

FUNDAÇÃO GETULIO VARGAS
ESCOLA DE MATEMÁTICA APLICADA – FGV/EMAp
CURSO DE GRADUAÇÃO EM MATEMÁTICA APLICADA

IMAGENS HDR E O MÉTODO DE TONE MAPPING

por
Fernanda da Cunha Duarte

Rio de Janeiro
2016

FUNDAÇÃO GETULIO VARGAS
ESCOLA DE MATEMÁTICA APLICADA – FGV/EMAp
CURSO DE GRADUAÇÃO EM MATEMÁTICA APLICADA

IMAGENS HDR E O MÉTODO DE TONE MAPPING

"Declaro ser o único autor do presente projeto de monografia que se refere ao plano de trabalho a ser executado para continuidade da monografia e ressalto que não recorri a qualquer forma de colaboração ou auxílio de terceiros para realizá-lo a não ser nos casos e para os fins autorizados pelo professor orientador".

Fernanda da Cunha Duarte

Orientador: Paulo Cezar Carvalho
Coorientador: Asla Medeiros e Sá

Rio de Janeiro
2016

FERNANDA DA CUNHA DUARTE

IMAGENS HDR E O MÉTODO DE TONE MAPPING

“Projeto de Monografia apresentado à Escola de Matemática Aplicada – FGV/EMAp
como requisito parcial para continuidade ao trabalho de monografia.”

Aprovado em _____ de _____ de _____.

Grau atribuído ao Projeto de Monografia: _____.

Professor Orientador: Paulo Cezar Carvalho
Coorientador: Asla Medeiros e Sá
Escola de Matemática Aplicada – FGV/EMAp
Fundação Getulio Vargas

Agradecimentos

Agradeço à minha família, por terem confiado em minhas decisões e me apoiado mesmo quando o caminho a seguir parecia incerto e (um tanto) nebuloso. E ainda, por terem me ensinado tantos valores maravilhosos e mostrado o que é viver a vida com amor.

Aos meus professores orientadores, Paulo Cezar Carvalho e Asla Medeiros e Sá, cujo conhecimento e apoio foram indispensáveis para a conclusão desta etapa tão importante.

Aos meus amigos de curso, em especial à minha amiga (e irmã de alma) Polyana Barboza, que foi a mais fiel companheira de trabalhos e que esteve sempre ao meu lado, do início ao fim, compartilhando os momentos mais felizes e oferecendo o ombro nas situações mais difíceis. E ainda, aos meus amigos Gabriel Mesquita, João Amaro e José Valentim Rosa.

A todos os professores e funcionários da Escola de Matemática Aplicada, cujos ensinamentos e dedicação foram essenciais para o meu crescimento acadêmico e pessoal.

À Claudia Cossich e à Beralda Lima, que me deram conselhos essenciais que me levaram a ingressar no curso de Matemática Aplicada.

SUMÁRIO

1	Introdução	5
1.1	Imagens Digitais	5
1.2	Intensidade de luz e Alcance Dinâmico	6
2	Aquisição de Imagens HDR	6
2.1	Função Resposta da Câmera	7
2.2	O Algoritmo	8
2.3	Escolha das Localizações de Pixels	10
2.3.1	Filtro Gaussiano	10
2.3.1	Algoritmo do Melhor Candidato	11
3	Visualização de Imagens HDR	12
3.1	Luminância	12
3.2	Operador Global	13
3.3	Operador Local	15
4	Ajustes Finais	16
5	Resultados	16
6	Referências Bibliográficas	25

1 Introdução

O tema do presente trabalho está relacionado a um problema muito comum no contexto da fotografia, que surge quando se pretende fotografar uma cena com grande diferença de iluminação, ou seja, apresentando partes de muita e pouca iluminação que uma câmera convencional não é capaz de capturar simultaneamente. Tomando como foco câmeras fotográficas digitais, o objetivo é estudar e reproduzir métodos capazes de contornar tais limitações encontradas nos dispositivos disponíveis no mercado atualmente. A figura 1 a seguir ilustra bem a situação retratada.



Fig. 1: Exemplo de uma cena apresentando extremos de iluminação. Neste caso, o fotógrafo teria que escolher entre capturar os detalhes do interior menos iluminado da igreja, ou os detalhes do vitral mais iluminado.

1.1 Imagens Digitais

Uma imagem colorida contínua pode ser modelada como uma função $f : U \subset \mathbb{R}^2 \rightarrow C \subset \mathbb{R}^3$, em que cada coordenada (x, y) do subconjunto retangular U corresponde à localização de um pixel e está associada a um ponto $c = f(x, y)$ pertencente ao subconjunto C , cujos valores estão relacionados a um espaço de cores.

A imagem digital corresponde a uma representação discreta da imagem contínua, apresentando, portanto, uma quantidade finita de pontos em cada subconjunto anteriormente definido. Quando se trata de câmeras fotográficas digitais, imagens digitais são obtidas através da discretização da incidência contínua de luz nos sensores da câmera. O sinal de imagem recebido pela câmera é convertido em uma quantidade finita de pontos, que definem a resolução espacial e a resolução de cor das imagens geradas.

Resolução espacial pode ser definida como a quantidade de pontos resultantes da discretização do sinal recebido, por unidade de área. Resolução de cor corresponde à quantidade de bits utilizada para armazenar as informações de cores obtidas através da discretização do fluxo de luz em um subconjunto discreto finito do espaço de cores associado a C .

1.2 Intensidade de Luz e Alcance Dinâmico

A incapacidade de capturar simultaneamente extremos de iluminação está relacionada ao conceito de alcance dinâmico. Alcance dinâmico em fotografia pode ser definido como a razão entre os valores de maior e menor luminosidade presentes em uma cena. Este termo é utilizado para mensurar as intensidades de iluminação presentes na cena e também para indicar o quanto o dispositivo em questão é capaz de capturá-las ou representá-las. Deste modo, o problema de produzir uma representação fiel ao que está sendo observado ocorre porque os dispositivos convencionais possuem baixo alcance dinâmico, conseguindo representar apenas 2^8 níveis diferentes de intensidade de luz, enquanto que a visão humana é capaz de enxergar cerca de 2^{24} níveis.

Imagens e dispositivos de baixo alcance dinâmico são chamados de LDR (*Low Dynamic Range*). Dispositivos LDR utilizam imagens 24-bits, ou seja, cada pixel da imagem é representado por 3 bytes, cada um relacionado a um canal de cor R (*Red*), G (*Green*) e B (*Blue*). Cada byte, por sua vez, possui 8 bits em sua composição, podendo assumir, portanto, 2^8 valores distintos, o que explica o intervalo de valores variando de 0 a 255 para cada canal de cor. Tal composição explica a representação limitada dos diferentes níveis de intensidade luminosa. Um dos meios encontrados para contornar a incapacidade dos dispositivos de representar o intervalo maior de intensidade de luz presente na cena real foi a construção das chamadas imagens HDR (*High Dynamic Range*), ou seja, de alto alcance dinâmico.

Diante do exposto, tem-se como objetivo estudar e reproduzir algumas técnicas desenvolvidas para a aquisição e visualização de imagens HDR. A etapa de aquisição consiste na geração dos dados que irão compor o arquivo de imagem de alto alcance dinâmico e envolve conceitos fundamentais em fotografia como exposição, irradiância e função resposta de uma câmera. Já a etapa de visualização está relacionada à forma como estes dados serão tratados para permitirem a reprodução da imagem gerada em dispositivos LDR convencionais, uma vez que eles possuem limitações que impossibilitam a representação direta destes dados, conforme dito anteriormente. Os algoritmos utilizados para este tratamento final dos dados compõe a chamada técnica de Mapeamento de Tons (ou *Tone Mapping*). Todos os algoritmos foram implementados em Python, por questões de praticidade, domínio e disponibilidade, e os resultados finais obtidos estão expostos na última seção deste trabalho.

2 Aquisição de imagens HDR

O processo de aquisição de imagens HDR consiste na obtenção de valores relacionados à intensidade de luz que se aproximem de forma visualmente satisfatória da iluminação real percebida na cena. O método exposto a seguir se baseia no algoritmo proposto no artigo "*Recovering High Dynamic Range Radiance Maps from Photographs*" [2]. A ideia geral é capturar um número suficiente de fotos com diferentes exposições de uma cena estática que apresente alto alcance dinâmico.

Exposição pode ser entendida de forma simplificada como a quantidade de luminosidade recebida pela câmera durante o período de tempo em que as lentes permanecem abertas para captar a luz. Deste modo, para uma mesma quantidade de luz, quanto mais tempo as lentes ficam abertas, maior será a exposição dos sensores digitais à luminosidade da cena e mais clara sairá a foto. Assim, cada foto tirada deve conseguir captar detalhes que as outras fotos com diferentes exposições são incapazes de capturar, de forma que o alcance dinâmico da cena seja obtido de forma completa.

As imagens digitais geradas são representadas através de matrizes, em que cada posição representa um pixel da imagem e contém informações de cor e intensidade de luz. O objetivo do algoritmo é combinar as informações de cada localização de pixel das diferentes imagens em uma única imagem, ou seja, obter um único mapeamento de luminosidade em valores de pixel que seja capaz de cobrir o alcance dinâmico presente na cena retratada.

Na figura 2 abaixo mostramos alguns exemplos de imagens com diferentes exposições.

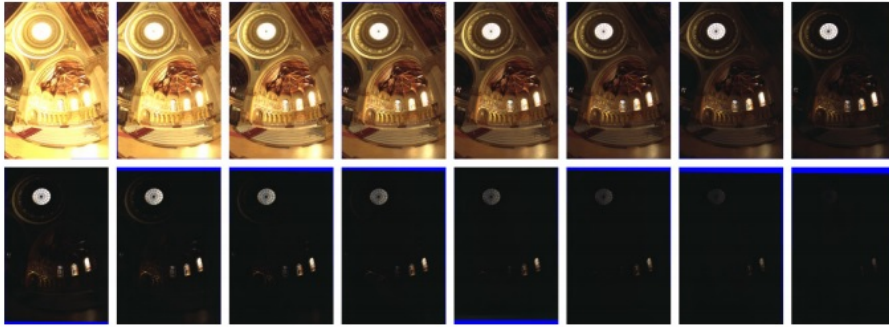


Fig. 2: Exemplos de imagens com múltiplas exposições. As bordas azuis foram introduzidas no processo de escaneamento das imagens para que todas estivessem corretamente alinhadas.

2.1 Função resposta da câmera

As câmeras fotográficas digitais convencionais são modeladas de forma a tentar reproduzir a maneira como nossos olhos percebem a luz. Deste modo, os receptores de energia luminosa presentes nas câmeras são constituídos por filtros vermelhos, verdes e azuis que, através da captação destas três cores primárias na luz incidente, são capazes de combiná-las em diferentes proporções para armazenar e reproduzir uma versão digitalizada em pixels da cor que esteja perceptualmente próxima àquela visualizada na cena. Sendo assim, cada pixel irá armazenar três números, cada um referente à quantidade de energia luminosa captada de cada cor.

Uma das quantidades utilizadas para mensurar a luz incidente nestes receptores é a chamada irradiância, que mede o fluxo de energia captado por área (W/m^2 , watts por metro quadrado). Pensando na irradiância como sendo a quantidade de luz exata que chega até os sensores da câmera, conhecê-la significa saber os valores que cada pixel da imagem deveria assumir para estar de acordo com o real alcance dinâmico da cena. No entanto, em função de limitações inerentes aos dispositivos convencionais, o valor registrado e armazenado não corresponde à exata quantidade de luz incidente.

Para se obter precisamente os valores de irradiância, é preciso primeiramente relacioná-lo ao conceito de exposição. Pode-se definir exposição como sendo o produto da irradiância pelo tempo em que determinado pixel ficou exposto à luz. O valor digitalizado de cada pixel é uma função não linear da quantidade de exposição qual ele foi submetido. Chamamos esta função de função resposta característica f de cada câmera.

A função f nos permite ter conhecimento sobre a forma como os valores de exposição são mapeados em valores de pixel e, por conseguinte, como é feito o processo contrário também, através da função inversa f^{-1} . Uma vez obtida a função resposta da câmera, podemos calcular a sua inversa para cada localização de pixel nas

diferentes imagens com diferentes exposições e, através da definição de exposição feita anteriormente, obter um mapeamento completo da irradiância da cena de alto alcance dinâmico.

2.2 O algoritmo

Seja Z o valor digitalizado de determinada localização de pixel e seja X a exposição a que este pixel foi submetido. De acordo com a definição citada no item anterior, podemos escrever X como o produto da irradiância E pelo tempo de exposição Δt , ou seja, $X = E \Delta t$. Assim, sendo f a função resposta da câmera que relaciona Z e X , podemos escrever $Z = f(X) = f(E \Delta t)$. Além disso, tendo em mente que é intuitivo pensar em f como sendo uma função monótona crescente, já que uma maior exposição à luz deve resultar em valores digitalizados mais altos, podemos assumir que a inversa de f^{-1} está bem definida e, portanto, $X = f^{-1}(Z)$. Como o objetivo aqui é obter um mapeamento da irradiância, obtendo f^{-1} e calculando X , é possível então encontrar $E = \frac{X}{\Delta t}$.

O primeiro passo é representar as P imagens de resolução $m \times n$, com diferentes tempos de exposição conhecidos Δt_j , na forma de matrizes de pixels com dimensões $m \times n$. É importante ressaltar que para imagens coloridas, os canais de cores são tratados separadamente e de forma independente, ou seja, é encontrada uma função resposta para cada canal. Assumindo que as imagens foram capturadas de um mesmo ponto de observação e que o processo de captura é realizado de forma rápida o suficiente para que alterações de iluminação de uma imagem para outra sejam ignoradas, o valor E_i de irradiância para uma mesma localização de pixel i em cada imagem será considerado constante.

Pretende-se assim, criar uma matriz de pixels $Z_{N \times P}$, em que o número de linhas corresponde à quantidade de localizações de pixels escolhidas para utilizar na recuperação de f^{-1} (este processo será melhor detalhado posteriormente) e o número de colunas corresponde à quantidade de fotografias capturadas. Dito isto, para facilitar cálculos futuros, serão feitas as seguintes definições:

$$\begin{aligned} Z_{ij} &= f(E_i \Delta t_j) & (1) \\ \implies f^{-1}(Z_{ij}) &= E_i \Delta t_j \\ \implies \ln f^{-1}(Z_{ij}) &= \ln E_i + \ln \Delta t_j \end{aligned}$$

Fazendo $g = \ln f^{-1}$, tem-se:

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

em que g e E_i são as variáveis desconhecidas a serem encontradas. Para recuperar g precisamos apenas de um número finito de valores, já que a quantidade de diferentes intensidades Z_{ij} que os pixels de um dispositivo de reprodução convencional pode assumir é finita (geralmente indo de 0 a 255, quando temos 8 bits por canal). O próximo passo a ser realizado é a utilização do método de mínimos quadrados para encontrar

os valores de $g(Z_{ij})$ e E_i que melhor satisfazem a equação (2) e que minimizem a seguinte função objetivo:

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

onde Z_{min} e Z_{max} são os valores mínimo e máximo que os pixels do dispositivo em questão podem assumir. O primeiro termo da função objetivo acima refere-se ao erro que queremos minimizar. Já o segundo termo, que está em função da segunda derivada g'' , serve para garantir suavização para a função g encontrada. Como estamos lidando com valores discretos, definimos $g'' = g(z-1) - 2g(z) + g(z+1)$. O parâmetro λ é uma constante definida de acordo com a quantidade desejada de suavização da função resposta que se deseja obter e deve ser escolhido levando em conta a quantidade de ruído esperada nos valores Z_{ij} . Logo, quanto maior for o λ escolhido, mais suave será a curva retornada.

A função $w(z)$ que aparece multiplicando ambos os termos da função objetivo merece atenção especial. Como o problema das imagens que estamos lidando ocorre justamente nas regiões com extremos de iluminação, pode-se esperar que perto desses extremos a função resposta será aproximada com maior imprecisão. Deste modo, a inserção de $w(z)$ é feita para tentar amenizar tais efeitos e priorizar as regiões da curva de resposta que estão mais bem definidas. A ideia é fazer com que $w(z)$ sirva como uma função de pesos, assumindo valores baixos para valores mais extremos de Z_{ij} - e, conseqüentemente, diminuindo o peso que a parcela multiplicada por w terá no processo de estimação - e valores mais altos para valores intermediários de Z_{ij} . Sendo assim, a função w é definida da seguinte forma:

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

Além da função objetivo, uma restrição é introduzida ao problema para que seja estabelecido um fator de escala, cujo intuito é atribuir a quantidade de exposição $X = 1$ para o pixel de valor intermediário $Z_{mid} = \frac{1}{2}(Z_{min} + Z_{max})$. Desta forma, introduzimos a restrição $g(Z_{mid}) = 0$.

O sistema linear proposto pode ser facilmente resolvido utilizando o método de decomposição em valores singulares (SVD). Uma vez obtido o resultado, o próximo passo é utilizar os valores encontrados $g(Z_{ij})$ para encontrar todos os possíveis valores de irradiância E_i . Ao invés de encontrar E_i pela simples substituição de valores na equação (2), pode-se obter resultados mais robustos utilizando uma função que pondera (utilizando w novamente) as exposições conhecidas de todas as imagens para uma mesma localização de pixel, através da seguinte equação:

$$\ln E_i = \frac{\sum_{j=1}^P w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^P w(Z_{ij})} \quad (5)$$

Assim, encontrando os valores de E_i para cada localização de pixel i , obtemos a imagem HDR. É conveniente acrescentar que a curva resposta obtida pode ser usada para determinar valores de irradiância para qualquer outra imagem adquirida pelo mesmo dispositivo de fotografia em questão.

2.3 Escolha das localizações de pixels

No artigo [2] estudado, o processo de escolha das localizações de pixels que servirão como entrada do algoritmo é feito de forma manual. No presente trabalho, optou-se por realizar tal escolha de forma automatizada, utilizando-se primeiramente uma técnica de filtragem como um passo de pré-processamento das imagens e aplicando posteriormente um algoritmo para a escolha automática das localizações.

Primeiramente, é importante definir qual é a quantidade total mínima de pixels a serem escolhidos para que o sistema linear seja determinado. Como são escolhidas N localizações para cada uma das P fotografias, tem-se então N valores indeterminados de $\ln E_i$ e $(Z_{max} - Z_{min})$ de g (pois $g(Z_{mid})$ já está definido como restrição do problema). Logo,

$$\begin{aligned} NP &> N + (Z_{max} - Z_{min}) \\ \implies NP - N &> (Z_{max} - Z_{min}) \\ \implies N(P - 1) &> (Z_{max} - Z_{min}) \end{aligned}$$

é a quantidade de pixels necessária para que o sistema seja resolvido adequadamente.

2.3.1 Filtro Gaussiano

O Filtro Gaussiano é uma técnica muito utilizada em processamento de imagens, cujo objetivo mais comum é a diminuição de ruídos em imagens. A filtragem é feita através da aplicação de um *kernel* de tamanho arbitrário $n \times n$, com n ímpar, na imagem através de um processo de convolução, de tal modo que o valor de cada pixel seja substituído por uma média ponderada entre o seu valor original e os valores dos pixels na sua vizinhança. A ideia é atribuir maior peso aos pixels que estão mais próximos do pixel em questão, o que explica a utilização de uma distribuição Gaussiana no processo. O *kernel* é construído associando o pixel analisado à posição central, com coordenadas $(0,0)$, e os pixels da sua vizinhança às posições restantes correspondentes. Assim, cada posição (x, y) do *kernel* assumirá um valor $f(x, y)$ definido pela função densidade de probabilidade da distribuição Gaussiana no caso bidimensional:

$$f(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

em que σ é um dos parâmetros escolhidos e corresponde ao desvio-padrão da distribuição. Em seguida, realizamos um processo de normalização para que a soma dos pesos seja igual a 1, ou seja, dividimos cada valor encontrado pela soma total. Definido o *kernel*, cada valor é então multiplicado pelo pixel correspondente e a soma das parcelas corresponderá ao novo valor do pixel em questão na imagem filtrada. Esse processo é repetido até que todos os pixels da imagem tenham sido analisados.

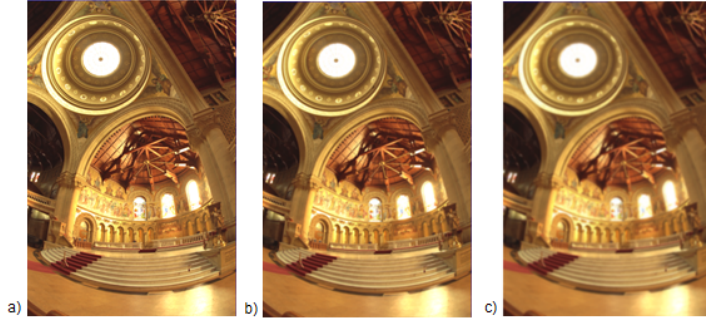


Fig. 3: (a) Imagem original sem filtro; (b) Imagem após a aplicação do filtro Gaussiano, com $\sigma = 1$; (c) Imagem após a aplicação do filtro Gaussiano, com $\sigma = 2$

2.3.2 Algoritmo do “Melhor Candidato” (*Best-Candidate Algorithm*)

A ideia do algoritmo proposto em [4] é escolher pixels aleatoriamente de tal forma que, a cada iteração, o novo pixel escolhido seja aquele de maior distância em relação aos pixels já escolhidos, o que resulta em uma amostra melhor distribuída, se comparada à amostragem feita apenas pela aplicação de uma distribuição uniforme.

A cada iteração o algoritmo seleciona 10 novos candidatos, de acordo com uma distribuição uniforme, e analisa qual é aquele que se encontra mais distante das localizações de pixels já escolhidas. O melhor candidato é então inserido na lista de pixels escolhidos e uma nova busca é iniciada.

Comparando o resultado obtido pelo algoritmo em questão com o resultado obtido utilizando apenas uma distribuição uniforme, podemos ver que o algoritmo escolhe localizações bem distribuídas, sem a ocorrência de “aglomerações” de pixels:

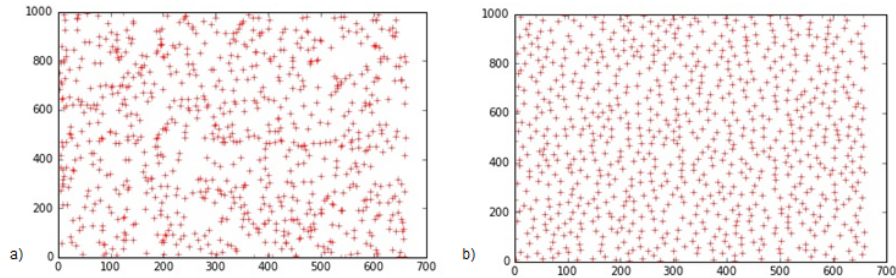


Fig. 4: (a) Amostragem de pontos utilizando apenas a distribuição uniforme; (b) Amostragem utilizando o algoritmo proposto.

3 Visualização de imagens HDR: O método de Mapeamento de Tons (*Tone Mapping*)

Após estudar a aquisição de imagens HDR, o próximo passo é tratar do processo de visualização deste tipo de arquivo. Como foi dito anteriormente, os dispositivos convencionais não são capazes de representar o grande intervalo de luminosidade presente na cena real. O chamado método de *Tone Mapping* foi criado para contornar a incapacidade de visualizar a imagem HDR gerada no processo de aquisição. O objetivo é produzir uma nova imagem LDR a partir das informações presentes no arquivo de imagem HDR.

Os operadores de *Tone Mapping* geralmente são separados em duas categorias: operadores globais e operadores locais. Operadores globais são caracterizados por tratarem todos os pixels da mesma forma, sem levar em conta particularidades de cada região da imagem. Por apresentarem tal comportamento generalizado, os algoritmos que definem os métodos globais costumam ser computacionalmente simples e ter baixa complexidade. No entanto, a maior desvantagem está no fato de não conseguirem preservar de forma acurada detalhes e contrastes locais da imagem. Deste modo, quando se pretende dar maior atenção a detalhes de pequenas regiões, o método local passa a ser mais conveniente. Apesar de ter maior complexidade de implementação, este método geralmente apresenta resultados mais fiéis à cena capturada, por tratar cada pixel individualmente e levando em conta apenas informações presentes em regiões específicas.

Os métodos aqui estudados foram retirados do artigo “*Tone Mapping High Dynamic Range Images by Novel Histogram Adjustment*” [5]. O primeiro método apresenta um operador global, enquanto que o segundo aproveita esse operador global para propor uma abordagem local.

3.1 Luminância

Antes de comentar os métodos estudados, é importante introduzir um conceito fundamental no processo de mapeamento de tons e que servirá como entrada para todos os métodos. Luminância é uma medida utilizada para mensurar a intensidade de luz por unidade de área em uma determinada superfície. A unidade SI é cd/m^2 (candela por metro ao quadrado). É através das informações de luminância que a compressão da imagem de alto alcance dinâmico é feita, uma vez que estamos interessados justamente no intervalo de intensidade luminosa da cena.

Existem diversos métodos para extrair os valores de luminância de uma imagem. A ideia é separar a matriz de pixels entre informações de luminância e de cromaticidade. A diferença de um método para outro está no quão eficaz é essa separação, ou seja, na capacidade de criar matrizes de luminância e cromaticidade de modo que uma não contenha informações da outra. Realizando pesquisas a respeito do tema, optou-se por utilizar a transformação para o espaço de cores YUV para extrair os dados de luminância. A transformação é aplicada sobre a matriz de pixels da imagem HDR e retorna três matrizes Y , U e V , em que Y é a matriz de luminância. Para extrair Y diretamente, aplicamos a seguinte fórmula:

$$Y = 0.2126R + 0.7152G + 0.0722B$$

Uma vez obtida a matriz Y , utilizamos as informações presentes em cada localização de pixel para aplicar os algoritmos de Tone Mapping e obter a imagem LDR final.

3.2 Operador global

Como foi dito anteriormente, um operador global não leva em conta características locais, ou seja, não há uma preocupação em analisar cada pixel individualmente e as características do seu entorno. Sendo assim, operadores globais modificam todos os pixels igualmente, resultando em um mesmo tratamento tanto para as regiões mais escuras, quanto para as regiões mais claras da imagem, o que muitas vezes pode acarretar em uma perda significativa de informação.

O algoritmo escolhido proposto em [2] consiste em duas etapas. O primeiro passo é comprimir os valores de luminância da imagem HDR para o intervalo limitado do dispositivo LDR através de uma simples função logarítmica. A intuição por trás de tal aplicação é aumentar o brilho e o contraste para os pixels com baixa luminância e diminuir tais características para os valores de alta luminância. Assim, aplicamos a seguinte função:

$$D(I) = (D_{max} - D_{min}) \frac{\log(L + \tau) - \log(L_{min} + \tau)}{\log(L_{max} + \tau) - \log(L_{min} + \tau)} \quad (6)$$

onde L_{min} e L_{max} são os valores de mínima e máxima luminância da cena, D_{min} e D_{max} são os valores mínimo e máximo do intervalo permitido pelo dispositivo de visualização e τ é um parâmetro arbitrário que controla a intensidade de brilho da imagem resultante.

Pode-se notar que, como o parâmetro τ aparece como um termo de acréscimo aos valores de luminância, quanto maior for o valor escolhido para τ , mais escura será a imagem. Isso porque, pelo formato da função log e pela normalização realizada na equação (6), valores de baixa luminância acabam sendo mapeados para valores menores do que seriam caso um valor mais baixo de τ fosse escolhido. A figura abaixo exemplifica bem tal comportamento:

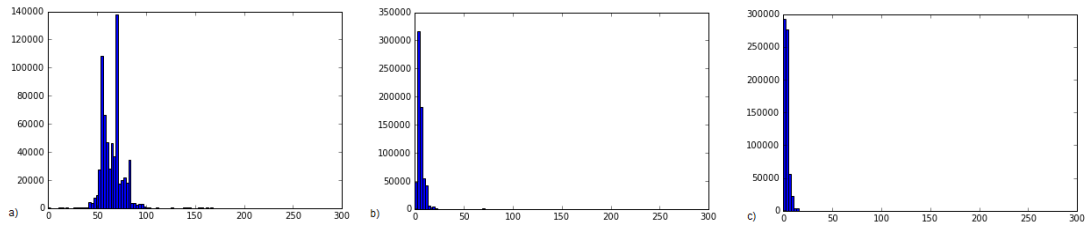


Fig. 5: Histogramas dos valores de luminância após a aplicação da equação (6) com (a) $\tau = 0.001$; (b) $\tau = 0.5$; (c) $\tau = 0.9$

Por outro lado, quanto menor τ , mais clara será a imagem resultante e, portanto, o parâmetro escolhido deve variar de acordo com as particularidades da cena retratada e com os objetivos que se pretendem atingir. Caso a preocupação principal sejam as características inerentes à cena, um método para calcular o parâmetro de forma automatizada é proposto. A ideia é mapear a média do log da luminância da cena L_{med} para um ponto específico (chamado de valor chave) do intervalo do dispositivo, de acordo com a iluminação geral presente. Deste modo, quanto mais clara for a cena,

mais próxima a média calculada deve estar do extremo mais alto do intervalo de baixo alcance dinâmico e, portanto, mais alto deve ser o valor chave. Os seguintes passos mostram como calcular τ :

1. Calcular $L_{med} = e^{\frac{1}{N} \sum_{x,y} \log(\epsilon + L(x,y))}$
2. Calcular $k = 0.42^{\frac{2\log L_{med} - \log L_{min} - \log L_{max}}{\log L_{max} - \log L_{min}}}$
3. Encontra o valor τ tal que $k = \frac{\log(L_{med} + \tau) - \log(L_{min} + \tau)}{\log(L_{max} + \tau) - \log(L_{min} + \tau)}$

onde N é o número total de pixels da imagem, ϵ é um valor pequeno para evitar que valores nulos sejam passados no logaritmo e k é o valor chave mencionado anteriormente. O valor de τ pode ser encontrado através do método de Newton.

A segunda etapa do operador global proposto é amenizar a falta de detalhes e de contraste da imagem resultante da aplicação anterior. Tal limitação ocorre pois o mapeamento é feito de forma linear, sem levar em consideração a distribuição dos pixels presentes na imagem, ou seja, no caso dos dispositivos convencionais com intervalos de 0-255, o intervalo é subdividido em 256 intervalos iguais e pixels que forem mapeados em um mesmo subintervalo, apresentarão um mesmo valor d ao final do processo. Pensando no histograma da imagem, tal quantização linear ignora características particulares de regiões mais densas, uma vez que uma grande quantidade de pixels poderá ser introduzida em um mesmo subintervalo, o que resultará na indesejada perda de detalhes. Ao passo que em regiões menos densas, quantidades ínfimas de pixels ocuparão um subintervalo com tamanho igual aos intervalos ocupados por regiões mais populosas.

Para contornar tal situação, os autores propõem uma junção da técnica de quantização linear com a técnica de equalização de histograma (*histogram equalization*), que leva em conta a distribuição dos pixels. A ideia desta segunda técnica é dividir o intervalo 0-255 em subintervalos que levem em conta apenas a população de pixels que caracteriza o histograma, ou seja, cada subintervalo deve conter a mesma quantidade de pixels, não importando os valores $D(I)$ assumidos por cada pixel.

Enquanto o método linear provoca perda de detalhes e de contraste, a equalização de histograma provoca o efeito inverso: a imagem resultante apresenta um contraste exagerado. Isso ocorre pois em regiões mais populosas os subintervalos resultantes são bem menores, já em regiões com poucos pixels estes subintervalos podem se tornar consideravelmente grandes. A junção de ambas as técnicas é feita então para que uma compense os efeitos negativos da outra, definindo assim pontos de cortes le_n para cada subintervalo que ponderem os pontos de cortes definidos por cada uma.

Esta etapa recebe como entrada os valores $D(I)$ resultantes da etapa anterior e o algoritmo que a define é chamado de quantizador linear e equalizador baseado em ajuste de histograma (*emphhistogram adjustment based linear to equalized quantizer*, HALEQ). Definindo l_i como os pontos de corte do método linear e e_i como os pontos de corte do método por histograma, definimos os novos pontos de corte como:

$$le_n = l_n + \beta(e_n - l_n) \quad (7)$$

onde $0 < \beta < 1$ é um parâmetro arbitrário.

Com essa definição, é fácil perceber que se $\beta = 0$, os pontos de corte seguirão o método linear, enquanto que se $\beta = 1$, seguirão o método por histograma. Nesta etapa

do presente trabalho, foram testados vários valores de β para cada imagem e foram escolhidos aqueles que produziram resultados visualmente mais satisfatórios.

Assim, com os valores D (já mapeados no intervalo de baixo alcance dinâmico) o algoritmo pode ser iniciado. A ideia principal é ir dividindo recursivamente o intervalo $[D_{min}, D_{max}]$. Definido o valor β , iniciamos a primeira iteração encontrando o valor l_0 que divide o intervalo em duas partes iguais e o valor e_0 que define duas partes do intervalo que contêm a mesma quantidade de pixels. Em seguida, calculamos o valor $le_0 = l_0 + \beta(e_0 - l_0)$ e realizamos o primeiro corte do intervalo neste ponto, obtendo $[D_{min}, le_0]$ e $[le_0, D_{max}]$. Estes passos são repetidos recursivamente nos novos subintervalos definidos por cada corte le_i até que se atinja o número desejado de divisões. Assim, para dispositivos com valores de 0-255, o intervalo $[D_{min}, D_{max}]$ é dividido em 256 partes com 8 iterações, tal que à primeira parte atribuímos o valor 0, à segunda o valor 1 e assim sucessivamente até atribuir o valor 255 à última parte. Por fim, substituímos cada valor D pelo valor correspondente ao subintervalo que ele está inserido e obtemos a imagem LDR final.

3.3 Operador Local

O operador local proposto é inspirado no sistema adaptativo da visão humana, caracterizado pela capacidade de se adaptar ao nível de um determinado campo de visão ao redor de um ponto específico da cena. O operador é classificado como um Ajuste por Histograma Adaptativo Local (*Adaptive Local Histogram Adjustment*, ALHA) e a ideia geral é utilizar o método global discutido anteriormente através de uma abordagem local, ou seja, propõe-se a segmentação da imagem em regiões menores e a posterior aplicação do operador HALEQ em cada região.

No artigo, a partição proposta é uma simples divisão da imagem em retângulos de tamanhos iguais. No entanto, tal metodologia acaba colocando partes muito semelhantes da imagem em regiões diferentes, podendo levar a alguns efeitos indesejáveis nas imagens finais, que podem ser corrigidos por funções propostas pelos autores. Inicialmente, optou-se por seguir esta mesma abordagem, mas os resultados não foram tão satisfatórios quanto o esperado. Deste modo, optou-se por segmentar a imagem de tal forma que pixels com características semelhantes fossem alocados em uma mesma região, ou seja, em um mesmo cluster, utilizando o método de agrupamento K-means.

O método de clusterização K-means divide um conjunto de n observações em k clusters, de tal modo que a distância entre cada observação e a média do cluster ao qual ela pertence seja mínima. Na primeira iteração do algoritmo, k pontos são escolhidos aleatoriamente para ser o centroide de cada cluster. Assim, uma observação irá pertencer ao cluster cujo centroide esteja mais próximo dela. Definidos todos os clusters, calcula-se então o ponto médio entre todos os pontos de um mesmo cluster e este ponto passará a ser o novo centroide. Na próxima iteração os clusters são redefinidos seguindo a mesma regra de distância mínima e o algoritmo para quando os clusters param de ser modificados ou quando um número máximo de iterações é atingido.

Dito isto, o primeiro passo do operador local é aplicar a equação (6) na imagem HDR, assim como foi feito no método global. Com a imagem resultante, aplica-se um algoritmo de K-means para dividir a imagem em 10 clusters (a quantidade de clusters foi definida de acordo com o número k que forneceu os resultados mais satisfatórios visualmente). Em seguida, utilizando o mesmo parâmetro β escolhido para o operador global ($\beta = 0.5$ forneceu bons resultados para a maioria das imagens), a função HALEQ é aplicada em cada cluster separadamente, de modo que cada pixel assumira um valor inteiro $d(x, y)$ dado por

$$d(x, y) = HALEQ_k[D(x, y)] \quad (x, y) \in k$$

onde k representa o k -ésimo cluster.

4 Ajustes finais

Uma vez encontrada a matriz de luminância da nova imagem LDR com a aplicação do método de *Tone Mapping*, precisamos, por fim, combinar todas as informações - de luminância e cromaticidade - para compor o resultado final. Os valores de baixo alcance dinâmico dos pixels de cada canal foram calculados através das seguintes fórmulas:

$$\begin{aligned} R_{out} &= \left(\frac{R_{in}}{L_{in}}\right)^\gamma L_{out} \\ G_{out} &= \left(\frac{G_{in}}{L_{in}}\right)^\gamma L_{out} \\ B_{out} &= \left(\frac{B_{in}}{L_{in}}\right)^\gamma L_{out} \end{aligned}$$

em que L_{in} e L_{out} são os valores de luminância de alto e baixo alcance dinâmico, respectivamente, R_{in} , G_{in} e B_{in} são os valores dos pixels da imagem de alto alcance dinâmico (obtida após o processo de aquisição) e R_{out} , G_{out} e B_{out} são os valores dos pixels da imagem LDR final. O parâmetro γ é arbitrário e, para o presente experimento, forneceu bons resultados para valores entre 0.6 e 0.8.

Após obter a matriz LDR final, utilizamos um filtro de nitidez (*sharpening filter*) para aumentar a nitidez das imagens que apresentaram certa perda de informação durante todo o processo de compressão. Na próxima seção apresentamos os resultados obtidos com os algoritmos implementados e os parâmetros escolhidos para cada caso.

5 Resultados

O código implementado para produzir os resultados a seguir foi colocado em um repositório do GitHub e pode ser acessado através do link: <https://github.com/fernandacduarte/TCC>.

5.1



Fig. 6: Múltiplas exposições variando de 15s a 0.001s, com incrementos de 1-stop .
Fonte: [9]

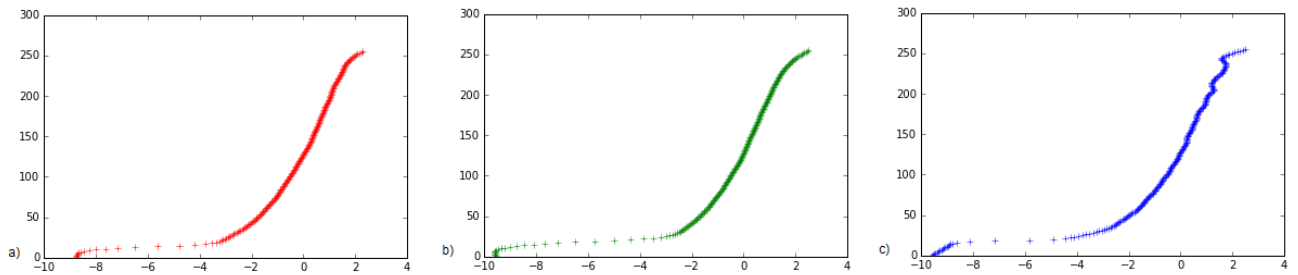


Fig. 7: Função resposta da câmera recuperada para cada canal separadamente: (a) Vermelho com $\lambda = 32$; (b) Verde com $\lambda = 67$; (c) Azul com $\lambda = 10$



Fig. 8: Imagens LDR após a aplicação dos algoritmos de Tone Mapping; (a) Método global (b) Método local.

5.2

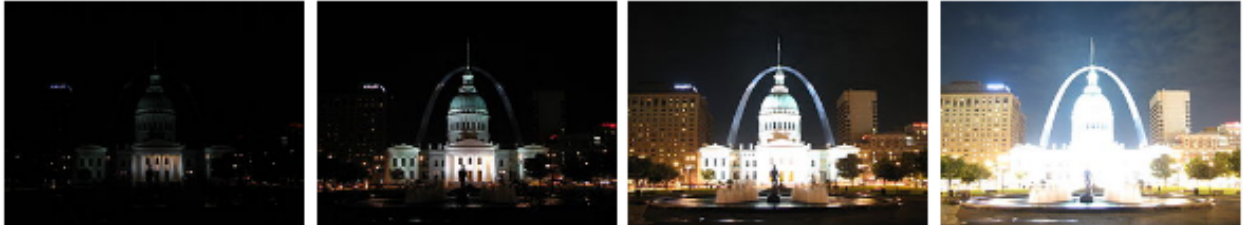


Fig. 9: Imagens com tempo de exposição de 0.033s, 0.25s, 2.5s e 15s, respectivamente.
Fonte: [10]

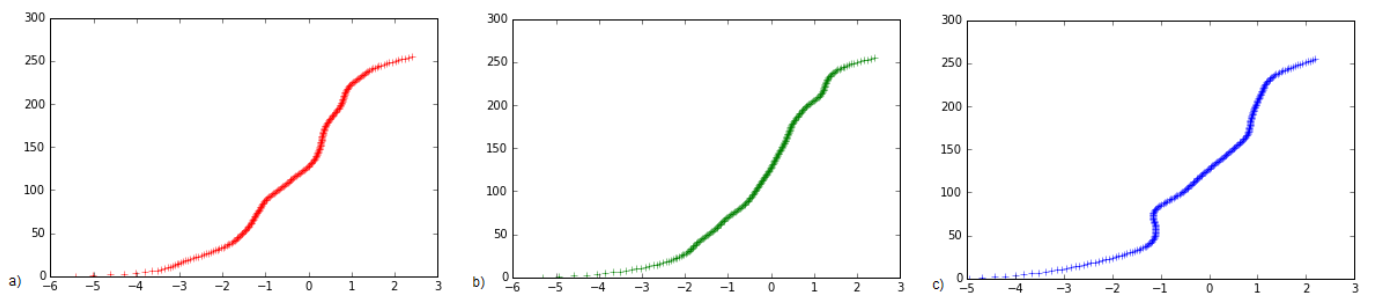


Fig. 10: Função resposta da câmera recuperada para cada canal separadamente: (a) Vermelho com $\lambda = 50$; (b) Verde com $\lambda = 50$; (c) Azul com $\lambda = 80$



Fig. 11: Imagens LDR após a aplicação dos algoritmos de Tone Mapping; (a) Método global (b) Método local.

5.3



Fig. 12: Imagens com tempo de exposição de 0.6s, 1.3s, 2.5s, 4s, 8s e 15s, respectivamente. Fonte: [11]

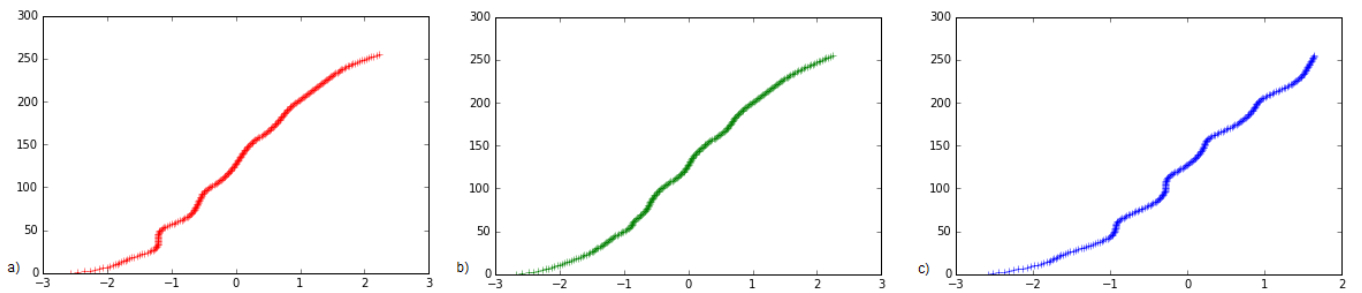


Fig. 13: Função resposta da câmera recuperada para cada canal separadamente: (a) Vermelho com $\lambda = 50$; (b) Verde com $\lambda = 30$; (c) Azul com $\lambda = 40$



a)



b)

Fig. 14: Imagens LDR após a aplicação dos algoritmos de Tone Mapping; (a) Método global (b) Método local.

5.4

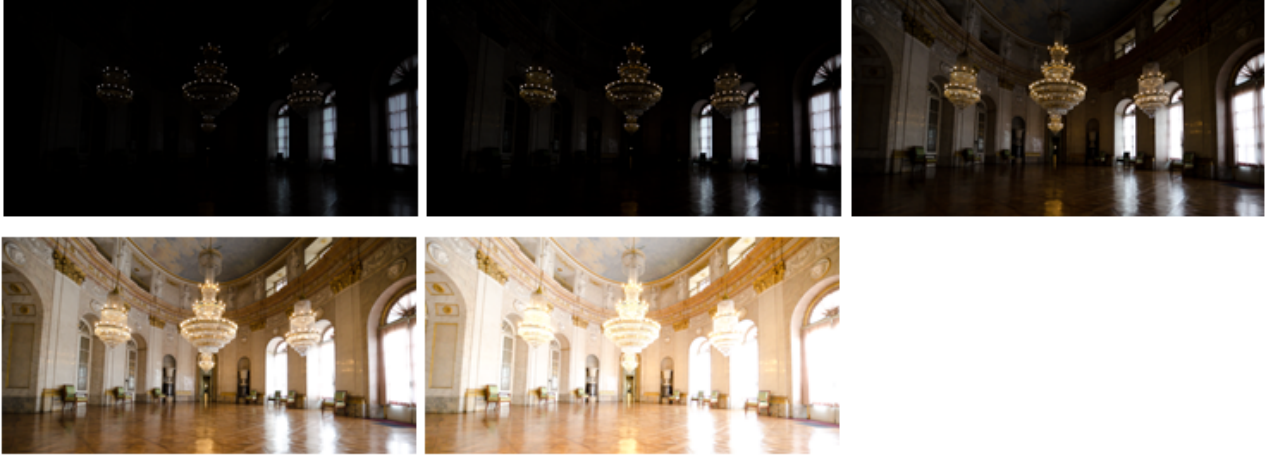


Fig. 15: Imagens com tempo de exposição de 0.0004s , 0.0016s, 0.00625s, 0.025s e 0.1s, respectivamente. Fonte: [11]

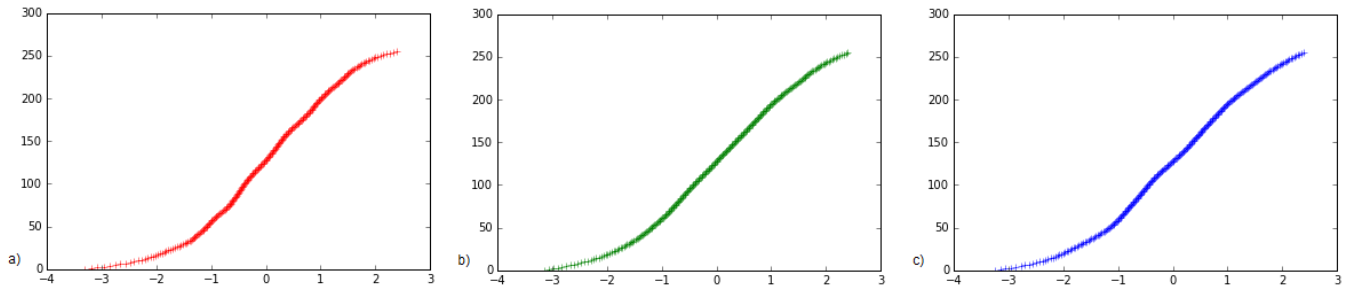


Fig. 16: Função resposta da câmera recuperada para cada canal separadamente: (a) Vermelho com $\lambda = 20$; (b) Verde com $\lambda = 80$; (c) Azul com $\lambda = 50$



Fig. 17: Imagens LDR após a aplicação dos algoritmos de Tone Mapping; (a) Método global (b) Método local.

6 Referências Bibliográficas

- [1] SÁ, A. M., CARVALHO, P. C., VELHO, L. *High Dynamic Range Image Reconstruction*. Primeira Edição. Morgan & Claypool, 2007. 53 p.
- [2] DEBEVEC, P. E., MALIK, J. *Recovering High Dynamic Range Radiance Maps from Photographs*. Berkeley: University of California. 10 p.
- [3] REINHARD, E., WARD, G., PATTANAIK, S., DEBEVEC, P. *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*. San Francisco: Elsevier, 2005. 508 p.
- [4] BOSTOCK, Mike. *Visualizing Algorithms*, 2014. Disponível em: <https://bost.ocks.org/mike/algorithms/>. Acesso em: out. 2016.
- [5] DUAN, J., BRESSAN, M., DANCE, C., QIU, G. *Tone-mapping High Dynamic Range Images by Novel Histogram Adjustment*. Chengdu City: South Western University of Finance and Economics, 2009. 16 p.
- [6] DUAN, J., QIU, G. *Fast Tone Mapping for High Dynamic Range Images*. Nottingham: University of Nottingham. 4 p.
- [7] JESUS, E. O., COSTA R. *A Utilização de Filtros Gaussianos na Análise de Imagens Digitais*. Itajubá: Universidade Federal de Itajubá, 2014. 7 p.
- [8] *K-means Clustering and Vector Quantization*. Disponível em: <https://docs.scipy.org/doc/scipy-0.16.0/reference/generated/scipy.cluster.vq.kmeans2.html>.
- Imagens:
- [9] *Light Probe Image Gallery*, 2004. Disponível em: <http://www.pauldebevec.com/Probes/>.
- [10] *The Jefferson National Expansion Memorial*, 2008. Disponível em: <https://commons.wikimedia.org/wiki/File:StL4.72.JPG>.
- [11] *HDR Pics to play with*, 2013. Disponível em: <http://farbspiel-photo.com/learn/hdr-pics-to-play-with>.