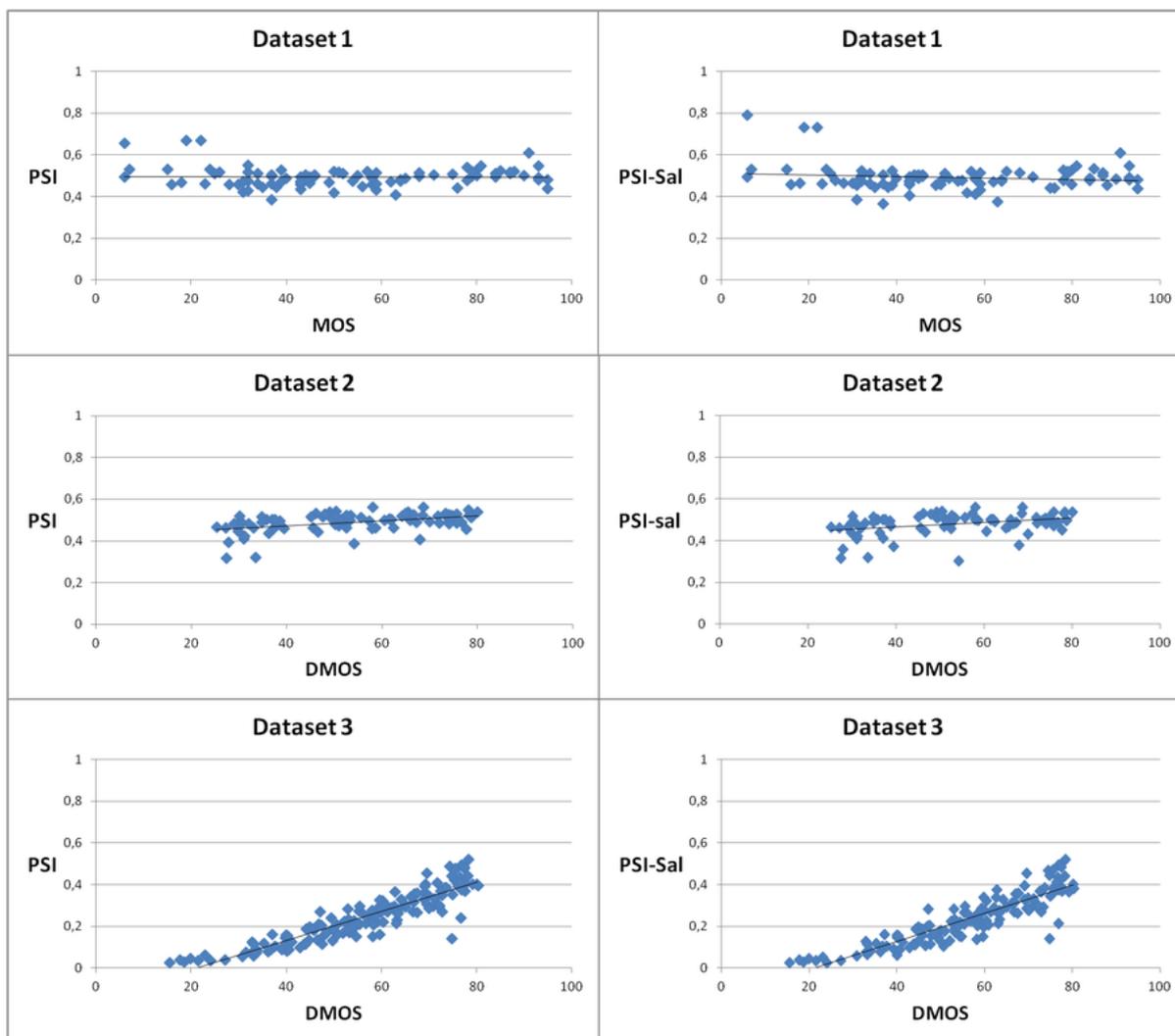


Tabela 3 – Desempenho de PSI e PSI-Sal

	PSI			PSI-Sal		
	D1	D2	D3	D1	D2	D3
PLCC	-0,022474	0,446721	0,907013	-0,139894	0,354547	0,889581
SROCC	0,103999	0,426806	0,917637	0,026854	0,298633	0,904366

Os resultados de PSI e PSI-Sal foram os piores das três métricas em D1 e D2. Esses resultados se devem ao fato do PSI calcular a nitidez baseado nas arestas do gráfico, ou seja, ele depende essencialmente da estrutura da imagem.

Porém como podemos ver na Figura 9 a compressão dos arquivos gera o efeito de *blockiness* nas imagens e isto afeta o desempenho da métrica. O problema principal é que o *blockiness* gera arestas artificiais na imagem.

Por outro lado, mesmo se saindo bem ao avaliar o D3, o desempenho do PSI foi melhor que o do PSI-Sal.

Figura 9 – Duas imagens do mesmo *dataset* com compressões em taxas diferentes. O efeito de *blockiness* pode ser notado na imagem à esquerda.



LIVE database release 1

5 Conclusão

Este trabalho propôs a combinação de três métricas de qualidade com a detecção de regiões de saliência, com o intuito de estreitar a correlação entre suas avaliações e os resultados subjetivos. As métricas abordadas foram: SSIM [12], uma métrica FR, baseada na estrutura, luminância e contraste das imagens. PB [7], também FR, baseada na variação absoluta de luminância entre pixels próximos. E o PSI [1], uma métrica NR, baseada na nitidez ao redor das arestas na imagem.

Para a obtenção dos resultados, todas as métricas foram aplicadas sobre três *datasets* originais do LIVE e suas respectivas versões contendo apenas as regiões de saliência das imagens. Na detecção de saliência utilizou-se a técnica descrita em [2]. Os mapas gerados foram pós-processados para obter imagens apenas das áreas de maior atenção visual.

Mesmo com alguns resultados positivos, como a métrica PB nos *datasets* D1 e D2, os resultados, em sua maioria, ficaram abaixo do esperado. Os destaques negativos resultaram da aplicação do PSI sobre D1 e D2. No caso do PSI, a correlação quase nula não se deve à aplicação nas regiões de saliência, mas a problemas entre o que a métrica avalia e o que as distorções, nestes casos, provocavam na imagem.

Os resultados do SSIM-Sal, embora um pouco abaixo, se mantiveram próximos aos do SSIM. O principal fator da queda de rendimento em regiões de saliência se deve ao fato de que estas regiões possuem, geralmente, menos variação de luminância e contraste que a imagem completa, já que a maior parte do contraste está entre a região saliente e o *background*, que foi cortado fora. Apesar disso, a correlação com os valores subjetivos se manteve alta.

Apesar dos resultados, o estudo realizado mostrou que avaliação de qualidade em regiões de saliência possuem uma boa correlação com os MOS e DMOS. Também se destaca a relação que as métricas tem com as distorções que a imagem sofreu, pois mesmo com as métricas mais robustas (PB e SSIM), as avaliações do D1 foram sempre as piores em relação aos resultados subjetivos.

Como trabalho futuro fica a reformulação da metodologia aplicada, principalmente na forma que as métricas vão ser aplicadas nas regiões de atenção visual. Uma possível mudança seria, ao invés de recortar uma subimagem, aplicar a métrica sobre a imagem cheia, mas dando um peso maior aos valores obtidos sobre a saliência.

Referências Bibliográficas

[1] - FEICHTENHOFER, Christoph; FASSOLD, Hannes; SCHALLAUER, Peter. A Perceptual Image Sharpness Metric Based on Local Edge Gradient Analysis. IEEE SIGNAL PROCESSING LETTERS, v. 20, n. 4, abr. 2013.

[2] - ZHANG, H. et al. A Simple and Effective Saliency Detection Approach. International Conference on Pattern Recognition (ICPR), Tsukuba, Japan, nov. 2012.

[3] - YU, Chun-yan; ZHANG, Wei-shi; WANG, Chun-li. A Saliency Detection Method Based on Global Contrast. International Journal of Signal Processing, Image Processing and Pattern Recognition, [S.L], v. 8, n. 7, p. 111-122, jan. 2015.

[4] - ROSSINI, Joaquim Carlos; GALERA, César. Atenção visual: estudos comportamentais da seleção baseada no espaço e no objeto. Estudos de Psicologia, São Paulo, v. 11, n. 1, p. 79-86, jan. 2006.

[5] - FERZLI, Rony; KARAM, Lina J.. A No-Reference Objective Image Sharpness Metric Based on the Notion of Just Noticeable Blur (JNB). IEEE Transactions On Image processing, [S.L], v. 18, n. 4, p. 717-728, abr. 2009.

[6] - VIDEO CLARITY. Understanding mos, jnd, and psnr. Disponível em: <<http://videoclarity.com/wpunderstandingjnddmospnr/>>. Acesso em: 21 set. 2017.

[7] - Frederique Crete-Roffet, Thierry Dolmiere, Patricia Ladret, Marina Nicolas. The Blur Effect: Perception and Estimation with a New No-Reference Perceptual Blur Metric. SPIE Electronic Imaging Symposium Conf Human Vision and Electronic Imaging, Jan 2007, San Jose, United States. XII, pp.EI 6492-16, 2007.

[8] - ROMANI, Eduardo. Avaliação de qualidade de vídeo utilizando modelo de atenção baseado em saliência. Universidade Tecnológica do Paraná, Curitiba, 2015.

[9] - MARTINEZ, Herald Becerra. Métricas de qualidade para sinais áudio-visuais. Universidade de Brasília - departamento de ciência da computação, Brasília, 2013.

[10] - MARZILIANO, P. et al. Perceptual blur and ringing metrics: application to JPEG2000. Signal processing: image communication, Suíça, n. 19, p. 163-172, 2004.

[11] - Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Processing Letters* , , vol. 9, no. 3, pp. 81-84, March, 2002.

[12] - Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity 112. (Winner of the IEEE Signal Processing Society Best Paper Award for 2009)," *IEEE Transactions on Image Processing* , vol. 13, no. 4, pp. 600 - 612, April, 2004.

[13] - D., Niranjana; KARAM, Narvekar; KARAM, Lina. A No-Reference imageblur metric based on the cumulative probability of blur detection (CPBD). *IEEE transactions on image processing*, [S.L], v. 20, n. 9, p. 2678-2683, set. 2011.

[14] - HEINEN, Milton Roberto; ENGEL, Paulo Martins. Modelo de Atenção visual seletiva. *Anais do VII Simpósio de Informática da Região Centro do RS, Santa Maria - RS*, jun. 2008.

[15] - OPREA, Cristina C.; PREDA, Radu O.. Using Visual Attention in Video Quality Assessment. *INNOV 2013 : The Second International Conference on Communications, Computation, Networks and Technologies*, Bucharest, Romania, p. 18-22, nov. 2013.

[16] - PERAZZI, F. et al. Saliency Filters: Contrast Based Filtering for Salient Region Detection. *Conference on Computer Vision and Pattern Recognition, EUA*, jun. 2012.

[17] - Ming-Ming Cheng et al. Global contrast based salient region detection, *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, p.409-416, June 20-25, 2011

[18] - ZHANG, Jianming; SLAROFF, Stan. Saliency Detection: A Boolean Map Approach. *IEEE International Conference on Computer Vision, EUA*, dez. 2013.

[19] - V. Mazza, M. Turatto, and C. Umiltà. Foreground-background segmentation and attention: A change blindness study. *Psychological Research*, 69(3):201-210, 2005.

[20] - E. Rubin. Figure and ground. *Readings in Perception*, pages 194-203, 1958.

[21] - W. Eihhauser and P. König, "Does luminance-contrast contribute to a saliency map for overt visual attention?" *European Journal of Neuroscience*, vol. 17, pp. 1089-1097, 2003.

[22] - ITTI, Laurent; KOCH, Christof. COMPUTATIONAL MODELLING OF VISUAL ATTENTION. NATURE REVIEWS NEUROSCIENCE, [S.L], v. 2, p. 194-203, mar. 2001.

[23] - D. H. Hubel and T. N. Wiesel, "Early exploration of the visual cortex," Neuron, vol. 20, no. 3, pp. 401–412, 1998.

[24] - CAMBRIDGE IN COLOUR. Compreendendo a nitidez. Disponível em: <<http://www.cambridgeincolour.com/pt-br/tutoriais/sharpness.htm>>. Acesso em: 02 nov. 2017.

Apêndices

APÊNDICE A – Perceptual Blur

Este apêndice contém o código fonte fornecido pelo autor em Frederique Crete *et al.* (2007).

```
function blur = blurMetric(imagem_original, imagem_borrada, mediablur)
f = imread(imagem_original);
b = imread(imagem_borrada);
[y x] = size(f);
D_F_Ver = abs(f(:,1:x-1) - f(:,2:x));
D_F_Hor = abs(f(1:y-1,:) - f(2:y,:));
D_S_Ver = abs(b(:,1:x-1)-b(:,2:x));
D_S_Hor = abs(b(1:y-1,:)-b(2:y,:));
T_Ver = D_F_Ver - D_S_Ver;
T_Hor = D_F_Hor - D_S_Hor;
V_Ver = max(0,T_Ver);
V_Hor = max(0,T_Hor);
S_D_Ver = sum(sum(D_F_Ver(2:y-1,2:x-1)));
S_D_Hor = sum(sum(D_F_Hor(2:y-1,2:x-1)));
S_V_Ver = sum(sum(V_Ver(2:y-1,2:x-1)));
S_V_Hor = sum(sum(V_Hor(2:y-1,2:x-1)));
blur_F_Ver = (S_D_Ver-S_V_Ver)/S_D_Ver;
blur_F_Hor = (S_D_Hor-S_V_Hor)/S_D_Hor;
blur = max(blur_F_Ver,blur_F_Hor)
blurstr = num2str(blur);
mediablur = blurstr;
endfunction
```

APÊNDICE B – SSIM

Este apêndice contém a implementação da métrica de qualidade SSIM apresentada em Z. Wang, *et al.* (2004).

```
function [mssim, ssim_map] = ssim(img1_filename, img2_filename, K, window, L)

% =====
% SSIM Index with automatic downsampling, Version 1.0
% Copyright(c) 2009 Zhou Wang
% All Rights Reserved.
%
% -----
% Permission to use, copy, or modify this software and its documentation
% for educational and research purposes only and without fee is hereby
% granted, provided that this copyright notice and the original authors'
% names appear on all copies and supporting documentation. This program
% shall not be used, rewritten, or adapted as the basis of a commercial
% software or hardware product without first obtaining permission of the
% authors. The authors make no representations about the suitability of
% this software for any purpose. It is provided "as is" without express
% or implied warranty.
%-----
%
% This is an implementation of the algorithm for calculating the
% Structural SIMilarity (SSIM) index between two images
%
% Please refer to the following paper and the website with suggested usage
%
% Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image
% quality assessment: From error visibility to structural similarity,"
% IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612,
% Apr. 2004.
%
% http://www.ece.uwaterloo.ca/~z70wang/research/ssim/
%
% Note: This program is different from ssim_index.m, where no automatic
```

```
% downsampling is performed. (downsampling was done in the above paper
% and was described as suggested usage in the above website.)
%
% Kindly report any suggestions or corrections to [email protected]
%
%-----
%
%Input : (1) img1: the first image being compared
%        (2) img2: the second image being compared
%        (3) K: constants in the SSIM index formula (see the above
%            reference). default value: K = [0.01 0.03]
%        (4) window: local window for statistics (see the above
%            reference). default window is Gaussian given by
%            window = fspecial('gaussian', 11, 1.5);
%        (5) L: dynamic range of the images. default: L = 255
%
%Output: (1) mssim: the mean SSIM index value between 2 images.
%          If one of the images being compared is regarded as
%          perfect quality, then mssim can be considered as the
%          quality measure of the other image.
%          If img1 = img2, then mssim = 1.
%          (2) ssim_map: the SSIM index map of the test image. The map
%          has a smaller size than the input images. The actual size
%          depends on the window size and the downsampling factor.
%
%Basic Usage:
%   Given 2 test images img1 and img2, whose dynamic range is 0-255
%
%   [mssim, ssim_map] = ssim(img1, img2);
%
%Advanced Usage:
%   User defined parameters. For example
%
%   K = [0.05 0.05];
%   window = ones(8);
%   L = 100;
%   [mssim, ssim_map] = ssim(img1, img2, K, window, L);
%
%Visualize the results:
```

```

%
%  mssim                      %Gives the mssim value
%  imshow(max(0, ssim_map).^4) %Shows the SSIM index map
%=====

img1 = imread(img1_filename);
img2 = imread(img2_filename);

if (nargin < 2 || nargin > 5)
    mssim = -Inf;
    ssim_map = -Inf;
    return;
end

if (size(img1) ~= size(img2))
    mssim = -Inf;
    ssim_map = -Inf;
    return;
end

[M N] = size(img1);

if (nargin == 2)
    if ((M < 11) || (N < 11))
        mssim = -Inf;
        ssim_map = -Inf;
        return
    end
    window = fspecial('gaussian', 11, 1.5);    %
    K(1) = 0.01;                               % default settings
    K(2) = 0.03;                               %
    L = 255;                                   %
end

if (nargin == 3)
    if ((M < 11) || (N < 11))
        mssim = -Inf;
        ssim_map = -Inf;
        return
    end
end

```

```
end
window = fspecial('gaussian', 11, 1.5);
L = 255;
if (length(K) == 2)
    if (K(1) < 0 || K(2) < 0)
        mssim = -Inf;
        ssim_map = -Inf;
        return;
    end
else
    mssim = -Inf;
    ssim_map = -Inf;
    return;
end
end

if (nargin == 4)
    [H W] = size(window);
    if ((H*W) < 4 || (H > M) || (W > N))
        mssim = -Inf;
        ssim_map = -Inf;
        return
    end
    L = 255;
    if (length(K) == 2)
        if (K(1) < 0 || K(2) < 0)
            mssim = -Inf;
            ssim_map = -Inf;
            return;
        end
    else
        mssim = -Inf;
        ssim_map = -Inf;
        return;
    end
end

if (nargin == 5)
    [H W] = size(window);
```

```
if ((H*W) < 4 || (H > M) || (W > N))
    mssim = -Inf;
    ssim_map = -Inf;
    return
end
if (length(K) == 2)
    if (K(1) < 0 || K(2) < 0)
        mssim = -Inf;
        ssim_map = -Inf;
        return;
    end
else
    mssim = -Inf;
    ssim_map = -Inf;
    return;
end
end

img1 = double(img1);
img2 = double(img2);

% automatic downsampling
f = max(1,round(min(M,N)/256));
%downsampling by f
%use a simple low-pass filter
if(f>1)
    lpf = ones(f,f);
    lpf = lpf/sum(lpf(:));
    img1 = imfilter(img1,lpf,'symmetric','same');
    img2 = imfilter(img2,lpf,'symmetric','same');

    img1 = img1(1:f:end,1:f:end);
    img2 = img2(1:f:end,1:f:end);
end

C1 = (K(1)*L)^2;
C2 = (K(2)*L)^2;
window = window/sum(sum(window));
```

```
mu1    = filter2(window, img1, 'valid');
mu2    = filter2(window, img2, 'valid');
mu1_sq = mu1.*mu1;
mu2_sq = mu2.*mu2;
mu1_mu2 = mu1.*mu2;
sigma1_sq = filter2(window, img1.*img1, 'valid') - mu1_sq;
sigma2_sq = filter2(window, img2.*img2, 'valid') - mu2_sq;
sigma12 = filter2(window, img1.*img2, 'valid') - mu1_mu2;

if (C1 > 0 && C2 > 0)
    ssim_map = ((2*mu1_mu2 + C1).*(2*sigma12 + C2))./((mu1_sq + mu2_sq + C1).*(sigma1_sq + sigma2_sq + C2));
else
    numerator1 = 2*mu1_mu2 + C1;
    numerator2 = 2*sigma12 + C2;
    denominator1 = mu1_sq + mu2_sq + C1;
    denominator2 = sigma1_sq + sigma2_sq + C2;
    ssim_map = ones(size(mu1));
    index = (denominator1.*denominator2 > 0);
    ssim_map(index) = (numerator1(index).*numerator2(index))./(denominator1(index).*denominator2(index));
    index = (denominator1 ~= 0) & (denominator2 == 0);
    ssim_map(index) = numerator1(index)./denominator1(index);
end

mssim = mean2(ssim_map);

return
```

APÊNDICE C – Perceptual Sharpness Index

Neste apêndice está contido o código fornecido pelos autores da métrica proposta em FEICHTENHOFER, Christoph *et al.* (2013).

```
function s = PSI( I, percentile )
% PSI(I) determines the perceptual sharpness s of an image I
%
%Reference:
%
% C Feichtenhofer, H Fassold, P Schallauer
% "A perceptual image sharpness metric based on local edge gradient
% analysis", IEEE Signal Processing Letters, 20 (4), 379-382, 2013
%
%
% Written by Christoph Feichtenhofer (cfeichtenhofer AT gmail.com)
% feichtenhofer.github.io
%

%% Parameters
if nargin < 2
    percentile=22; % percentage of blocks to use for metric
end
% BLOCKSIZE [Def:32] Size for averaging of width measurements
% THRESHOLD_W [Def:2] Sum of widths in block to process block further
BLOCKSIZE = 32;
THRESHOLD_W = 2;

if ( size(I,3) > 1 )
    I = rgb2gray(I);
end

% sobel_tr [Def: []] If value is assigned, this is the constant sobel threshold,
sobel_tr = [];
[edges] = edge(I,'Sobel',sobel_tr);

I = double(I) / 255;
```

```

QUOTA_W = percentile/100;

row_blocks = floor(size(I,1)/BLOCKSIZE);
col_blocks = floor(size(I,2)/BLOCKSIZE);

% calculate angles and round them, then calc. horz/vert widths.
[m, n] = size(I);
edge_widths = zeros(m,n);
widths_count = 0;

% calculate gradient
Ix = [I(:,2)-I(:,1), 0.5*(I(:,3:end)-I(:,1:end-2)), I(:,end)-I(:,end-1)];
Iy = [I(2,:)-I(1,:); 0.5*(I(3:end,:)-I(1:end-2,:)); I(end,:)-I(end-1,:)];

%% calculate gradient angle
phi = atan2(Iy,Ix)*180/pi;

%% calculate length for horizontal / vertical edges
t = 8; %angle tolerance t
w_JNB = 3;
[row_idx, col_idx] = find(edges);
for k=1:length(row_idx)
    i = row_idx(k);
    j = col_idx(k);
    width_up=0; width_down=0;
    if (Ix(i,j) == 0 && Iy(i,j) == 0)
        continue; % not really necessary
    end
    %% check for horizontal edge, gradient pointing upwards -> ~ 90°, ~ -270°
    if( abs(phi(i,j)+90) < t )
        min = 0;
        max = 0;
        for d = 1:m
            up = i-d;
            if (up < 1)
                width_up = - 1;
                break;
            end

```

```

        if( I(up,j) <= I(up+1,j) ) %up+1 is max
            width_up = d - 1;
            max = I(up+1,j);
            break;
        end
    end
end
for d = 1:m
    down = i+d;
    if (down > m)
        width_down = - 1;
        break;
    end
    if( I(down,j) >= I(down-1,j) ) % down-1 is min
        width_down = d - 1;
        min = I(down-1,j);
        break;
    end
end
end
if(width_up ~= -1 && width_down ~= -1)
    widths_count = widths_count+1;
    phi2 = (phi(i,j)+90)*pi/180;
    edge_widths(i,j) = (width_up+width_down)/cos(phi2);
    slope = (max-min) / edge_widths(i,j);
    if (edge_widths(i,j) >= w_JNB)
        edge_widths(i,j) = edge_widths(i,j) - slope;
    end
end
end
end

%% check for horizontal edge, gradient pointing downwards - -> ~ -90°, ~ 270°
if( abs(phi(i,j)-90) < t )
    min = 0;
    max = 0;
    for d = 1:m
        up = i-d;
        if (up < 1)
            width_up = - 1;
            break;
        end
    end
end

```

```

        if( I(up,j) >= I(up+1,j) ) % up+1 is min
            width_up = d - 1;
            min = I(up+1,j);
            break;
        end
    end
end
for d = 1:m
    down = i+d;
    if (down > m)
        width_down = - 1;
        break;
    end
    if( I(down,j) <= I(down-1,j) ) %down-1 is max
        width_down = d - 1;
        max = I(down-1,j);
        break;
    end
end
if(width_up ~= -1 && width_down ~= -1)
    widths_count = widths_count+1;
    phi2 = (phi(i,j)-90) *pi/180;
    edge_widths(i,j) = (width_up+width_down)/cos(phi2);
    slope = (max-min) / edge_widths(i,j);
    if (edge_widths(i,j) >= w_JNB)
        edge_widths(i,j) = edge_widths(i,j) - slope;
    end
end
end
end

end

%% calculate average edge widths in each block
avg_w = zeros(row_blocks,col_blocks);
for i=2:row_blocks-1 % skipping image borders
    for j=2:col_blocks-1
        block_row = (i-1)*BLOCKSIZE;
        block_col = (j-1)*BLOCKSIZE;
        block_widths = edge_widths(block_row+1:block_row+BLOCKSIZE,block_col+1:blo
w_sum = sum(sum(block_widths));
        if ( w_sum >= THRESHOLD_W ) % enough widths found

```

```
        % calculate average width of the whole block
        avg_w(i,j) = w_sum / (sum(sum(block_widths ~= 0)));
    end
end
end
avg_w = avg_w(avg_w ~= 0);
avg_w = avg_w(:);
nr_of_used_blocks = ceil( numel(avg_w) * QUOTA_W );
if (nr_of_used_blocks == 0)
    s = 0;
    return;
end

avg_sorted = sort(avg_w);
sharpest_edges = avg_sorted(1:nr_of_used_blocks);

if (widths_count == 0)
    s=0;
else
    s = numel(sharpest_edges) / sum(sharpest_edges);
end

end
```