

Universidade Federal do ABC
Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas
Curso de Engenharia de Informação

João Laio Paz de Melo Soares

Simulação e síntese de imagens com Faixa Dinâmica Larga (HDR)

Trabalho de Graduação III



Santo André – SP

Novembro de 2017

Simulação e síntese de imagens com Faixa Dinâmica Larga (HDR)

João Laio Paz de Melo Soares

Relatório submetido como requisito parcial para obtenção grau de
bacharel em Engenharia de Informação

Orientado por Prof. Dr. Cláudio José Bordin Junior



Universidade Federal do ABC

Santo André – SP

Novembro de 2017

SUMÁRIO

1 INTRODUÇÃO	5
1.1 Tema de Estudo	5
1.2 <i>Tone Mapping</i>	5
1.3 Objetivos	8
2 IMAGENS HDR	9
2.1 Síntese de imagens HDR	9
2.1.1 Construindo o <i>High Dynamic Range Radiance Map</i>	14
2.1.2 Funções de Ponderação	15
3 <i>TONE MAPPING</i>	19
3.1 Operadores de <i>Tone Mapping</i>	19
3.1.1 Curvas de tom (<i>Tone-curve</i>)	20
3.1.2 Adaptação local	21
3.1.3 Algoritmos de <i>Tone Mapping</i>	23
3.1.3.1 Mantiuk	23
3.1.3.1 Fattal	26
3.1.3.1 Drago	26
3.1.3.1 Durand	28
3.1.3.1 Reinhard	28
3.1.3.1 Ashikhmin	31
3.1.3.1 Pattanaik	31

4 PARTE EXPERIMENTAL	33
4.1 Descrição do Experimento	31
5 CONCLUSÃO	58
6 REFERÊNCIAS	59

1 INTRODUÇÃO

1.1 Tema de Estudo

A criação de imagens com faixa dinâmica larga (HDR) permite a visualização de uma gama extremamente ampla da iluminação presente em uma cena produzindo, assim, imagens que se assemelham mais ao que vemos com os nossos próprios olhos [1]. Com a popularização das câmeras digitais, que são dotadas de sensores incapazes de capturar corretamente cenas com grandes variações de brilho, despertou-se o interesse na comunidade acadêmica por métodos computacionais para processamento e exibição de imagens HDR. A partir de várias fotografias de uma mesma cena é possível reproduzir uma representação visual mais próxima daquela que o olho humano é capaz de perceber, ou, também produzir imagens de cunho artístico [2].

1.2 *Tone Mapping*

O alcance dinâmico de uma imagem é determinado pela relação entre as intensidades máximas e mínimas de uma imagem. Quando fala-se em *faixa dinâmica larga* (HDR), que é o existente no mundo real e capturado através de uma imagem, ela precisa ser adaptada e reconstruída de tal maneira que um *display*, como os atuais monitores de vídeo que produzem imagens de *baixo alcance dinâmico* (LDR), também sejam capazes de reproduzir imagens HDR.

O *Tone mapping* (TM) consiste em uma técnica de processamento de imagens que realiza o mapeamento de imagens HDR em imagens LDR de tal forma que seja possível exibi-las em *displays* convencionais LDR. [2] O TM é necessário para que seja possível que a grande variedade de luminosidade de um ambiente do mundo real seja transmitida por um *display* LDR; por essa técnica, o intervalo de luminância é

comprimido, produzindo uma imagem que é capaz de despertar a mesma sensação que uma pessoa teria ao olhar para a mesma cena em um ambiente real.

Para a implementação das técnicas de TM são utilizados os chamados *Tone Mapping Operators (TMO's)* ou simplesmente, Operadores de *Tone Mapping* que consistem em algoritmos que realizam operações de transformação em determinadas características de uma imagem. É importante ressaltar que sempre haverá perda de informação quando se aplica um TMO [43]; porém, utilizando-se o algoritmo adequado, preservam-se características da imagem tais como brilho, cor, contraste e alguns detalhes da imagem que poderiam ser perdidos ou destacados, valendo-se de reprodução de cor e compressão de luminância considerando características estatísticas locais ou o sistema visual humano (Figura 1).

Os TMO's são classificados levando-se em conta a maneira como as imagens são processadas. Podem ser classificados em duas categorias: globais ou locais. Os operadores globais basicamente aplicam uma mesma transformação para todos os pixels da imagem [43], sendo de simples implementação e rápidos computacionalmente justamente por mapearem cada pixel tendo como base apenas a intensidade e as características globais da imagem, não importando a posição que o pixel ocupa na imagem como um todo [43]. Já os operadores locais utilizam-se de valores de parâmetros diferentes de compressão, ou diferentes operadores para partes distintas da imagem. Nesse caso são levados em conta os pixels vizinhos de cada pixel de que se pretende realizar o mapeamento, ou seja, pixels de mesma intensidade podem ser mapeados com valores diferentes de acordo com a sua respectiva vizinhança [43], que é determinada de acordo com a distância geométrica e alguma outra semelhança existente entre esses pixels, como por exemplo suas respectivas intensidades. Em resumo, operadores globais são menos custosos

computacionalmente ao passo que operadores locais costumam ter melhor desempenho nos seus resultados, com uma maior preservação de detalhes e características presente em cada região da imagem a ser mapeada.

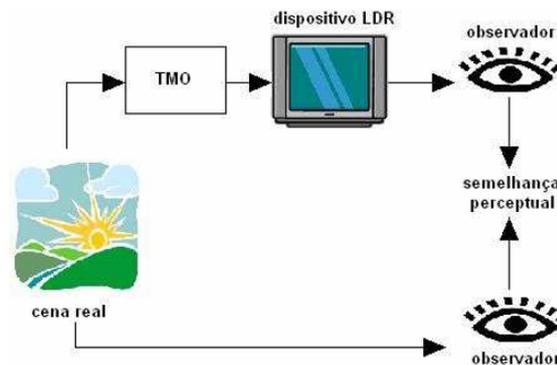


Figura 1: Operador de *Tone Mapping* “ideal” [43].

Uma característica importante que diz respeito ao *Tone Mapping*, que independe de o operador ser global ou local, é o tempo. Se o TMO foi projetado para atuar em uma sequência de imagens (vídeo) considerando adaptação ao Sistema Visual Humano (HVS) ao longo do tempo, ele é chamado de TMO dependente do tempo. Se foi projetado para atuar em imagens estáticas é denominado de TMO independente do tempo [42].

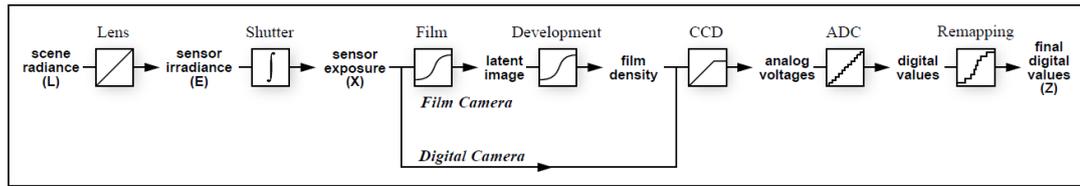


Figura 2 - *Pipeline* de aquisição de imagens - mostra como a radiância da cena se torna valores de pixel tanto para filmes como para câmeras digitais. Mapeamentos não-lineares desconhecidos podem ocorrer durante a exposição, desenvolvimento, digitalização e remapeamento. O algoritmo de Debevec, descrito neste trabalho, determina o mapeamento agregado da radiação de cena L para valores de pixels Z de um conjunto de imagens com diferentes valores de exposição [44].

1.3 Objetivos

Neste trabalho serão descritas todas as etapas realizadas para se gerar imagens que passaram pelo processo de *Tone mapping*, desde a síntese das imagens HDR a partir de várias fotos com exposições diferentes utilizando o algoritmo de Debevec, que será descrito no Capítulo 2. No Capítulo 3 serão descritos os algoritmos de *tone mapping*, e suas particularidades no processamento das imagens e finalmente no Capítulo 4 será descrito o experimento computacional realizado com imagens adquiridas em ambientes variados e processadas com os algoritmos estudados. Ao final do experimento também foi realizada pesquisa com observadores a fim de se avaliar subjetivamente o resultado do processamento dos algoritmos.

2 IMAGENS HDR

Neste capítulo serão detalhadas as etapas para a criação de imagens HDR que passarão pelo processo de *Tone mapping*.

2.1 Síntese de imagens HDR

Para sintetizar imagens HDR é necessária uma determinada quantidade de fotos de um mesmo ângulo, preferencialmente, obtidas com diferentes valores de exposição. De maneira geral, a exposição é a quantidade de fótons que um filme ou sensor fotográfico irá captar.

Para a síntese das imagens HDR foi utilizado o algoritmo de Debevec [44], que têm como objetivo consolidar em uma só imagem todas as outras com diferentes valores de exposição, do mais baixo ao mais alto, e desta forma gerar uma única imagem com um grande intervalo de exposição. Como não é possível visualizar essa imagem em monitores convencionais, faz-se necessário o uso de técnicas de *Tone mapping*, que serão descritas posteriormente neste trabalho, para aproximar a visualização de uma imagem HDR em monitores convencionais LDR.

O algoritmo de Debevec [44] é baseado na exploração de uma propriedade física presente em sistemas de captação de imagem, tanto fotoquímicos quanto eletrônicos, conhecida como *reciprocidade*. Consideremos primeiramente o filme fotográfico. A resposta de um filme fotográfico a variações na exposição é resumida através de uma curva característica (ou curva de Hurter-Driffield). Este é um gráfico da densidade óptica D do filme processado pelo logaritmo da exposição X para o qual foi submetido. A exposição X é definida como o produto da irradiância E no filme e tempo de exposição

Δt , tendo como unidades Jm^{-2} . A chave para o próprio conceito da curva característica é a suposição de que apenas o produto $E \cdot \Delta t$ é importante, que diminuindo E pela metade e dobrando o valor de Δt não se altera o valor da densidade óptica D resultante. Em condições extremas (Δt muito grande ou muito pequeno), a hipótese da reciprocidade pode não funcionar, o que origina uma situação descrita como falha de reciprocidade. Em filmes fotográficos típicos, a reciprocidade se mantém dentro de $\frac{1}{3}$ de stop para tempos de exposição de 10 segundos a 1/10.000 segundo. 1 stop é um termo fotográfico para um fator de dois; $\frac{1}{3}$ stops são, portanto, $2^{\frac{1}{3}}$.

Uma faixa dinâmica ainda maior pode ser coberta usando filtros de densidade neutra para diminuir a quantidade de luz que atinge a película para um determinado tempo de exposição.

Após o desenvolvimento dos processos de escaneamento e digitalização, obtém-se agora um número digital Z , que é uma função não-linear da exposição X original do pixel. Chama-se essa função de f , que é a composição da curva característica do filme, bem como todas as não linearidades introduzidas pelas etapas de processamento posteriores. O primeiro objetivo é recuperar esta função f . Uma vez recuperada, pode-se calcular a exposição X em cada pixel através da função $X = f^{-1}(Z)$. É feita uma razoável suposição de que a função f aumenta monotonicamente, de modo que o inverso f^{-1} seja bem definido. Conhecendo a exposição X e o tempo de exposição Δt , a irradiância E é recuperada através da equação $E = X/\Delta t$, que é proporcional à radiância L na cena. L é proporcional a E para qualquer pixel particular, mas é possível que o fator de proporcionalidade seja diferente em diferentes locais do sensor. Uma fórmula para esta variação é $E = L \frac{\pi}{4} \left(\frac{d}{f}\right)^2 \cos^4 \alpha$, onde α mede o ângulo do pixel do eixo óptico da lente. No entanto, as mais modernas lentes de câmera são

projetadas para compensar esse efeito e fornecer um mapeamento quase constante entre radiação e irradiância em f/8 e aberturas menores.

Antes de prosseguir, deve-se discutir as consequências da resposta espectral do sensor da câmera. A exposição X deve ser pensada como uma função de comprimento de onda $X(\lambda)$, e a abscissa na curva característica deve ser a integral $\int X(\lambda)R(\lambda) d\lambda$ onde $R(\lambda)$ é a resposta espectral do elemento de detecção na localização do pixel.

A entrada para o algoritmo é uma série de fotografias digitalizadas tiradas de uma mesma posição com diferentes exposições conhecidas de duração Δt_j . Assumiremos que a cena é estática e que o processo é concluído rápido o suficiente para que as mudanças de iluminação possam ser seguramente ignoradas. Pode-se então assumir que os valores da irradiância do filme E_i para cada pixel i são constantes. Denotaremos valores de pixel por Z_{ij} onde i é um índice espacial sobre os pixels e os índices j são sobre a exposição multiplicado por Δt_j . Pode-se agora escrever a equação de reciprocidade do filme como:

$$Z_{ij} = f(E_i \Delta t_j) \quad (1)$$

Como se assume que f é monotônica, esta função é inversível e pode-se reescrever (1) como:

$$f^{-1}(Z_{ij}) = E_i \Delta t_j$$

Tomando o logaritmo natural de ambos os lados, temos:

$$\ln f^{-1}(Z_{ij}) = \ln E_i + \ln \Delta t_j$$

Para simplificar a notação, define-se a função $g = \ln f^{-1}$. Dessa forma, tem-se o seguinte conjunto de equações:

$$g(Z_{ij}) = \ln E_i + \ln \Delta t_j \quad (2)$$

Onde i varia ao longo dos pixels e j varia ao longo das durações de exposição.

Neste conjunto de equações, os Z_{ij} são conhecidos, assim como o Δt_j . As incógnitas são as irradiações E_i , bem como a função g , embora foi assumido que g é suave e monotônico.

Deseja-se recuperar a função g e as irradiações E_i que melhor satisfazem o conjunto de equações que surgem da Equação 2 em uma função de mínimos quadrados. Pode-se observar que a recuperação de g exige apenas recuperar o número finito de valores que $g(Z)$ pode tomar desde que o domínio de Z (valores de brilho dos pixels) seja finito. Tomando Z_{min} e Z_{max} como o menor e maior valores de pixel (inteiros), respectivamente, N sendo o número de locais de pixels e P o número de fotografias, formula-se o problema para encontrar os valores $(Z_{max} - Z_{min} + 1)$ de $g(Z)$ e os valores N de $\ln E_i$ que minimizam a seguinte função quadrática objetiva:

$$\mathcal{Q} = \sum_{i=1}^N \sum_{j=1}^P [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]^2 + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} g''(z)^2 \quad (3)$$

O primeiro termo garante que a solução satisfaça o conjunto de equações decorrente da Equação 2 em um sentido de mínimos quadrados. O segundo termo é um termo de suavidade na soma de valores quadrados da segunda derivada de g para garantir que a função g seja suave; nessa configuração discreta, utiliza-se $g''(z) = g(z-1) - 2g(z) + g(z+1)$. Este termo de suavidade é essencial para a formulação, na medida em que fornece acoplamento entre os valores $g(z)$ na minimização. O escalar λ pondera o termo de suavidade em relação ao termo de ajuste de dados, e deve ser escolhido adequadamente para a quantidade de ruído esperado nas medidas de Z_{ij} .

Por ser quadrático nos E_i 's e $g(z)$'s, minimizando \mathcal{Q} é um problema diretamente linear de mínimos quadrados. O sistema de equações lineares pré-determinado é resolvido de forma robusta usando o método de decomposição em valores singulares (SVD).

Necessitam-se cobrir três pontos adicionais para completar a descrição do algoritmo:

Primeiro, a solução para os valores $g(z)$ e E_i só pode ser até um único fator de escala α . Se cada valor logarítmico de irradiância $\ln E_i$ fosse substituído por $\ln E_i + \alpha$, e a função g substituída por $g + \alpha$, o sistema de 2 equações e também a função objetiva \mathcal{Q} permaneceriam inalteradas. Para estabelecer um fator de escala, apresenta-se a restrição adicional $g(Z_{méd}) = 0$, onde $Z_{méd} = \frac{1}{2}(Z_{min} + Z_{max})$, simplesmente adicionando-a como uma equação no sistema linear. O significado dessa restrição é que um pixel com um valor entre Z_{min} e Z_{max} assumirá uma exposição unitária.

Em segundo lugar, a solução pode ser feita para ter um ajuste muito melhor ao antecipar a curva básica da função de resposta. Como $g(z)$ geralmente tem uma inclinação íngreme perto de Z_{min} e Z_{max} , deve-se esperar que $g(z)$ será menos suave e corresponderá mais aos dados próximos a esses extremos. Para reconhecer isso, pode-se introduzir uma função de ponderação $w(z)$ para enfatizar a suavidade e os termos adequados para o meio da curva. Uma escolha sensata de w é uma função “chapéu” simples:

$$w(z) = \begin{cases} z - Z_{min} & \text{para } z \leq \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{max} - z & \text{para } z > \frac{1}{2}(Z_{min} + Z_{max}) \end{cases} \quad (4)$$

A Equação 3 agora se torna:

$$\mathcal{Q} = \sum_{i=1}^N \sum_{j=1}^P \{w(Z_{ij})[g(Z_{ij}) - \ln E_i - \ln \Delta t_j]\}^2 + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} [w(z)g''(z)]^2$$

Finalmente, não é necessário usar todos os pixels disponíveis nesta solução. Dadas medidas de N pixels em P fotografias, temos que resolver N valores de $\ln E_i$ e $Z_{max} - Z_{min}$ amostras de g . Para garantir um sistema suficientemente dependente, queremos $N(P - 1) > (Z_{max} - Z_{min})$. Para o intervalo de valores de pixels ($Z_{max} - Z_{min}$) = 255, $P = 11$ fotografias, uma escolha de N na ordem de 50 pixels é mais do que adequado. Como o tamanho do sistema de equações lineares decorrentes da Equação 3 está na ordem de $N \times P + Z_{max} - Z_{min}$, considerações de complexidade computacional tornam impossível utilizar cada pixel neste algoritmo. Claramente, os pixels devem ser escolhidos de forma que eles tenham uma distribuição razoavelmente uniforme de valores entre Z_{min} e Z_{max} e para que eles estejam espacialmente bem distribuídos na imagem. Além disso, os pixels são mais bem amostrados das regiões da imagem com variância menor para que o brilho possa ser assumido como constante em toda a área do pixel, de modo que o efeito do borrão óptico do sistema de imagem seja minimizado.

Observe que não foi aplicada explicitamente a restrição de que g deve ser uma função monotônica. Se desejável, isso pode ser feito transformando o problema para um problema de mínimos quadrados não-negativos.

2.1.1 Construindo o *High Dynamic Range Radiance Map*

Uma vez que a curva de resposta g é recuperada, ela pode ser usada para converter rapidamente valores de pixels para valores de radiância (brilho) relativos, assumindo que a exposição Δt_j seja conhecida. Pode-se observar que a curva pode

ser usada para determinar os valores de brilho em qualquer imagem adquirida pelo processamento de imagem associado com g , e não apenas para as imagens usadas para recuperar a função de resposta.

Da Equação 2, obtém-se:

$$\ln E_i = g(Z_{ij}) + \ln \Delta t_j \quad (5)$$

Para a robustez e para recuperar valores de radiância de alta faixa dinâmica, devem-se usar todas as exposições disponíveis para um determinado pixel para calcular a sua radiância. Para isso, reutiliza-se a função de ponderação da Equação 4 para dar maior peso às exposições em que o valor do pixel esteja mais próximo do meio da função de resposta:

$$\ln E_i = \frac{\sum_{j=1}^P w(Z_{ij}) [g(Z_{ij}) - \ln \Delta t_j]}{\sum_{j=1}^P w(Z_{ij})} \quad (6)$$

Combinar as múltiplas exposições tem o efeito de reduzir ruído nos valores de radiância recuperados. Também reduz os efeitos de artefatos de imagem, como o efeito de granulação de filme. Como a função de ponderação ignora valores de pixels saturados, os artefatos *blooming* têm pouco impacto sobre os valores de radiância reconstruídos. O efeito *blooming* ocorre quando carga ou luz em locais altamente saturados na superfície da imagem “derrama” e afeta valores em pontos vizinhos.

2.1.2 Funções de ponderação

Uma função de ponderação é utilizada para escolher de maneira adequada a região a ser extraída para a composição da imagem HDR em uma fotografia. Por exemplo, considerando-se a cena de uma sala com uma janela exibindo um dia claro do lado de fora, é desejável que se extraia a cena de fora da janela deliberadamente

da imagem subexposta, e o interior da sala da imagem superexposta. A escolha de uma função de ponderação adequada é então crucial. Os resultados podem ser alcançados usando diferentes funções que mostram melhora ou piora dependendo das condições particulares de iluminação da cena. Para obter uma imagem final com um certo nível de cinza médio m é possível utilizar uma função gaussiana de ponderação na forma:

$$w(x) = e^{-\left(\frac{(x-m)^2}{\sigma^2}\right)}$$

onde σ é o desvio padrão da distribuição. Alternativamente, também é possível usar uma função triangular centrado em m , dado por:

$$w(x) = \begin{cases} (x - x_{min}) / (m - x_{min}) & \text{se } x \leq m \\ (x_{max} - x) / (x_{max} - m) & \text{se } x > m \end{cases}$$

onde geralmente $x_{min} = 0$ e $x_{max} = 255$. Ambas as funções têm o efeito de ponderar mais um pixel perto do valor m . Em vez de usar uma única função de ponderação para todos os quadros, também é possível aplicar uma abordagem diferente (gaussiano ou triangular) com base na centralização de cada função em valores diferentes. Uma possível escolha poderia ser centralizar cada função em:

$$m_i = x_{max} - mean_i \quad i = 1, 2, \dots, N$$

onde $mean_i$ é o nível médio de cinza da imagem i e N é o número de quadros de entrada. Outras opções possíveis para as funções de ponderação são o uso das derivadas da função de resposta da câmera, que muda adequadamente dependendo do nível de exposição dos quadros como descrito em [10]. Neste caso, um pixel é mais ponderado se a região correspondente da função de resposta é mais sensível

às mudanças na exposição. Dois exemplos funções de ponderação são mostradas na Figura 1. O valor final do pixel na imagem mesclada será dada por:

$$I(x, y) = \frac{\sum_{i=1}^N w(I_i(x, y)) \cdot I_i(x, y)}{\sum_{i=1}^N w(I_i(x, y))}$$

onde N é o número dos quadros, $I_i(x, y)$ é o valor do pixel em posição (x, y) e $w()$ é a função de ponderação correspondente.

Outra função de ponderação que foi utilizada foi a *Plateau* (Figura 4), que para o caso estudado nesse trabalho, a partir da verificação do código-fonte do Luminance-HDR, constatou-se obedecer a seguinte expressão:

$$w(x) = \begin{cases} y = 0, & \text{se } x < t \text{ ou } x > 1 - t \\ y = 1 - (2x - 1)^{12}, & \text{caso contrário} \end{cases}$$

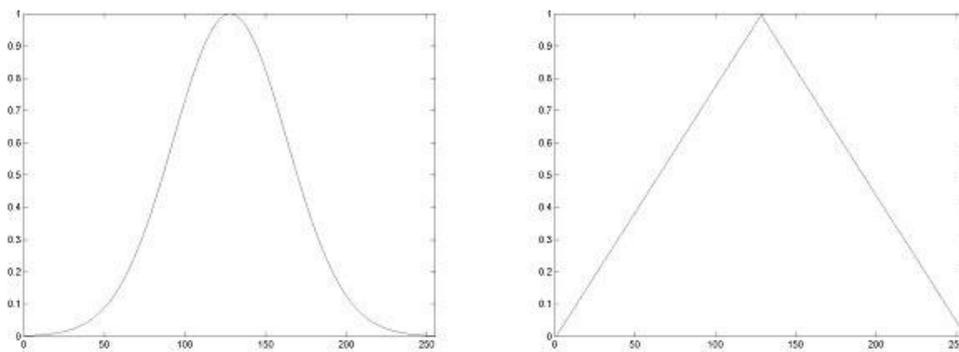


Figura 3 - Dois exemplos de função de ponderação: gaussiana e triangular [45].

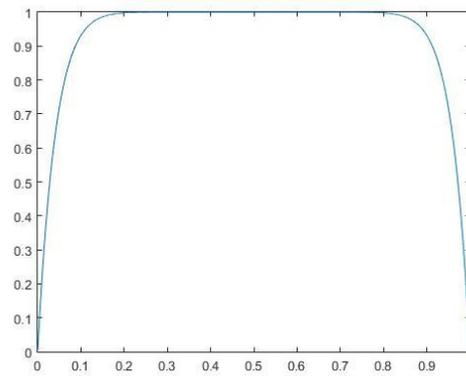


Figura 4 - Gráfico da função *Plateau*.

3 TONE MAPPING

Neste capítulo será descrito o princípio do processo de *Tone Mapping* e explicado os algoritmos estudados neste trabalho.

3.1 Operadores de *Tone Mapping*

O *Tone Mapping* de imagens HDR envolve a transformação de sinais HDR para sinais LDR que podem ser apresentados em *displays* padrão. Desta forma, essas transformações envolvem tipicamente uma função não linear e compressiva. O processo completo de mapeamento de tons pode ser realizado de diversas maneiras, porém uma das formas mais comuns é ilustrada na figura a seguir:

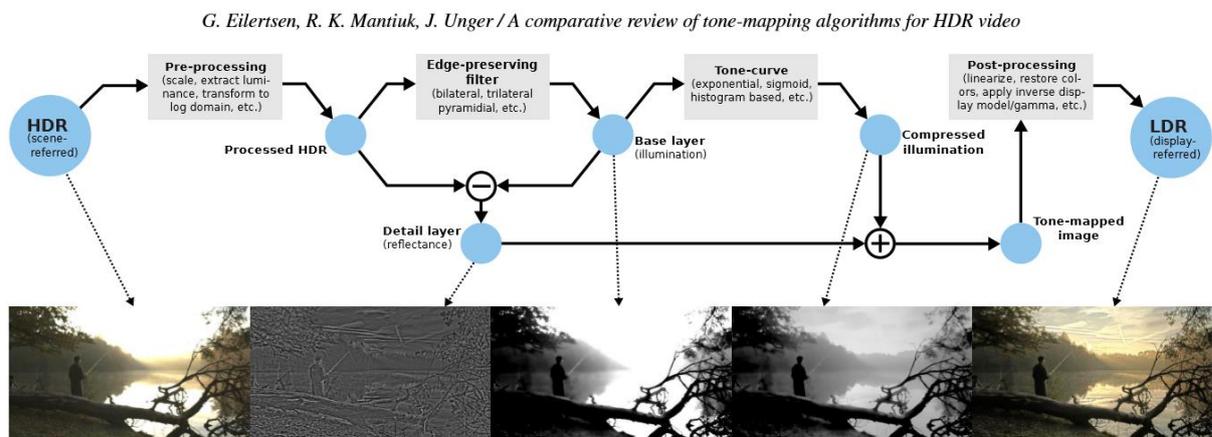


Figura 5 - Operador de Tone Mapping “ideal” [42].

Estas operações transformam a luminância de entrada para o domínio log, onde um filtro de preservação de “cantos” (*edge-preserving filter*) é aplicado. A imagem filtrada é então utilizada para extrair detalhes da imagem, que são mantidos separados da transformação compressiva realizada pela curva de tom. Assim, os detalhes locais são preservados enquanto a faixa dinâmica global é reduzida. Finalmente, o resultado

do mapeamento por tons é transformado de volta ao domínio linear, as cores são restauradas a partir da entrada original e o material é preparado para um dispositivo de exibição pré-determinado.

3.1.1 Curvas de tom (*Tone-curve*)

No núcleo de um algoritmo de mapeamento de tons está a função de mapeamento de um domínio para outro. Na maioria dos casos isso pode ser descrito como uma função contínua - a curva de tom [42].

O objetivo final de um Operador de Tone Mapping (TMO) é mapear a partir da entrada HDR $L \in R^+$ para valores de pixel exibidos no display $l \in Nd = \{0, \dots, 255\}$, usando uma transformação $M: L \rightarrow l$. L que pode ser descrita em unidades arbitrárias, mas para a maioria dos algoritmos de mapeamento de tom é esperado que se esteja linearmente relacionado com os valores de luminância capturados. Enquanto alguns TMOs tomam o mapa diretamente para os valores de pixel $\{0, \dots, 255\}$, outros mapeiam para exibir a luminância T e precisam de um modelo de exibição para transformá-los em valores de pixel. No que se segue, descrevemos a curva de tons como $V: L \rightarrow T$ para uma saída $T \in R^+$. Isso deve ser complementado com uma transformação final para um valor de exibição l (o passo final da Figura 5).

A função de curva de tom V pode depender de um conjunto de constantes $\alpha = \{\alpha_1, \dots, \alpha_K\}$ especificadas pelo usuário, e também um conjunto de estatísticas de imagem $\Phi(L) = \{\Phi_1(L_1, \dots, L_N), \dots, \Phi_K(L_1, \dots, L_N)\}$. As constantes α podem ser, por exemplo, luminosidade geral, contraste, saturação, etc. As estatísticas da imagem podem incluir medidas tais como a média da imagem $\Phi = \mu = \frac{1}{N} \sum_s L_s$, algum

percentil tal como o valor médio, máximo ou mínimo (por exemplo, o brilho, o contraste, a saturação etc. (0.5, 1 e 0 percentis, respectivamente), ou o histograma da imagem $\Phi_i = p_L(L_i)$.

Usando α e Φ , uma formulação geral de um mapeamento espacialmente invariante em uma posição de pixels s pode ser escrita como:

$$T_s = V(L_s; \Phi(L); \alpha) \quad (7)$$

Por exemplo, uma compensação de exposição simples pode ser descrita como $T_s = V(L_s; \alpha) = \alpha L_s$. Outro exemplo é a curva sigmoide, que pode ser formada para corresponder ao valor médio do rendimento, $\Phi = \mu$, para se adaptar à luminância global,

$$T_s = V(L_s; \Phi(L); \alpha) = \frac{L_s^\alpha}{L_s^\alpha + \Phi^\alpha} \quad (8)$$

3.1.2 Adaptação Local

A curva de tons na Equação 7 pode ser formulada para comprimir a gama dinâmica, como na Equação 8, mas isso ocorre com o custo de compressão de todos os contrastes da entrada [42]. O objetivo comum é preservar os contrastes locais - ou detalhes - para imitar as habilidades do SVH (Sistema Visual Humano). Para melhor reproduzir os contrastes locais podemos introduzir um conjunto de parâmetros $\xi_s(L)$ que da mesma forma que $\Phi(L)$ dependem da informação em L , mas ponderada em uma vizinhança local Ω_s em torno de um pixel s ,

$$\xi_s(L) = \xi(L_{\Omega_s}) = \{\xi_1(L_{\Omega_s}), \dots, \xi_{K_\xi}(L_{\Omega_s})\}. \quad (9)$$

Usando $\xi_s(L)$ para capturar propriedades locais em torno de cada pixel s , o mapeamento de tons pode se adaptar espacialmente para manter ou melhorar os detalhes da imagem. A curva de tom local assume agora a forma

$$T_s = V(L_s; \xi_s(L); \Phi(L); \alpha) \quad (10)$$

Um exemplo trivial de uma medida de adaptação local é a média local ponderada, calculada com um kernel gaussiano G_σ , $\xi_s(L) = (G_\sigma * L)_s$. As estatísticas variando espacialmente podem ser usadas diretamente na formulação de uma curva de tom, por exemplo na Equação 8 para se adaptar localmente na imagem em vez de usar a média sobre todos os pixels de imagem, $V(L_s; \Phi(L); \alpha) \rightarrow V(L_s; \xi_s(L); \alpha)$. No entanto, dado um filtro $\xi_s(L)$ que estima as mudanças globais na imagem, uma formulação comum é dividir a imagem em uma camada de base $B_s = \xi_s(L)$ e uma camada de detalhe $D_s = L_s - B_s$. A curva de tom é então aplicada na camada base, enquanto a camada de detalhe é adicionada de volta após o mapeamento,

$$\begin{aligned} T_s &= V(L_s; \xi_s(L); \Phi(L); \alpha) \\ &= \tilde{V}(B_s; \Phi(L); \alpha) + D_s \\ &= \tilde{V}(\xi_s(L); \Phi(L); \alpha) + L_s - \xi_s(L). \end{aligned} \quad (11)$$

Aqui, a curva de tons local V foi decomposta em um mapeamento global das relações globais, e os detalhes D , que são preservados separadamente do mapeamento. Deve-se notar também que essas operações geralmente são executadas no domínio log, mas isso não é explicitamente indicado para simplificar a notação.

Finalmente, podemos também alterar o mapeamento sobre a imagem, $\mathbf{V} = \{V_1, \dots, V_n\}$, e introduzir um conjunto de parâmetros que variam espacialmente $\theta_s =$

$\{\theta_1, \dots, \theta_{k_\theta}\}$ que não dependem do conteúdo da imagem. Estes parâmetros podem, por exemplo, ser formulados em termos das posições de pixel verticais e horizontais na imagem. Obtém-se agora à formulação mais geral de uma curva de tom,

$$T_s = V_s(L_s; \xi_s(L); \theta_s; \Phi(L); \alpha) \quad (12)$$

que é a base do *tone mapping*.

3.1.3 Algoritmos de *Tone Mapping*

Neste trabalho foram estudados os seguintes algoritmos de *Tone mapping*: Mantiuk ('06 e '08), Fattal, Drago, Durand, Reinhard ('02 e '05), Ashikhmin e Pattanaik

3.1.3.1 Mantiuk

Neste algoritmo os valores de luminância dos pixels de uma imagem são primeiramente transformados em valores de contraste físico, que são então transduzidos para os valores de resposta do HVS (*Human Visual System*).

A imagem resultante é então modificada alterando os valores de resposta, que estão intimamente relacionados com uma impressão subjetiva de contraste. Os valores de resposta modificados podem mais tarde ser convertidos de volta para valores de luminância através de processo inverso.

Embora várias definições de contraste tenham sido propostas na literatura, elas são geralmente aplicáveis a apenas um estímulo simples e não especificam como medir o contraste em cenas complexas. Esta questão foi abordada por [54], que percebeu que o processamento de imagens pode ser periódico ou local e, portanto, a representação do contraste em imagens também deve ser quase local. Analisando a analogia das estruturas da retina, ele propôs medir contraste em imagens complexas

como uma diferença entre os níveis selecionados de uma pirâmide gaussiana. No entanto, a diferença resultante de gaussianas leva a uma medida limitada de contraste na banda de passagem, que tende a introduzir artefatos de “halo” em bordas “afiadas” quando é modificado. Para evitar esse problema, apresenta-se uma medida de contraste de baixo risco. Utiliza-se uma relação logarítmica G como a medida do contraste entre dois pixels, o que é conveniente nos cálculos, uma vez que podem ser substituídos pela diferença de logaritmos. Portanto, o contraste passa-baixas é definido como uma diferença entre um pixel e um dos seus vizinhos em um nível particular, k , de uma pirâmide gaussiana, que pode ser escrita como:

$$G_{i,j}^k = \log\left(\frac{L_i^k}{L_j^k}\right) = x_i^k - x_j^k \quad (13)$$

onde L_i^k e L_j^k são valores de luminância para pixels vizinhos i e j . Para um único pixel i , existem duas ou mais medidas de contraste $G_{i,j}^k$, dependendo de quantos pixels vizinhos j forem considerados. Note-se que tanto L quanto x cobrem uma área cada vez maior de uma imagem quando se deslocam para os níveis mais “preenchidos” da pirâmide. Desta forma, a definição de contraste leva em consideração a percepção quase local de contraste, em que os detalhes finos são vistos apenas localmente, enquanto variações nas baixas frequências podem ser avaliadas para toda a imagem. A escolha de quantos pixels vizinhos, x_j , devem ser levados em consideração para cada pixel, x_i , geralmente depende da aplicação e do tipo das imagens. Para operações de mapeamento de tons em imagens complexas, constatou-se que dois vizinhos mais próximos são suficientes. Para outras aplicações, como um mapeamento de cor para cinza, e para imagens que contêm áreas planas (por exemplo, mapas vetoriais), consideram-se de 20 a 30 pixels vizinhos.

A Equação 13 pode ser usada para transformar luminância em contraste. Agora, executa-se a operação inversa que restaura uma imagem dos valores de contraste modificados \hat{G} . O problema é que provavelmente não há imagem que corresponda a esses valores de contraste. Portanto, procura-se uma imagem cujos valores de contraste estejam próximos, mas não necessariamente iguais a \hat{G} . Pode-se conseguir isso pela minimização da distância entre um conjunto de valores de contraste \hat{G} que especifica o contraste desejado, e G , que é o contraste da imagem real. O processo pode ser descrito na seguinte equação:

$$f(x_1^1, x_2^1, \dots, x_N^1) = \sum_{k=1}^K \sum_{i=1}^N \sum_{j \in \Phi_i} p_{i,j}^k (G_{i,j}^k - \hat{G}_{i,j}^k)^2 \quad (14)$$

em relação aos valores de pixel x_i^1 no melhor nível da pirâmide. Φ_i é um conjunto de vizinhos do pixel i , N é o número total de pixels e K é o número de níveis em uma pirâmide gaussiana.

O coeficiente $p_{i,j}^k$ na Equação 14 é um fator de ponderação constante, que pode ser usado para controlar um desajuste entre o contraste desejado e o contraste resultante da solução do problema de otimização. Se o valor desse coeficiente for alto, há uma penalidade maior por uma incompatibilidade entre $G_{i,j}^k$ e $\hat{G}_{i,j}^k$. Embora a escolha desses coeficientes possa depender da aplicação, na maioria dos casos, busca-se penalizar o desajuste do contraste em relação à sensibilidade ao contraste do HVS. Um maior desajuste deve ser permitido para as magnitudes de contraste às quais o olho é menos sensível. Desta forma, a visibilidade dos erros resultantes de tal

incompatibilidade seria igual para todos os valores de contraste. Pode-se alcançar isso assumindo que:

$$p_{i,j}^k = \begin{cases} \Delta G^{-1}(\hat{G}_{i,j}^k) & \text{se } \hat{G}_{i,j}^k \geq 0,001 \\ \Delta G^{-1}(0,001) & \text{caso contrário} \end{cases} \hat{G}_{i,j}^k \quad (15)$$

onde ΔG^{-1} é o inverso da função de discriminação de contraste citada em [50] e a segunda condição evita a divisão em 0 para um contraste muito baixo.

Após a definição do contraste, é aplicada a função de transdução que prediz a resposta hipotética do HVS para um determinado contraste físico e é definida como:

$$T(G) = 54,09288 \cdot G^{0,41850} \quad (16)$$

conforme é demonstrado em [50].

3.1.3.2 Fattal

Observando que mudanças drásticas na luminosidade em uma imagem HDR originam gradientes de luminância de grandes magnitudes, o algoritmo manipula o campo de gradiente de luminância da imagem atenuando as magnitudes de grandes gradientes. Uma nova imagem LDR é então obtida resolvendo uma equação de Poisson no campo de gradiente modificado.

3.1.3.3 Drago

O método baseia-se na compressão logarítmica de valores de luminância, imitando a resposta humana à luz. Uma função de limiar de potência é introduzida para variar de forma adaptativa bases logarítmicas, resultando em boa preservação de detalhes e

contraste. Para melhorar o contraste em áreas escuras, são aplicadas mudanças no procedimento de correção de gama.

Inicialmente, os dados de entrada são convertidos de seu formato original para uma representação em ponto flutuante de valores RGB lineares uma vez que a característica iluminante da cena não é conhecida com precisão na maioria dos casos, assumimos um ponto branco D_{65} , para converter valores tri-estimulo entre Rec.709 RGB e CIE XYZ.

A componente de luminância Y de cada pixel (L_w para luminância “global”) e a luminância máxima da cena L_{wmax} são divididos pela luminância de adaptação “global” L_{wa} e, eventualmente, multiplicados por um fator de exposição definido pelo usuário.

A função de mapeamento de tons apresentada na Equação 17 é usada para calcular um valor de exibição L_d para cada pixel. A equação requer valores de luminância L_w e L_{wmax} que caracterizam a cena, bem como L_{dmax} , que é a capacidade de luminância máxima do meio de exibição. O parâmetro da função *bias* é denotado por b , que é um parâmetro alterável pelo usuário e funciona como uma função de ajuste.

$$L_d = \frac{L_{dmax} \cdot 0,01}{\log(L_{wmax} + 1)} \cdot \frac{\log(L_w + 1)}{\log\left(2 + \left(\frac{L_w}{L_{wmax}}\right)^{\frac{\log(b)}{\log(0,5)}} \cdot 8\right)} \quad (17)$$

L_{dmax} , é usado como um fator de escala para adaptar a saída ao display desejado. No denominador, o logaritmo decimal é usado já que o valor máximo de luminância na cena é sempre reamostrado para o logaritmo decimal pela função *bias*. O parâmetro *bias* b é essencial para ajustar a compressão de altos valores e a visibilidade de detalhes em áreas escuras.

A correção gama deve ser aplicada aos dados mapeados por tons para compensar a não-linearidade dos dispositivos de exibição.

3.1.3.4 Durand

Este algoritmo reduz o contraste, preservando os detalhes. Baseia-se em uma decomposição em duas escalas da imagem em uma camada base, codificando variações em larga escala e em uma camada de detalhes. Apenas a camada base tem seu contraste reduzido, preservando os detalhes, e é obtida utilizando um filtro de preservação de bordas denominado filtro bilateral. Trata-se de um filtro não-linear, onde o peso de cada pixel é calculado usando um Gaussiano no domínio espacial multiplicado por uma função de influência no domínio da intensidade que diminui o peso de pixels com grandes diferenças de intensidade.

3.1.3.5 Reinhard

Este algoritmo é baseado no sistema de Sistema de Zonas (System Zone) [49]. Uma zona é definida como um número romano associado a um intervalo aproximado de luminância em uma cena, bem como um valor aproximado de refletância de uma impressão. Existem onze zonas de impressão, de preto puro (zona 0) a branco puro (zona X), cada um dobrando em intensidade e um número potencialmente maior de zonas de cena (Figura 6).

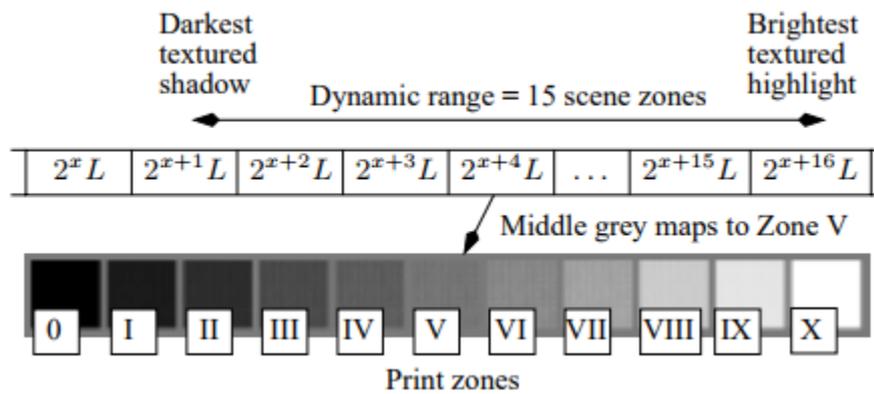


Figura 6 - O mapeamento de zonas de cena para zonas de impressão. As zonas de cena em ambos os extremos serão mapeadas para preto puro (zona 0) ou branco (zona X) se o alcance dinâmico da cena for de onze zonas ou mais.

Como muitos métodos de reprodução de tons [Tumblin e Rushmeier 1993; Ward 1994; Holm 1996], a log-média da luminância é vista como uma aproximação útil à *chave da cena*; a chave de uma cena indica se ela é subjetivamente clara, normal ou escura. Uma sala pintada de branco seria de alta chave (*high-key*), e um estável fraco seria de baixa-chave (*low-key*). Esta quantidade $\overline{L_w}$ é calculada por:

$$\overline{L_w} = \frac{1}{N} \exp(\sum_{x,y} \log(\delta + L_w(x, y))) \quad (18)$$

onde $L_w(x, y)$ é a luminância "mundial" para pixel (x, y) , N é o número total de pixels na imagem e δ é um pequeno valor para evitar a singularidade que ocorre se os pixels pretos estiverem presentes na imagem. Se a cena tiver uma chave normal, é ideal mapear esse resultado para o meio-cinza da imagem exibida, ou 0,18 em uma escala de zero a um. É o que sugere a equação:

$$L(x, y) = \frac{a}{L_w} L_w(x, y) \quad (19)$$

onde $L(x, y)$ é uma luminância escalada e $a = 0,18$. Para baixa-chave ou imagens de alta-chave, permite-se ao usuário mapear a média de log para valores.

O principal problema com a Equação 18 é que muitas cenas têm, em sua maioria, um alcance dinâmico normal, mas possui algumas regiões de alta luminância perto de destaques ou no céu. Na fotografia tradicional, esta questão é tratada pela compressão de altas e baixas luminâncias.

No entanto, a fotografia moderna abandonou essas curvas de transferência em forma de "s" em favor das curvas que comprimem principalmente as altas luminâncias [55;56]. Um simples operador de mapeamento de tons com essas características é dado por:

$$L_d(x, y) = \frac{L(x, y)}{1 + L(x, y)} \quad (20)$$

Nota-se que os altos valores de luminância são dimensionados em aproximadamente $\frac{1}{L}$ enquanto os baixos valores de luminância são dimensionados em 1. Esta formulação é garantida para trazer todas as luminâncias dentro do alcance exibível. No entanto, isso nem sempre é desejável. A Equação 20 pode ser estendida para permitir que altas luminâncias sejam desprezadas (*burned out*) de forma controlável:

$$L_d(x, y) = \frac{L(x, y) \left(1 + \frac{L(x, y)}{L_{white}^2}\right)}{1 + L(x, y)} \quad (20)$$

onde L_{white} é a menor luminância que será mapeada em branco puro. Esta função é uma mistura entre a Equação 20 e um mapeamento linear. Se o valor L_{white} for ajustado para a luminância máxima na cena L_{max} ou superior, não ocorrerá *burn-out*. Se estiver ajustado para infinito, a função retornará à Equação 20. Por padrão, estabelece-se L_{white} para a luminância máxima na cena. Se esse padrão for aplicado a cenas com um baixo alcance dinâmico (ou seja, $L_{max} < 1$), o efeito é um aprimoramento de contraste sutil.

Para muitas imagens HDR, a compressão proporcionada por esta técnica parece ser suficiente para preservar detalhes em áreas de baixo contraste, enquanto comprime altas luminâncias para um intervalo exibível. No entanto, para imagens de faixa dinâmica muito altas, detalhes importantes ainda estão perdidos. Para essas imagens, é necessário um algoritmo local de reprodução de tons.

3.1.3.6 Ashikhmin

O algoritmo baseia-se no funcionamento do HVS, porém sem tentar construir seu modelo sofisticado. A operação é realizada em três etapas. Primeiramente, estima-se a adaptação local da luminância em cada ponto da imagem. Então, uma função simples é aplicada a esses valores para compactá-los na faixa de exibição necessária. Uma vez que detalhes importantes da imagem podem ser perdidos durante este processo, eles são restabelecidos posteriormente por meio da equação:

$$L_d(x, y) = \frac{L(x, y)TM(L_a(x, y))}{L_a(x, y)} \quad (21)$$

onde $L_a(x, y)$ é o nível de adaptação local calculado como a luminância média em relação a alguma área da imagem local e $TM(L_a(x, y))$ é a função de mapeamento de tons.

3.1.3.7 Pattanaik

A visão humana leva tempo para se adaptar às grandes mudanças na intensidade da cena, e esses ajustes transitórios têm um efeito profundo na aparência visual. A referência [57] oferece um novo operador para incluir essas mudanças de aparência em animações ou simulações interativas em tempo real, e para

corresponder as respostas visuais de um usuário às o usuário experimentaria em uma cena do mundo real.

Mudanças amplas e abruptas na intensidade da cena podem causar uma grande compressão das respostas visuais, seguida de uma recuperação gradual de visão normal. Mecanismos assimétricos governam esses ajustes temporários e oferecem adaptação ao aumento da luz que é muito mais rápida do que o ajuste à escuridão. É o que busca o algoritmo Pattanaik que simula esses mecanismos. Recebendo como entrada um fluxo de quadros de intensidade de cena, o algoritmo cria um fluxo de imagens de exibição em cores.

Todos os componentes do operador são derivados de medidas quantitativas publicadas de fisiologia, psicofísica, ciência das cores e fotografia.

4 PARTE EXPERIMENTAL

Para avaliar o procedimento de síntese e visualização de imagens HDR, foram sintetizadas 5 imagens HDR em ambientes distintos, sendo que cada imagem HDR foi gerada a partir de 5 fotos com diferentes exposições. Para a geração dessas imagens foi utilizado o software livre *Luminance HDR 2.3.1*.

4.1 Descrição do experimento

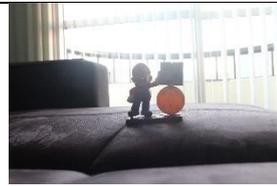
Para a geração das imagens HDR foram capturadas 5 fotografias de cada ambiente com exposições diferentes adaptadas às condições de iluminação do local. Para a aquisição das fotos, foi utilizada uma câmera Canon Rebel T3i (Canon EOS 550D), juntamente com o software *EOS Utility*. Todas as fotos foram capturadas com a câmera apoiada em uma superfície rígida e com disparos e mudanças de exposição feitos via *software*, pois uma vez utilizado o botão de disparo da câmera para as capturas ou para as mudanças de parâmetros, ocorreria uma movimentação da câmera suficiente para deslocar o equipamento e assim prejudicar o experimento. Esse cuidado faz-se necessário, pois o alinhamento entre as fotografias é essencial para que o processo de *tone mapping* tenha um bom resultado.

Para cada conjunto de imagens adquiridas foi gerado um grupo de 6 imagens HDR utilizando o algoritmo de Debevec, explicado no Capítulo 2, com diferentes *profiles*, ou seja, conjunto de parâmetros, diferente. Cada *profile* é definido conforme a tabela a seguir:

	Função de Ponderação (Weighting function)	Curva de resposta
Profile 1	Triangular	Linear
Profile 2	Triangular	Gamma
Profile 3	Plateau	Linear
Profile 4	Plateau	Gamma
Profile 5	Gaussian	Linear
Profile 6	Gaussian	Gamma

Tabela 1 – Descrição dos *profiles* utilizados na geração das imagens HDR

Para o experimento foi utilizado um grupo de 5 imagens com diferentes exposições:

HF	HF_ZOOM	HFPERTO	MARIO
			
f/5.6	f/5.6	f/5.6	f/5.6
tempo expo: 1/2s	tempo expo: 1/8s	tempo expo: 1/30s	tempo expo: 1/4s
ISO-100	ISO-100	ISO-100	ISO-400
			
f/5.6	f/5.6	f/5.6	f/5.6
tempo expo: 1/4s	tempo expo: 1/15s	tempo expo: 1/60s	tempo expo: 1/8s
ISO-100	ISO-100	ISO-100	ISO-400

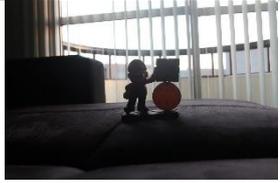
 <p>f/5.6 tempo expo: 1/8s ISO-100</p>	 <p>f/5.6 tempo expo: 1/30s ISO-100</p>	 <p>f/5.6 tempo expo: 1/125s ISO-100</p>	 <p>f/5.6 tempo expo: 1/15s ISO-400</p>
 <p>f/5.6 tempo expo: 1/15s ISO-100</p>	 <p>f/5.6 tempo expo: 1/60s ISO-100</p>	 <p>f/5.6 tempo expo: 1/250s ISO-100</p>	 <p>f/5.6 tempo expo: 1/30s ISO-400</p>
 <p>f/5.6 tempo expo: 1/30s ISO-100</p>	 <p>f/5.6 tempo expo: 1/125s ISO-100</p>	 <p>f/5.6 tempo expo: 1/500s ISO-100</p>	 <p>f/5.6 tempo expo: 1/60s ISO-400</p>

Tabela 2 – Fotos capturadas com diferentes parâmetros de exposição utilizadas como entrada para o algoritmo de geração de imagem HDR.

UFABC_HDR



f/5.6 tempo expo: 1s ISO-100



f/5.6 tempo expo: 1/2s ISO-100



f/5.6 tempo expo: 1/4s ISO-100



f/5.6 tempo expo: 1/8s ISO-100



f/5.6 tempo expo: 1/15s ISO-100

Tabela 3 – Fotos originais capturadas para a geração das imagens HDR que foram utilizadas no experimento com MOS.

	Durand
<i>Profile 1</i>	
<i>Profile 2</i>	

Profile 3



Profile 4



Profile 5



Profile 6



Tabela 4 – Fotos processadas com o algoritmo *Durand* com cada um dos *profiles*.

Mantiuk '06

Profile 1



Profile 2



Profile 3



Profile 4



Profile 5



Profile 6



Tabela 5 – Fotos processadas com o algoritmo *Mantiuk '06* com cada um dos *profiles*.

Nas imagens da Tabela 5, processadas com o algoritmo *Mantiuk*, pode-se notar artefatos “halo”, que nesta imagem se assemelha a uma “fumaça” próxima ao canto do edifício localizado mais à esquerda nas imagens, característicos deste algoritmo, conforme explicado na Seção 3.1.3.1.

A fim de se mensurar a qualidade das imagens, foi realizada uma pesquisa em campo, onde foram exibidas 60 imagens LDR geradas a partir de imagens HDR de cada um dos perfis da Tabela 1. Cada perfil passou pelo processamento de cada um dos 7 algoritmos *de tone mapping* descritos na Seção 3.1.3. O resultado do processamento foi apresentado a 10 pessoas para emitirem suas respectivas opiniões sobre a qualidade das imagens apresentadas. Para isso foi utilizado o método *MOS – Mean Opinion Score* [46], pelo qual voluntários atribuíram notas de 1 a 5 para cada uma das imagens, podendo assim obter um parâmetro quantitativo para medir a qualidade perceptual das imagens produzidas pelos algoritmos estudados. A *ITU - International Telecommunication Union* definiu o *score* de opinião como "valor em uma escala pré-definida que uma pessoa atribui a sua opinião da performance de um sistema " [46]. O MOS é a média dessas pontuações e pode ser utilizado em vários campos. O MOS surgiu como o descritor mais popular de qualidade da mídia percebida. Foi muito utilizado e teve grande sucesso no domínio da qualidade da fala, e, conseqüentemente, também foi usado para outras modalidades, tais como áudio, imagens, vídeo, conteúdo audiovisual e inúmeras aplicações desde testes de laboratório à monitoramento de serviços.

Como métricas objetivas de qualidade para análise das imagens processadas para se comparar juntamente ao MOS, foram utilizados a PSNR, MSE e a SSIM.

A PSNR (*Peak Signal Noise Ratio*) é o método objetivo de avaliação de imagens mais aceito na comunidade científica e trata-se de uma relação entre o máximo

possível de potência de um sinal, pela potência de um ruído, quando é comparado um sinal antes e depois de um processo degradativo. No caso das imagens, esse conceito também é aplicado e é tratado como a relação entre a entrada e a saída de um processo de compressão com perdas, avaliando o quanto esse processo de compressão introduziu ruídos na imagem. No caso deste trabalho, o que está sendo avaliado é o processo (algoritmo) de *tone mapping* a fim de avaliar as perdas em relação à imagem HDR original. “Valores de PSNR acima de 42dB correspondem à compressões que introduzem perdas imperceptíveis ao olho humano, o que significa uma qualidade excepcional.” [47]

A fórmula da PSNR é definida como:

$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \\ &= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned}$$

Onde MAX_I é o máximo valor possível do pixel da imagem. Quando os pixels são representados utilizando 8 bits por amostra, este valor máximo é 255.

O *MSE* (*Mean Square Error*) é o erro quadrático médio e mede a média dos quadrados dos erros ou desvios, ou seja, a diferença entre o estimador e o que é estimado. O MSE é uma função de risco, correspondente ao valor esperado do erro ao quadrado. [48] O MSE é, portanto, uma medida da qualidade de um estimador – é um número sempre não negativo e valores mais próximos de zero são melhores.

O índice de similaridade estrutural (SSIM – *Structural Similarity Index*) é um método para prever a qualidade percebida da televisão digital e imagens

cinematográficas, bem como outros tipos de imagens e vídeos digitais. A primeira versão do modelo foi desenvolvida no *Laboratory for Image and Video Engineering* (LIVE) na Universidade do Texas [58]. O SSIM é usado para medir a semelhança entre duas imagens. O índice SSIM é uma métrica de referência completa; em outras palavras, a medição ou previsão da qualidade da imagem é baseada em uma imagem inicial sem compressão ou sem distorção como referência. O SSIM é projetado para melhorar métodos tradicionais, como pico de relação sinal-ruído (PSNR) e erro quadrático médio (MSE).

Para calcular os indicadores PSNR, MSE e SSIM, as imagens processadas, bem como o original HDR, tiveram suas luminâncias equalizadas, pois os diferentes algoritmos de *Tone Mapping* atuam de diferentes formas sobre a luminância, causando notáveis diferenças entre as imagens. O cálculo desses parâmetros, bem como a equalização das luminâncias foram realizados via MATLAB. Os dados obtidos são mostrados nas tabelas a seguir:

	UFABC_HDR_PROFILE1			
	IMMSE	SSIM	PSNR	MOS
ashikhmin	0,0033	0,6614	24,8231	3
drago	0,0006	0,8575	31,9564	2
durand	0,0011	0,8707	29,4659	2
fattal	0,0008	0,8642	31,1104	3
mantiuk06	0,0022	0,8213	26,5723	2
mantiuk08	0,0006	0,8698	32,4938	3
pattanaik	0,0006	0,859	32,2208	2
pattanaik local	0,004	0,6126	24,0124	2
reinhard02	0,0006	0,859	32,2053	3
reinhard05	0,0007	0,8446	31,7407	3

Tabela 6 – MSE, SSIM, PSNR e MOS para os algoritmos gerados com *profile* 1.

	UFABC_HDR_PROFILE2			
	IMMSE	SSIM	PSNR	MOS
ashikhmin	0,002	0,7092	27,0104	3
drago	0,0013	0,816	28,8152	4
durand	0,0023	0,8229	26,4615	3
fattal	0,0015	0,8192	28,2672	4
mantiuk06	0,003	0,7679	25,2213	3
mantiuk08	0,0012	0,8275	29,191	4
pattanaik	0,0012	0,8215	29,0771	2
pattanaik local	0,0026	0,6955	25,7811	3
reinhard02	0,0012	0,8214	29,049	4
reinhard05	0,0013	0,7956	28,8164	3

Tabela 7 – MSE, SSIM, PSNR e MOS para os algoritmos gerados com *profile 2*.

	UFABC_HDR_PROFILE3			
	IMMSE	SSIM	PSNR	MOS
ashikhmin	0,0026	0,6956	25,8509	3
drago	0,0004	0,881	34,0346	3
durand	0,001	0,8928	30,1207	2
fattal	0,0005	0,8873	32,7748	3
mantiuk06	0,0021	0,8388	26,8421	2
mantiuk08	0,0003	0,8962	34,8713	3
pattanaik	0,0004	0,8797	34,3069	2
pattanaik local	0,0034	0,6091	24,6694	2
reinhard02	0,0003	0,8873	34,6152	2
reinhard05	0,0004	0,8622	33,7675	3

Tabela 8 – MSE, SSIM, PSNR e MOS para os algoritmos gerados com *profile 3*.

	UFABC_HDR_PROFILE4			
	IMMSE	SSIM	PSNR	MOS
ashikhmin	0,0015	0,732	28,2238	4
drago	0,0009	0,8426	30,6094	4
durand	0,0017	0,8485	27,6423	3
fattal	0,0011	0,8444	29,5777	4
mantiuk06	0,0025	0,7918	26,0054	3
mantiuk08	0,0008	0,8526	30,8921	4
pattanaik	0,0008	0,8513	31,0874	2
pattanaik local	0,0022	0,7005	26,6305	3
reinhard02	0,0008	0,8503	31,101	4
reinhard05	0,0009	0,8261	30,6723	3

Tabela 9 – MSE, SSIM, PSNR e MOS para os algoritmos gerados com *profile 4*.

UFABC_HDR_PROFILE5				
	IMMSE	SSIM	PSNR	MOS
ashikhmin	0,0041	0,6523	23,9175	3
drago	0,0007	0,8547	31,334	3
durand	0,0013	0,8635	28,8122	2
fattal	0,0009	0,8625	30,5932	3
mantiuk06	0,0023	0,8215	26,4444	2
mantiuk08	0,0007	0,8636	31,587	3
pattanaik	0,0007	0,8575	31,5584	2
pattanaik local	0,005	0,598	23,0536	3
reinhard02	0,0007	0,8595	31,5577	3
reinhard05	0,0008	0,8213	30,7272	2

Tabela 10 – MSE, SSIM, PSNR e MOS para os algoritmos gerados com *profile* 5.

UFABC_HDR_PROFILE6				
	IMMSE	SSIM	PSNR	MOS
ashikhmin	0,002	0,6874	26,9589	3
drago	0,0012	0,8154	29,3398	4
durand	0,0025	0,8211	25,9639	3
fattal	0,0014	0,8211	28,5748	4
mantiuk06	0,0028	0,7762	25,5771	3
mantiuk08	0,0011	0,8287	29,7877	4
pattanaik	0,0011	0,8227	29,7453	3
pattanaik local	0,0027	0,6879	25,6463	3
reinhard02	0,001	0,8252	29,7946	4
reinhard05	0,0012	0,7772	29,0547	3

Tabela 11 – MSE, SSIM, PSNR e MOS para os algoritmos gerados com *profile* 6.

A partir dos dados obtidos através do MOS e relacionando-os com o PSNR calculado para cada grupo de algoritmos separados por *profile*, obtiveram-se os seguintes gráficos a seguir:

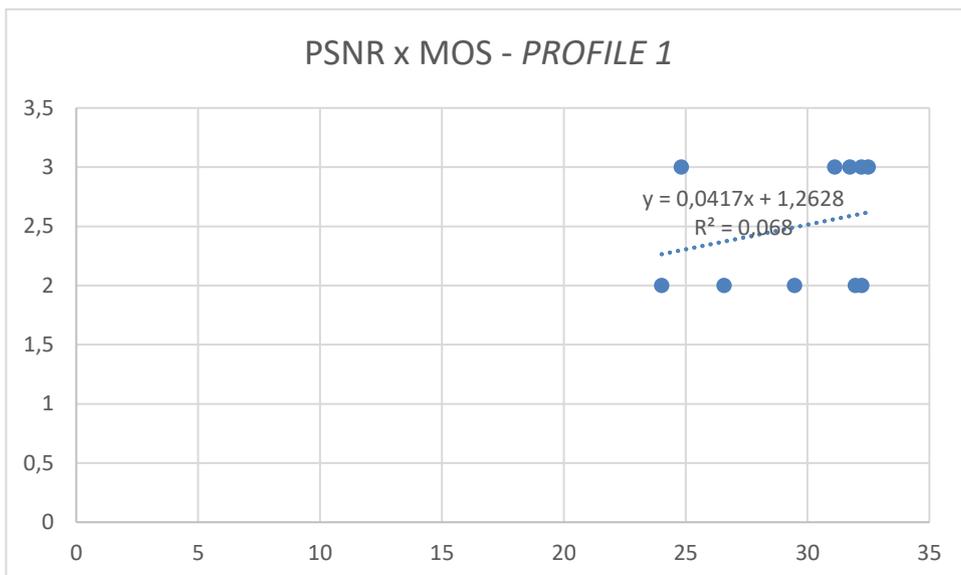


Figura 3 – Gráfico da PSNR x MOS para o grupo de algoritmos gerados a partir do *profile 1*.

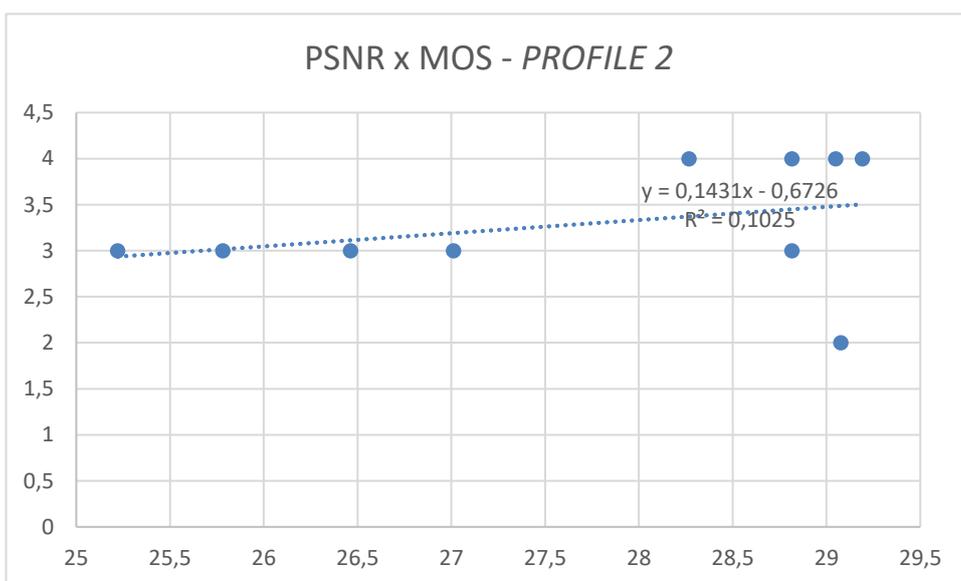


Figura 4 – Gráfico da PSNR x MOS para o grupo de algoritmos gerados a partir do *profile 2*.

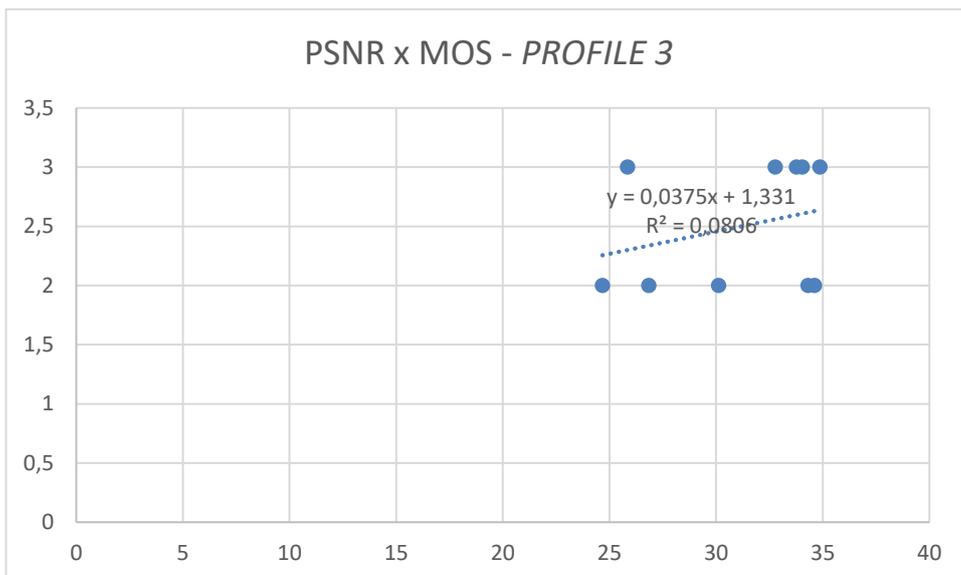


Figura 5 – Gráfico da PSNR x MOS para o grupo de algoritmos gerados a partir do *profile 3*.

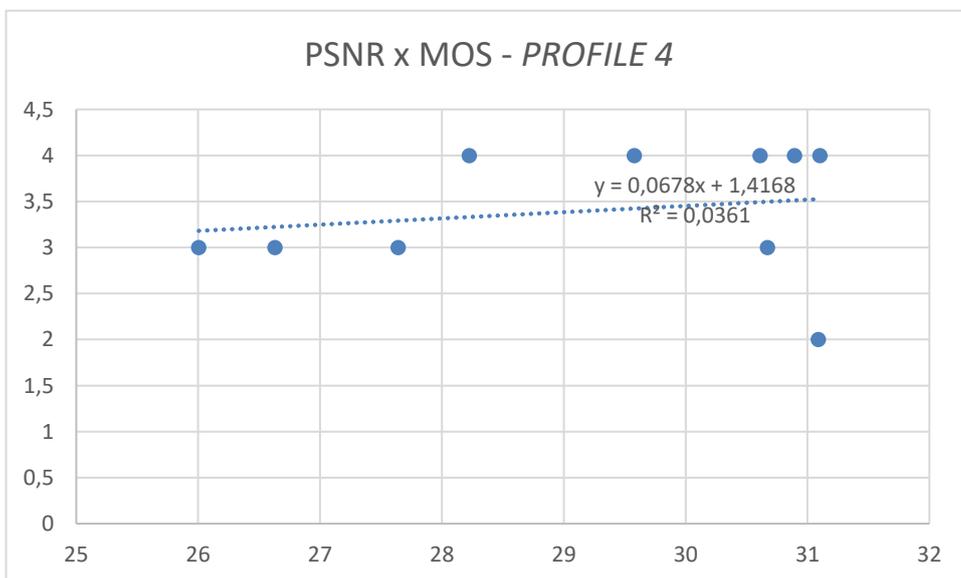


Figura 6 – Gráfico da PSNR x MOS para o grupo de algoritmos gerados a partir do *profile 4*.

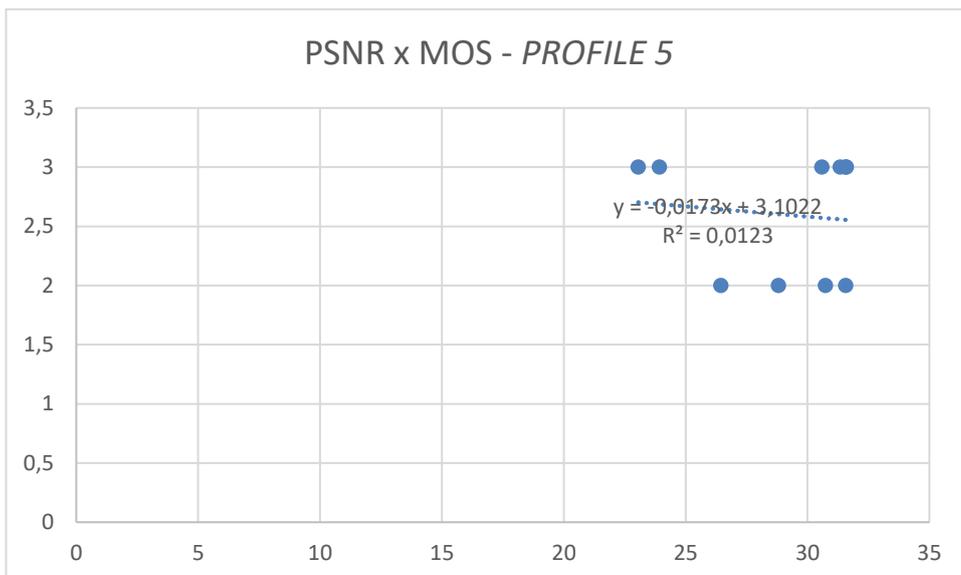


Figura 7 – Gráfico da PSNR x MOS para o grupo de algoritmos gerados a partir do *profile 5*.

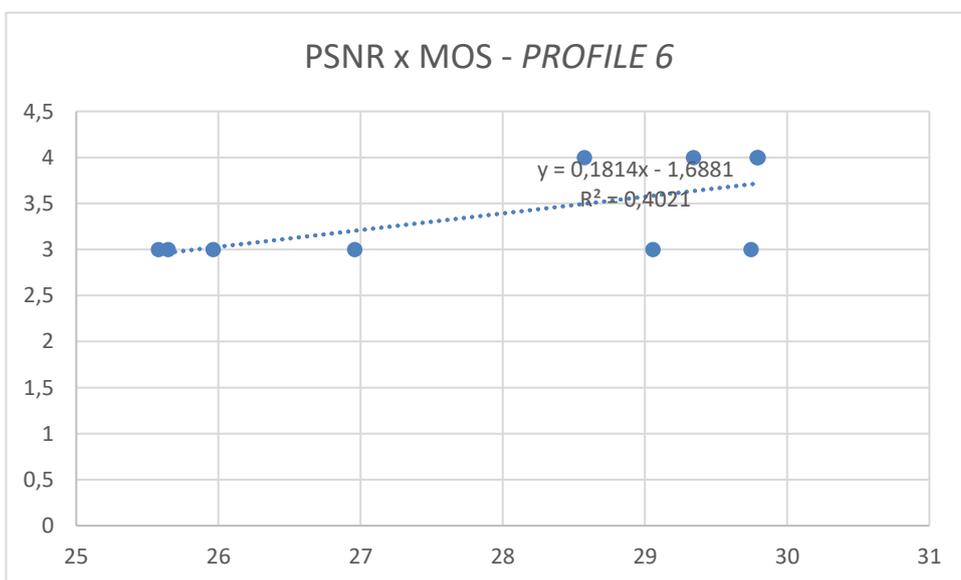


Figura 8 – Gráfico da PSNR x MOS para o grupo de algoritmos gerados a partir do *profile 6*.

Analisando-se os gráficos, percebe-se que se trata de uma nuvem de pontos e que as retas de tendência não são significativas, levando a concluir que o PSNR, no caso dos algoritmos de *Tone Mapping*, não está diretamente associada à percepção

visual do observador. O mesmo comportamento se repete quando compara-se o MOS com o parâmetro SSIM. Porém algo interessante a se notar é que curiosamente, para estas amostras de imagens, os *profiles* 2, 4 e 6 que possuem em comum a curva de resposta *gamma*, obtiveram maior valor de PSNR e também maior avaliação MOS.

Para as outras imagens, obtiveram-se os seguintes dados:

	HF_PROFILE1				HF_PROFILE2		
	IMMSE	SSIM	PSNR		IMMSE	SSIM	PSNR
ashikhmin	0,0059	0,774	22,2704	ashikhmin	0,0095	0,7258	20,2078
drago	0,0022	0,877	26,5432	drago	0,0027	0,8595	25,7085
durand	0,0024	0,8853	26,2617	durand	0,0028	0,8678	25,5231
fattal	0,0034	0,8835	24,6862	fattal	0,0038	0,8673	24,1906
mantiuk06	0,0051	0,8373	22,9568	mantiuk06	0,0054	0,8153	22,6955
mantiuk08	0,002	0,8954	27,0614	mantiuk08	0,0021	0,8825	26,6878
pattanaik	0,0021	0,8825	26,8446	pattanaik	0,0022	0,8687	26,5911
pattanaik				pattanaik			
local	0,0111	0,6468	19,5353	local	0,0088	0,7587	20,5560
reinhard02	0,002	0,8889	27,0440	reinhard02	0,0021	0,875	26,6759
reinhard05	0,002	0,8828	26,9614	reinhard05	0,0022	0,8729	26,6053

	HF_PROFILE3				HF_PROFILE4		
	IMMSE	SSIM	PSNR		IMMSE	SSIM	PSNR
ashikhmin	0,0062	0,7716	22,0470	ashikhmin	0,0097	0,7187	20,1216
drago	0,0026	0,8579	25,9010	drago	0,0027	0,8528	25,6377
durand	0,0027	0,875	25,6261	durand	0,0029	0,8592	25,3529
fattal	0,0037	0,8688	24,2635	fattal	0,004	0,8637	23,9752
mantiuk06	0,0054	0,8226	22,6477	mantiuk06	0,0062	0,8102	22,0745
mantiuk08	0,0023	0,8827	26,4218	mantiuk08	0,0022	0,877	26,5842
pattanaik	0,0024	0,8653	26,1835	pattanaik	0,0023	0,8633	26,4465
pattanaik				pattanaik			
local	0,0114	0,6262	19,4302	local	0,0091	0,7449	20,4043
reinhard02	0,0023	0,8779	26,4079	reinhard02	0,0022	0,873	26,5929
reinhard05	0,0023	0,8741	26,3872	reinhard05	0,0022	0,8663	26,5147

	HF_PROFILES5				HF_PROFILE6		
	IMMSE	SSIM	PSNR		IMMSE	SSIM	PSNR
ashikhmin	0,0058	0,7578	22,34	ashikhmin	0,0095	0,7347	20,2193
drago	0,0021	0,877	26,7189	drago	0,0026	0,8624	25,8352
durand	0,0023	0,8825	26,2962	durand	0,0029	0,8636	25,3086
fattal	0,0034	0,8789	24,7202	fattal	0,0036	0,8656	24,43
mantiuk06	0,0052	0,8302	22,8606	mantiuk06	0,0052	0,8131	22,866
mantiuk08	0,0019	0,8918	27,1936	mantiuk08	0,0021	0,8837	26,8103
pattanaik	0,002	0,8791	27,0033	pattanaik	0,0021	0,8639	26,6832
pattanaik				pattanaik			
local	0,011	0,6386	19,5775	local	0,0084	0,7541	20,78
reinhard02	0,0019	0,8872	27,17	reinhard02	0,0021	0,877	26,8071
reinhard05	0,0019	0,8845	27,1319	reinhard05	0,0021	0,8696	26,6968

Tabela 12 – MSE, SSIM, PSNR e para os algoritmos gerados para a imagem “HF” com *profile* 1, 2, 3, 4, 5 e 6.

	HF_ZOOM_PROFILE1				HF_ZOOM_PROFILE2		
	IMMSE	SSIM	PSNR		IMMSE	SSIM	PSNR
ashikhmin	0,0103	0,753	19,8811	ashikhmin	0,0193	0,6139	17,1441
drago	0,0019	0,9039	27,3019	drago	0,0021	0,898	26,7479
durand	0,0041	0,8025	23,866	durand	0,004	0,8071	23,95
fattal	0,0046	0,8981	23,4183	fattal	0,0047	0,8963	23,3015
mantiuk06	0,0056	0,8537	22,4956	mantiuk06	0,0059	0,847	22,2638
mantiuk08	0,0017	0,9187	27,7025	mantiuk08	0,0017	0,9186	27,6709
pattanaik	0,0019	0,8986	27,2707	pattanaik	0,0018	0,9097	27,4097
pattanaik				pattanaik			
local	0,0106	0,7103	19,7276	local	0,0106	0,718	19,7526
reinhard02	0,0017	0,9175	27,6855	reinhard02	0,0017	0,9179	27,6656
reinhard05	0,0017	0,9179	27,6735	reinhard05	0,0018	0,9141	27,3789

	HF_ZOOM_PROFILE3				HF_ZOOM_PROFILE4		
	IMMSE	SSIM	PSNR		IMMSE	SSIM	PSNR
ashikhmin	0,0095	0,7659	20,2031	ashikhmin	0,0181	0,636	17,4218
drago	0,002	0,9	27,0757	drago	0,0021	0,904	26,8821
durand	0,0038	0,8138	24,1997	durand	0,0038	0,8152	24,1721
fattal	0,0042	0,9003	23,7634	fattal	0,0047	0,8982	23,2358
mantiuk06	0,0051	0,8591	22,9538	mantiuk06	0,0057	0,8506	22,4101
mantiuk08	0,0017	0,916	27,623	mantiuk08	0,0017	0,9197	27,6372
pattanaik	0,002	0,8895	27,0875	pattanaik	0,0018	0,9111	27,432
pattanaik				pattanaik			
local	0,0106	0,7052	19,7481	local	0,01	0,7347	19,9942
reinhard02	0,0017	0,917	27,6356	reinhard02	0,0017	0,9201	27,6516
reinhard05	0,0017	0,9166	27,5737	reinhard05	0,0018	0,9175	27,4923

	HF_ZOOM_PROFILE5				HF_ZOOM_PROFILE6		
	IMMSE	SSIM	PSNR		IMMSE	SSIM	PSNR
ashikhmin	0,0107	0,7433	19,7118	ashikhmin	0,023	0,5594	16,3754
drago	0,0018	0,9049	27,4011	drago	0,0026	0,8672	25,8064
durand	0,0045	0,7932	23,5015	durand	0,0064	0,7424	21,9102
fattal	0,0046	0,8956	23,4039	fattal	0,0053	0,8835	22,7763
mantiuk06	0,0061	0,8445	22,1416	mantiuk06	0,0062	0,8414	22,1067
mantiuk08	0,0017	0,9176	27,7441	mantiuk08	0,0018	0,9106	27,5308
pattanaik	0,0018	0,8975	27,3531	pattanaik	0,0019	0,8933	27,1732
pattanaik				pattanaik			
local	0,0107	0,7055	19,7233	local	0,0124	0,684	19,0718
reinhard02	0,0017	0,9174	27,7421	reinhard02	0,0018	0,9102	27,5361
reinhard05	0,0017	0,9173	27,73	reinhard05	0,0019	0,9026	27,1173

Tabela 13 – MSE, SSIM, PSNR e para os algoritmos gerados para a imagem

“HF_ZOOM” com *profile* 1, 2, 3, 4, 5 e 6.

	HFPERTO_PROFILE1				HFPERTO_PROFILE2		
	IMMSE	SSIM	PSNR		IMMSE	SSIM	PSNR
ashikhmin	0,0011	0,7667	29,6771	ashikhmin	0,0028	0,6401	25,4848
drago	0,0005	0,8481	32,6226	drago	0,0012	0,794	29,0644
durand	0,0006	0,8483	32,4511	durand	0,0012	0,7946	29,1822
fattal	0,0011	0,8422	29,579	fattal	0,0021	0,7842	26,8559
mantiuk06	0,0024	0,8134	26,2869	mantiuk06	0,0032	0,7431	24,8991
mantiuk08	0,0005	0,8518	32,753	mantiuk08	0,0012	0,8011	29,3063
pattanaik	0,0005	0,8485	32,6597	pattanaik	0,0012	0,796	29,1917
pattanaik				pattanaik			
local	0,0011	0,7535	29,7805	local	0,0016	0,7239	27,8718
reinhard02	0,0005	0,8491	32,7208	reinhard02	0,0012	0,7991	29,2991
reinhard05	0,0006	0,8355	32,386	reinhard05	0,0013	0,7709	28,7662

	HFPERTO_PROFILE3				HFPERTO_PROFILE4		
	IMMSE	SSIM	PSNR		IMMSE	SSIM	PSNR
ashikhmin	0,001	0,7586	29,9441	ashikhmin	0,0025	0,6491	25,9949
drago	0,0004	0,8614	33,8736	drago	0,0013	0,7971	28,965
durand	0,0004	0,8667	33,8953	durand	0,0013	0,7944	28,9858
fattal	0,001	0,8594	30,139	fattal	0,0021	0,7786	26,683
mantiuk06	0,0022	0,8222	26,6538	mantiuk06	0,0033	0,7447	24,8735
mantiuk08	0,0004	0,8706	34,1692	mantiuk08	0,0013	0,7998	29,0147
pattanaik	0,0004	0,8671	34,058	pattanaik	0,0013	0,7949	28,813
pattanaik				pattanaik			
local	0,0009	0,7526	30,3189	local	0,0016	0,7246	27,9118
reinhard02	0,0004	0,8637	33,9217	reinhard02	0,0013	0,798	29,0218
reinhard05	0,0004	0,8505	33,5779	reinhard05	0,0014	0,7718	28,5664

	HFPERTO_PROFILE5				HFPERTO_PROFILE6		
	IMMSE	SSIM	PSNR		IMMSE	SSIM	PSNR
ashikhmin	0,0012	0,7594	29,0844	ashikhmin	0,0031	0,6214	25,0536
drago	0,0007	0,8424	31,745	drago	0,0012	0,7997	29,3551
durand	0,0007	0,8427	31,7124	durand	0,0011	0,8014	29,4867
fattal	0,0012	0,8363	29,1311	fattal	0,002	0,7871	26,8867
mantiuk06	0,0026	0,8075	25,9179	mantiuk06	0,0032	0,7512	24,9775
mantiuk08	0,0006	0,8477	32,009	mantiuk08	0,0011	0,8058	29,5847
pattanaik	0,0007	0,8402	31,6411	pattanaik	0,0011	0,799	29,4561
pattanaik				pattanaik			
local	0,0012	0,7432	29,1267	local	0,0017	0,7215	27,8047
reinhard02	0,0007	0,8426	31,8222	reinhard02	0,0011	0,7989	29,4087
reinhard05	0,0007	0,8254	31,3834	reinhard05	0,0012	0,7822	29,3439

Tabela 14 – MSE, SSIM, PSNR e para os algoritmos gerados para a imagem “HFPERTO” com *profile* 1, 2, 3, 4, 5 e 6.

	MARIO_PROFILE1				MARIO_PROFILE2		
	IMMSE	SSIM	PSNR		IMMSE	SSIM	PSNR
ashikhmin	0,0007	0,7869	31,5783	ashikhmin	0,0008	0,7923	30,7444
drago	0,0002	0,9114	36,6429	drago	0,0004	0,8833	34,0674
durand	0,0006	0,907	32,5284	durand	0,0007	0,8821	31,5221
fattal	0,0032	0,8878	24,8882	fattal	0,005	0,853	22,9808
mantiuk06	0,0025	0,8501	26,0924	mantiuk06	0,0028	0,8118	25,4533
mantiuk08	0,0003	0,9071	35,4638	mantiuk08	0,0004	0,8851	34,42
pattanaik	0,0002	0,9071	36,5014	pattanaik	0,0003	0,9033	34,6892
pattanaik				pattanaik			
local	0,0014	0,7201	28,5608	local	0,0007	0,8054	31,8363
reinhard02	0,0002	0,9139	37,072	reinhard02	0,0004	0,887	34,3774
reinhard05	0,0002	0,9048	36,5404	reinhard05	0,0004	0,8795	34,3043

	MARIO_PROFILE3				MARIO_PROFILE4		
	IMMSE	SSIM	PSNR		IMMSE	SSIM	PSNR
ashikhmin	0,0005	0,8164	33,3796	ashikhmin	0,0007	0,8169	31,8273
drago	0,0001	0,9176	38,2419	drago	0,0003	0,899	34,6784
durand	0,0009	0,9	30,5641	durand	0,0012	0,8784	29,2091
fattal	0,0029	0,8961	25,4482	fattal	0,004	0,8712	23,9632
mantiuk06	0,0025	0,8469	26,0909	mantiuk06	0,0026	0,8285	25,8018
mantiuk08	0,0001	0,9238	39,2616	mantiuk08	0,0003	0,9076	35,4884
pattanaik	0,0001	0,9186	38,7689	pattanaik	0,0003	0,9111	35,3188
pattanaik				pattanaik			
local	0,0013	0,7121	28,7737	local	0,0006	0,8102	32,3779
reinhard02	0,0001	0,9258	39,1617	reinhard02	0,0003	0,9074	35,4183
reinhard05	0,0001	0,9212	38,7401	reinhard05	0,0003	0,9005	35,1312

	MARIO_PROFILE5				MARIO_PROFILE6		
	IMMSE	SSIM	PSNR		IMMSE	SSIM	PSNR
ashikhmin	0,0007	0,7717	31,5598	ashikhmin	0,0009	0,7766	30,6717
drago	0,0002	0,9019	36,0675	drago	0,0004	0,8809	34,4085
durand	0,001	0,8972	30,0239	durand	0,0007	0,8782	31,3231
fattal	0,0032	0,8859	24,9904	fattal	0,0055	0,8431	22,6045
mantiuk06	0,0025	0,8433	25,9493	mantiuk06	0,003	0,8039	25,2171
mantiuk08	0,0003	0,9052	34,7187	mantiuk08	0,0003	0,8877	35,1429
pattanaik	0,0002	0,9109	36,8262	pattanaik	0,0003	0,8994	34,8348
pattanaik local	0,0014	0,7135	28,3886	pattanaik local	0,0007	0,7974	31,7392
reinhard02	0,0002	0,9046	36,5382	reinhard02	0,0003	0,8855	34,8482
reinhard05	0,0003	0,8892	35,1629	reinhard05	0,0003	0,8794	34,7892

Tabela 15 – MSE, SSIM, PSNR e para os algoritmos gerados para a imagem “MARIO” com *profile* 1, 2, 3, 4, 5 e 6.

Para a imagem HF_ZOOM:

Algoritmo	<i>Profile 1</i>	<i>Profile 2</i>
Pattanaik Global		
Pattanaik Local		

Mantiuk '08 (contraste global)		
Mantiuk '06		

Tabela 16 – Imagem “HF ZOOM” processada com os algoritmos Pattanaik (local e global) e Mantiuk (local e global).

Para as imagens da Tabela 16 processadas com o algoritmo *Mantiuk '06*, pode-se notar artefatos “halo”, que aparecem como componentes de baixa de frequência sobre uma superfície com conteúdo de alta frequência (“listras da camisa”), o que ocorre devido ao mapeamento de contrastes conforme explicado na Seção 3.1.3.1.

5 CONCLUSÃO

Analisando os gráficos, podemos observar que o MOS não reflete diretamente aos parâmetros MSE, SSIM e PSNR, concluindo-se desta forma que uma imagem HDR é melhor observada em um display próprio para exibição. Em alguns trabalhos acadêmicos, utilizou-se PSNR para tentar medir a qualidade relativa das imagens que passaram pelo processo de *tone mapping* e neste trabalho buscou-se verificar se a PSNR se traduzia em qualidade perceptual. Verificou-se experimentalmente que a qualidade percebida não reflete a diferença entre a imagem HDR e a imagem que sofreu o processo de *tone mapping*.

Por mais que se busque aproximar a visualização de uma imagem HDR em dispositivos LDR, a impressão que o usuário terá da imagem sempre será subjetiva, e extremamente dependente das características intrínsecas dos algoritmos de *tone mapping*, uma vez que cada um deles processa as imagens de forma diferente e estão sujeitos a erros de processamento dependendo das imagens que forem inseridas como entrada. Desta forma, ainda enfatizado pelo resultado do experimento realizado com observadores, em que se obtiveram resultados aparentemente aleatórios em cada observação, conclui-se que apesar do esforço, para se conseguir reproduzir imagens HDR confiavelmente é necessário um dispositivo próprio em que não haja compressão de luminância.

6 REFERÊNCIAS

- [1] P. Sen, C. Aguerrebere. "Practical High Dynamic Range Imaging of Everyday Scenes", IEEE Signal Processing Magazine, Signal Processing for Computational Photography and Displays, September 2016
- [2] Rodrigues, L. M. "Processamento de Imagens HDR Utilizando Filtros Não Lineares e Decomposição Multiescala, Universidade Federal do ABC, 2014.
- [3] Magic Lantern HDR video to tonemapped video with MATLAB scripts. Disponível em: <<http://www.semifluid.com/2013/10/15/magic-lantern-hdr-video-to-tonemapped-video-with-matlab-scripts>>. Acesso em 8 de dezembro de 2016
- [4] C. Aguerrebere, J. Delon, Y. Gousseau, and P. Musé, "Best algorithms for HDR image generation. A study of performance bounds," SIAM J. Imaging Sci., vol. 7, no. 1, pp. 1–34, 2014.
- [5] N. K. Kalantari, E. Shechtman, C. Barnes, S. Darabi, D. B. Goldman, and P. Sen, "Patch-based high dynamic range video," ACM Trans. Graph. (Proc. SIGGRAPH Asia), vol. 32, no. 6, pp. 202:1–202:8, Nov. 2013.
- [6] O. T. Tursun, A. O. Akyz, A. Erdem, and E. Erdem, "The state of the art in HDR deghosting: A survey and evaluation," Comput. Graph. Forum, vol. 34, no. 2, pp. 683–707, 2015.
- [7] B. C. Madden, "Extended intensity range imaging," Univ. of Pennsylvania, Tech. Rep. MS-CIS-93-96, 1993.
- [8] S. Mann and R. W. Picard, "On being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures," in Proc. IS&T, 1995, pp. 442–448.
- [9] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in Proc. SIGGRAPH, 1997, pp. 369–378.
- [10] S. Mann, "Comparametric Equations with Practical Applications in Quantigraphic Image Processing", IEEE Transactions on Image Processing, 9, 1389-1406, 2000.
- [11] V. Brajovic and T. Kanade, "A sorting image sensor: an example of massively parallel intensity-to-time processing for low-latency computational sensors," in Proc. ICRA, vol. 2, pp. 1638–1643, Apr. 1996.
- [12] H. Zhao, B. Shi, C. Fernandez-Cull, S.-K. Yeung, and R. Raskar, "Unbounded high dynamic range photography using a modulo camera," in Proc. IEEE Intl. Conf. Computational Photography (ICCP), Apr. 2015, pp. 1–10.
- [13] U. Seger, U. Apel, and B. Höfflinger, "HDRC-Imagers for natural visual perception," in Handbook of Computer Vision and Application, B. Jähne, H. Haußecker, and P. Geißler, Eds. New York: Academic, 1999, vol. 1, pp. 223–235
- [14] S. Nayar and T. Mitsunaga, "High dynamic range imaging: Spatially varying pixel exposures," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), June 2000, vol. 1, pp. 472–479.

- [15] S. K. Nayar, V. Branzoi, and T. E. Boult, "Programmable imaging: Towards a flexible camera," *Int. J. Comput. Vis.*, vol. 70, no. 1, pp. 7–22, 2006.
- [16] K. Hirakawa and P. Simon, "Single-shot high dynamic range imaging with conventional camera hardware," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2011, pp. 1339–1346.
- [17] M. Schöberl, A. Belz, J. Seiler, S. Foessel, and A. Kaup, "High dynamic range video by spatially non-regular optical filtering," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, 2012, pp. 2757–2760.
- [18] C. Aguerrebere, A. Almansa, J. Delon, Y. Gousseau, and P. Musé, "Single shot high dynamic range imaging using piecewise linear estimators," in *Proc. IEEE Int. Conf. Computational Photography (ICCP)*, 2014, pp. 1–10.
- [19] A. Serrano, F. Heide, D. Gutierrez, G. Wetzstein, and B. Masia, "Convolutional sparse coding for high dynamic range imaging," *Comput. Graph. Forum*, vol. 35, no. 2, 2016.
- [20] M. D. Tocci, C. Kiser, N. Tocci, and P. Sen, "A versatile HDR video production system," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 41:1–41:10, July 2011.
- [21] M. Aggarwal and N. Ahuja, "Split aperture imaging for high dynamic range," *Int. J. Comput. Vis.*, vol. 58, no. 1, pp. 7–17, June 2004.
- [22] A. Badki, N. K. Kalantari, and P. Sen, "Robust radiometric calibration for dynamic scenes in the wild," in *Proc. IEEE Int. Conf. Computational Photography (ICCP)*, Apr. 2015, pp. 1–10.
- [23] M. Granados, B. Ajdin, M. Wand, C. Theobalt, H. P. Seidel, and H. P. A. Lensch, "Optimal HDR reconstruction with linear digital cameras," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 215–222.
- [24] C. Aguerrebere, J. Delon, Y. Gousseau, and P. Musé, "Simultaneous HDR image reconstruction and denoising for dynamic scenes," in *Proc. IEEE Intl. Conf. Computational Photography (ICCP)*, 2013, pp. 1–11.
- [25] A. Tomaszewska and R. Mantiuk, "Image registration for multi-exposure high dynamic range image acquisition," in *Proc. Int. Conf. Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, 2007, pp. 49–56.
- [26] L. Bogoni, "Extending dynamic range of monochrome and color images through fusion," in *Proc. IEEE Intl. Conf. Pattern Recognition (ICPR)*, 2000, pp. 3007–3016.
- [27] M. Gupta, D. Iso, and S. Nayar, "Fibonacci exposure bracketing for high dynamic range imaging," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2013, pp. 1473–1480.
- [28] H. Zimmer, A. Bruhn, and J. Weickert, "Freehand HDR imaging of moving scenes with simultaneous resolution enhancement," *Comput. Graph. Forum*, vol. 30, no. 2, pp. 405–414, Apr. 2011.
- [29] T. Grosch, "Fast and robust high dynamic range image generation with camera and object movement," in *Proc. Int. Symp. Vision, Modeling and Visualization*, 2006, pp. 277–284.

- [30] O. Gallo, N. Gelfand, W. Chen, M. Tico, and K. Pulli, "Artifact-free high dynamic range imaging," in Proc. IEEE Intl. Conf. Computational Photography (ICCP), 2009, pp. 1–7.
- [31] E. Khan, A. Akyuz, and E. Reinhard, "Ghost removal in high dynamic range images," in Proc. IEEE Int. Conf. Image Processing (ICIP), 2006, pp. 2005–2008.
- [32] C. Lee, Y. Li, and V. Monga, "Ghost-free high dynamic range imaging via rank minimization," IEEE Signal Processing Lett., vol. 21, no. 9, pp. 1045–1049, Sept. 2014.
- [33] T.-H. Oh, J.-Y. Lee, and I. S. Kweon, "Robust high dynamic range imaging by rank minimization," IEEE Trans. Pattern Anal. Machine Intelli., vol. 37, no. 6, pp. 1219–1232, June 2015.
- [34] J. Hu, O. Gallo, K. Pulli, and X. Sun, "HDR deghosting: How to deal with saturation?" in Proc. IEEE Conf. Computer Vision and Pattern Recog. (CVPR), June 2013, pp. 1163–1170.
- [35] S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High dynamic range video," ACM Trans. Graph., vol. 22, no. 3, pp. 319–325, July 2003.
- [36] R. J. McEliece, "The theory of information and coding", 2nd ed, Cambridge: Cambridge University Press, 2002, 397 p. (Encyclopedia of mathematics and its applications).
- [37] W. E. Ryan, S. Lin, "Channel codes: classical and modern", Cambridge: Cambridge University Press, 2009, 692 p.
- [38] Arquivo: Oscar-gustave-rejlander two ways of life.jpg. Disponível em: <https://upload.wikimedia.org/wikipedia/commons/3/31/Oscar-gustave-rejlander_two_ways_of_life.jpg>. Acesso em 8 de dezembro de 2016.
- [39] High Dynamic Range. Disponível em: <https://pt.wikipedia.org/wiki/High_dynamic_range>. Acesso em 8 de dezembro de 2016
- [40] Arquivo: HDRI-Example.jpg. Disponível em: <<https://upload.wikimedia.org/wikipedia/commons/b/b3/HDRI-Example.jpg>>. Acesso em 8 de dezembro de 2016.
- [41] Arquivo: Old_saint_pauls_1.jpg. Disponível em: <https://upload.wikimedia.org/wikipedia/commons/6/6e/Old_saint_pauls_1.jpg>. Acesso em 8 de dezembro de 2016.
- 2010.
- [42] "A comparative review of tone-mapping algorithms for high dynamic range video" Gabriel Eilertsen, Rafal K. Mantiuk, Jonas Unger. In: Eurographics State-of-The-Art Report, To appear, 2017. Disponível em: <http://www.cl.cam.ac.uk/~rkm38/pdfs/eilertsen2017tmo_star.pdf>. Acesso em 21 de abril de 2017.
- [43] "Desenvolvimento de um operador de tone mapping para dispositivos com ecrã pequeno". Disponível em:<https://www.researchgate.net/publication/259570120_Desenvolvimento_de_um_operador_de_tone_mapping_para_dispositivos_com_ecra_pequeno>. Acesso em 21 de abril de 2017.

- [44] P. E. Debevec and J. Malik, Recovering high dynamic range radiance maps from photographs, in SIGGRAPH, 1997, pp. 369–378
- [45] A. Castorina, A. Capra, A. Bruna, S. Battiato, A Novel Data Fusion Technique for Imaging Devices In Consumer Electronics, 2004 IEEE International Symposium on
- [46] R. C. Streij, S. Winkler, D.S. Hands, Mean Opinion Score (MOS) revisited: Methods and applications, limitations and alternatives
- [47] “A Universal Image Quality Index”. Disponível em:<<https://pt.slideshare.net/michelalves/presentation26outubro2010-uirpsnrmseimage-qualityindexes>> Acesso em: 16 de novembro de 2017
- [48] Lehmann, E. L.; Casella, George (1998). Theory of Point Estimation (2nd ed.). New York: Springer.
- [49] E. Reinhard, M. Stark, P. Shirley, J. Ferwerda, Photographic Tone Reproduction for Digital Images In ACM Transactions on Graphics, 2002
- [50] R. Mantiuk, K. Myszkowski, e H.P. Seidel, A Perceptual Framework for Contrast Processing of High Dynamic Range Images Em ACM Transactions on Applied Perception, 2006
- [51] R. Fattal, D. Lischinski, and M. Werman, Gradient Domain High Dynamic Range Compression In ACM Transactions on Graphics, 2002
- [52] F. Durand and J. Dorsey, Fast Bilateral Filtering for the Display of High-Dynamic-Range Images In ACM Transactions on Graphics, 2002
- [53] M. Ashikhmin, A Tone Mapping Algorithm for High Contrast Images Em the 13th Eurographics Workshop on Rendering, 2002
- [54] Peli, E. 1990. Contrast in complex images. Journal of the Optical Society of America A 7, 10, 2032–2040.
- [55] Mitchell, E. N. 1984. Photographic Science. John Wiley and Sons, New York.
- [56] Stroebel, L., Compton, J., Current, I., and Zakia, R. 2000. Basic photographic materials and processes, second ed. Focal Press.
- [57] S.N. Pattanaik, J. Tumblin, H. Yee, D.P. Greenberg, Time-Dependent Visual Adaptation for Realistic Image Display In Proceedings of ACM SIGGRAPH 2000.
- [58] 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, 2004.
- [59] “HDR Photography with F/OSS Tools (LuminanceHDR)”. Disponível em:<<https://patdavid.net/2013/05/hdr-photography-with-foss-tools.html>> Acesso em: 16 de novembro de 2017