

## 3

### O Tratamento HDR

Nesse capítulo, os trabalhos sobre HDR de Paul Debevec que norteiam essa dissertação são apresentados. O primeiro desses trabalhos apresenta uma forma de gerar imagens *HDR* através de um conjunto de fotografias retiradas com equipamento convencional. As imagens geradas por esse método são utilizadas em diferentes algoritmos de iluminação [34], e no método objeto dessa dissertação. A seguir é apresentado um dos principais trabalhos de iluminação de objetos sintéticos utilizando imagens *HDR*. Por fim, são apresentados os formatos mais utilizados para armazenagem de imagens HDR.

#### 3.1

##### Aquisição de imagens HDR

O trabalho apresentado em [52] é um dos responsáveis pelo crescente interesse em imagens HDR como forma de representação de imagens nos diferentes segmentos da computação gráfica. O método apresentado, se baseia em coerência espacial de pixels e presunções sobre os dispositivos de aquisições de imagens. Seu objetivo principal é a recuperação da curva de resposta característica do filme, que permite através de sua função inversa, associar valores de brilho de pixel à radiância original de cena.

É um método que não apresenta alta complexidade de implementação, porém não possui muita robustez, uma vez que é suscetível a ruídos, pequenas discrepâncias espaciais no conjunto de imagens LDR utilizado como entrada, e aos parâmetros de *data fitting* utilizado. Nas próximas seções serão descritas as técnicas base desse trabalho.

### 3.1.1 Recuperação de curvas de resposta

O processo de recuperação de mapas de radiância descrito em [52] se utiliza de um conjunto de fotografias tiradas com câmeras tradicionais variando-se o tempo de exposição e utilizando uma iluminação estática para a cena. O processo é baseado em correspondência espacial e devido a este fato a câmera é assumida estática.

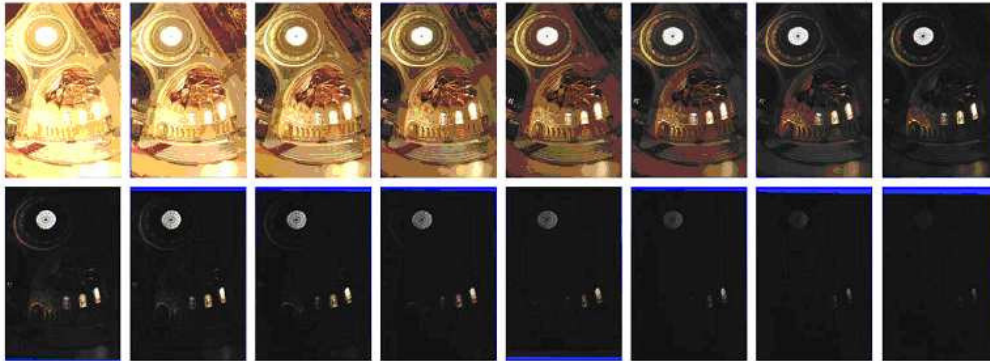


Figura 3.1: Conjunto de imagens com exposição diferentes

O processo básico dos algoritmos para criação de imagens HDR baseia-se na propriedade física dos sistemas de aquisição de imagens chamado reciprocidade.

A resposta de um filme ou dispositivo CCD em relação a variações na exposição da cena é definida por sua curva característica. Esta curva é um gráfico da densidade óptica  $D$  do filme pelo logaritmo da exposição  $E$ .

A exposição  $E$  é o produto da irradiância  $I$  e do tempo de exposição do filme  $\Delta t$ .

$$E = I \times \Delta t$$

é importante salientar que para a curva característica, apenas o produto da irradiância pelo tempo de exposição é importante. Ao dobrar o valor da radiância em um pixel, basta dividir o tempo de exposição da cena para atingir a mesma exposição de cena, de forma que a densidade ótica  $D$  não é afetada. Sob condições extremas de tempos de exposição (tempo de exposição muito grande ou curto demais), a regra de reciprocidade pode ser quebrada. Esse fato é denominado falha de reciprocidade. Em filmes tradicionais, a reciprocidade é mantida para tempos de exposição de 1/10.000 de segundo até 10 segundos. Os CCD encontrados em câmeras digitais seguem a propriedade de reciprocidade uma vez que a medição em cada sensor mede o total de fótons absorvidos durante o tempo de exposição.

Após o armazenamento em câmera digital ou escaneamento e digitalização em câmeras tradicionais, cada pixel é representado por um número binário  $Z$ , tradicionalmente codificado em 8 bits por canal.  $Z$  é o resultado de uma série de mapeamentos não lineares da exposição original  $E$  do pixel na cena.

Esse mapeamento não linear entre o valor do pixel  $Z$  e da exposição original  $E$  é na verdade uma composição da curva característica do filme e um série de funções não lineares introduzidas nos estágios subseqüentes do processo de aquisição das imagens. Seja  $f$  a composição de funções não lineares, definida como uma única função não linear. O primeiro objetivo do algoritmo de criação de mapas de radiância é recuperar  $f$ . Seja:

$$Z = f(E) = f(I\Delta t) \quad (3-1)$$

Dessa maneira, a exposição original do pixel é simplesmente a função inversa de  $f$ :

$$E = f^{-1}(Z)$$

Para que a função inversa de  $f$  exista, é necessário assumir que  $f$  seja bijetora. Assumindo que  $f$  é uma função monotônica e crescente, o que é algo razoável, uma vez que é esperado que valores de exposição maiores resultem em valores de pixel maiores, a obrigação de bijetividade é atingida. Após a exposição  $E$  ser definida, e uma vez que os tempos de exposição das imagens são conhecidos no momento em que as fotos foram tiradas, e conseqüentemente são constantes para o problema, o valor de irradiância  $I$  é recuperado através de

$$I = E/\Delta t \quad (3-2)$$

A entrada para o algoritmo é um conjunto de fotografias digitalizadas tiradas da mesma posição e com tempos de exposição  $\Delta t$  diferentes para cada fotografia no conjunto. Assumindo que a cena é estática e o tempo necessário para a aquisição de imagens é pequeno o suficiente para garantir que mudanças na iluminação da cena podem ser facilmente descartadas ou inexistentes, os valores de irradiância em cada pixel para todas as imagens do conjunto são constantes.

O conjunto de imagens pode ser entendido como um cubo de dimensões  $w \times h \times d$ , onde  $w$  é a largura em pixels da imagem,  $h$  é a altura em pixels da imagem, e  $d$  é o valor de profundidade, diretamente proporcional ao número de fotografias no conjunto. É importante ressaltar que os valores da largura e comprimento são constantes para todas as imagens do conjunto.

Podemos denotar os valores de exposição dos pixels em uma fotografia por  $E_{ik}$ , onde  $ik$  é o índice do pixel na fotografia. Seja  $j$  o índice do tempo

de exposição  $\Delta t$ , ou seja, o índice da fotografia. Podemos reescrever 1 como:

$$Z_{ik}^j = f(I_{ik}\Delta t_j) \quad (3-3)$$

Logo:

$$f^{-1}(Z_{ik}^j) = I_{ik}\Delta t_j$$

Utilizando o logaritmo natural em ambos os lados da equação:

$$\ln f^{-1}(Z_{ik}^j) = \ln(I_{ik}\Delta t_j)$$

$$\ln f^{-1}(Z_{ik}^j) = \ln I_{ik} + \ln \Delta t_j$$

É comum definir  $\ln f^{-1}$  como  $g$  para simplificação de notação. A equação é reescrita como:

$$g(Z_{ik}^j) = \ln I_{ik} + \ln \Delta t_j \quad (3-4)$$

Uma forma mais compacta para a escrita dessas equações pode ser utilizada, definindo-se as imagens como um vetor unidimensional de pixels. Tal fato é bastante útil para a implementação, uma vez que o algoritmo não irá utilizar todos os pixels de uma fotografia e nessa parte da implementação não estamos interessados na localização espacial dos pixels na imagem. (Repare que a correspondência espacial entre pixels nas diferentes imagens é um pre-requisito do método, logo qualquer pixel selecionado obrigatoriamente terá suas contrapartes em outras imagens corretas). Apenas os valores corretos de irradiância os tempos de exposição para imagem são importantes. Dessa forma, o conjunto de equações pode ser reescrito como:

$$Z_{ij} = f(I_i\Delta t_j)$$

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

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

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

$$g(Z_{ij}) = \ln I_i + \ln \Delta t_j \quad (3-5)$$

Onde  $i$  itera sobre os pixels e  $j$  itera sobre os tempos de exposição das imagens.

Nesse conjunto de equações,  $Z_{ij}$  é conhecido, uma vez que representa o valor do pixel na localização  $ij$ . Os tempos de exposição  $j$  são previamente conhecidos. Os valores a serem determinados são a Irradiância em cada

pixel  $I_i$  e a função  $g$ . A tarefa do algoritmo é então recuperar tais valores. É importante salientar que a função  $f$  possui um contra-domínio bem definido, tipicamente dentro dos limites de representação de um *unsigned byte*, entre 0 e 255. Dessa maneira, a função inversa  $f^{-1}$  terá esses limites como seu domínio, e a recuperação da função  $g$  requer apenas a descoberta desse conjunto de valores que  $g(z)$  pode assumir.

Sejam  $Z_{min}$  e  $Z_{max}$ , respectivamente, o maior e o menor valor de pixel possíveis.  $N$  o número total de pixels utilizados pelo algoritmo e  $P$  o número de fotografias. Dessa maneira, o problema pode ser formulado como encontrar os  $|Z_{min} - Z_{max} + 1|$  valores de  $g(z)$  e os  $N$  valores do logaritmo natural da irradiância ( $\ln I_i$ ) que minimizem a função objetivo:

$$O = \sum_{i=1}^N \sum_{j=1}^P [g(Z_{ij}) - \ln I_i - \ln \Delta t_j]^2 + \lambda \sum_{Z_{min}+1}^{Z_{max}-1} g''(z)^2 \quad (3-6)$$

O primeiro termo garante que a solução satisfaz no sentido de mínimos quadrados. O segundo termo atua como um fator de suavização, sendo definido como a segunda derivada de  $g$ . O escalar  $\lambda$  é utilizado para pesar a quantidade de ruído esperada nas medições de  $Z_{ij}$ .

Esse sistema é super determinado e pode ser resolvido através de diferentes métodos, entre eles *singular value decomposition* (SVD). Para ajuste da curva [52] utiliza dois métodos, o primeiro consiste em obrigar que o meio da curva de resposta tenha exposição unitária. O meio da curva de resposta é  $Z_{mid} = \frac{1}{2}(Z_{max} + Z_{min})$ . Definindo  $g(Z_{mid} = 0)$ , obtemos  $e^{Z_{mid}} = 1$ . O mais importante porém é que as medições feitas próximas a  $Z_{min}$  and  $Z_{max}$  devido a problemas de pouca e super exposição respectivamente, sejam menos confiáveis que os valores da região central da curva. Dessa maneira, um segundo ajuste, que penaliza tais valores na reconstrução das curvas de resposta.

A equação 3-6 se torna:

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

Por último, não existe a necessidade de utilização de todos os pixels da imagem, na verdade, utilizar um número muito grande de pixels como entrada para o algoritmo, degrada significativamente a performance. Para

encontrar uma solução, com o limite tradicional entre  $[0..255]$  para  $Z_{min}$  and  $Z_{max}$ , utilizando 11 imagens, 50 pixels é um número adequado. Finalmente, a localização dos pixels na imagem é importante uma vez que pixels sub ou super expostos fornecem resultados menos confiáveis para a recuperação de  $g$ . É importante utilizar uma boa distribuição de pixels entre  $Z_{min}$  and  $Z_{max}$  para que a curva seja o mais exata possível.

### 3.1.2

#### Criação de mapas de radiância

Uma vez que a curva de resposta é encontrada, a tarefa seguinte é converter cada valor de pixel para os valores de radiância correspondentes, fornecendo os tempos de exposição  $\Delta t$  conhecidos. É interessante mencionar que a curva de resposta  $g$  pode ser utilizada para criar mapas de radiância para qualquer conjunto de imagens capturada como o mesmo sistema utilizado cuja curva de resposta foi recuperada. Para construir o mapa de radiância, a função  $g$  é aplicada aos valores de brilho de pixel  $Z_{ij}$ , associado ao tempo de exposição  $\Delta t$  correspondente.

Da equação 3.1.1 temos:

$$\ln I_i = g(Z_{ij}) - \ln \Delta t_j$$

Para maior robustez, todos os tempos de exposição disponíveis no conjunto de imagens devem ser utilizados como tentativa de redução de ruído nos valores de radiância recuperados. Uma vez que no conjunto provavelmente existem pixels sub expostos ou saturados, é necessário associar diferentes pesos de acordo com a confiança no valor do pixel associado ao tempo de exposição. A função de pesagem utilizada na recuperação da curva de resposta é reutilizada para essa tarefa. Pesos maiores serão dados a exposições em que o valor do pixel é mais próximo ao meio da curva de resposta, penalizando dessa forma valores saturados e sub expostos.

Assim, a construção do mapa de radiância é definida por:

$$\ln I_i = \frac{\sum_{j=1}^P w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^P w(Z_{ij})} \quad (3-7)$$

para cada pixel  $i$  na imagem. Os mapas de radiância construídos com esse método possuem a propriedade de representarem valores proporcionais as condições de iluminação encontradas no ambiente onde as fotografias

foram retiradas. A partir dessa propriedade, fundamental para os métodos apresentados nos capítulos seguintes, imagens geradas por esse método serão utilizadas na iluminação das cenas renderizadas.

## 3.2

### **Renderização de objetos sintéticos utilizando mapas de iluminação HDR**

A simulação dos efeitos da iluminação direta e indireta é o maior interesse da pesquisa em algoritmos de iluminação global. O método apresentado em [53] é uma das principais motivações para o trabalho apresentado nessa dissertação. Seu foco principal é a integração de objetos sintéticos em cenários do mundo real capturados em fotografias. Devido ao realismo necessário para que um observador humano não percebesse a diferença entre a cena real e os objetos adicionados, variações desse método em tempo real, que permitam renderizações de cenário foto-realistas são de profundo interesse. Os fundamentos técnicos desse método são mais detalhadamente explicados no Capítulo 5.

A renderização de objetos sintéticos em ambientes reais apresenta inúmeras dificuldades. A principal necessidade é de que objetos sejam integrados de forma perfeita a cena, dando a impressão a um observador que o objeto é parte da cena real. Para atingir tal resultado é necessário que os objetos sejam renderizados de forma a simular o comportamento que sua superfície exibiria caso estivesse na cena, e que as condições de iluminação do ambiente: posição das luzes e sua interação com objetos próximos. Técnicas anteriores se baseiam na busca manual da posição das luzes na cena e adição de luzes sintéticas na cena. Tais técnicas, geralmente não apresentam resultados muito satisfatórios.

Dispondo de um algoritmo de iluminação global, caso uma cena possua as informações geométricas e todas características de reflectância (BRDF) das superfícies na cena, é possível adicionar um objeto sintético ao modelo e recalcular a solução de iluminação global. Infelizmente, a obtenção de modelos geométricos e de reflectância completos para grandes ambientes é extremamente complexa. Além disso, soluções de iluminação global são computacionalmente muito pesadas.

A aquisição de informação sobre as condições de iluminação em um determinado ambiente é difícil, uma vez que estas apresentam grandes variações entre áreas muito iluminadas e pouco iluminadas, devido ao fato das fontes de luz serem normalmente concentradas. Como é fundamental

que tanto a luz indireta, refletida pelo ambiente, quanto as fontes de luz sejam capturadas de forma precisa para a solução de iluminação, a utilização de imagens HDR se torna uma obrigação. A seção anterior desse capítulo descreve uma técnica para criação de imagens HDR a partir de um conjunto de imagens tradicionais, tal técnica será utilizada para a geração de mapas de iluminação do ambiente.

### 3.2.1

#### Método geral

O método para adição de objetos sintéticos em cenas capturadas divide a cena em três partes componentes principais: Cena distante, cena local e objetos sintéticos. Tal separação é feita, principalmente devido ao fato de cada uma dessas regiões possuírem requisitos de detalhamento geométrico e complexidade da descrição dos modelos de reflectância das superfícies.

##### 1. Cena distante

A cena distante é contruída como um modelo de iluminação. Os objetos sintéticos irão receber, direta e indiretamente as luzes desse modelo. Dessa forma, é necessário que a cena distante armazene medições de radiância, ao invés de valores de brilho LDR dos pixels de imagens convencionais. Para a composição final da cena, esse modelo pode ser representado com pouca geometria indo até um modelo geométrico criado através de escaneamento 3D do ambiente .

A obrigação fundamental da cena distante é fornecer medições precisas da iluminação incidente nos objetos e também em uma direções particulares. Durante o cálculo da solução de iluminação global, a cena distante irradia luz em direção aos objetos e a cena local, porém a luz refletida que retorna a ela é ignorada. Tal fato é possível devido a nenhuma área da cena distante ser significativamente alterada pela luz refletida. Devido a este fato, não é necessário prover informações de reflectância para a cena distante, além disso, essa separação facilita o cálculo da solução de iluminação global. Caso a luz refletida seja significativa em alguma parte, esta deve ser movida para a cena local e ser provida de informação de reflectância necessária (BRDF) para que possa interagir com a luz.

##### 2. Cena local

A cena local consiste das superfícies que irão interagir fotometricamente com os objetos sintéticos e receber iluminação da cena distante. É a geometria onde os objetos irão projetar sombras



e refletir luz. A cena distante é parte fundamental da solução de iluminação global, e portanto sua geometria e características de reflectância precisam ser conhecidas, ao menos aproximadamente. No caso geral, a geometria da cena local é uma superfície plana onde os objetos se encontram e onde os objetos sintéticos serão posicionados.

No caso da cena local ser predominantemente difusa, a influência dos objetos, diminui proporcionalmente ao inverso do quadrado da distância entre eles. Mesmo assim, existe uma série de circunstâncias onde os objetos sintéticos influenciam partes mais distantes da cena. Tais circunstâncias são: Uma vez que a cena local apresenta um modelo de reflectância aproximado, ela pode ser adicionada ao problema de iluminação global da mesma maneira que qualquer objeto sintético.

3. **Objetos sintéticos** Os objetos sintéticos consistem dos objetos que serão adicionados à cena capturada. Esses objetos recebem iluminação direta proveniente da cena distante e indireta proveniente da cena local e de outros objetos sintéticos. Tais objetos precisam interagir perfeitamente com as condições de iluminação da cena e tratar reflexões de objetos próximos, por isso é necessária a utilização de um modelo de reflectância completo.

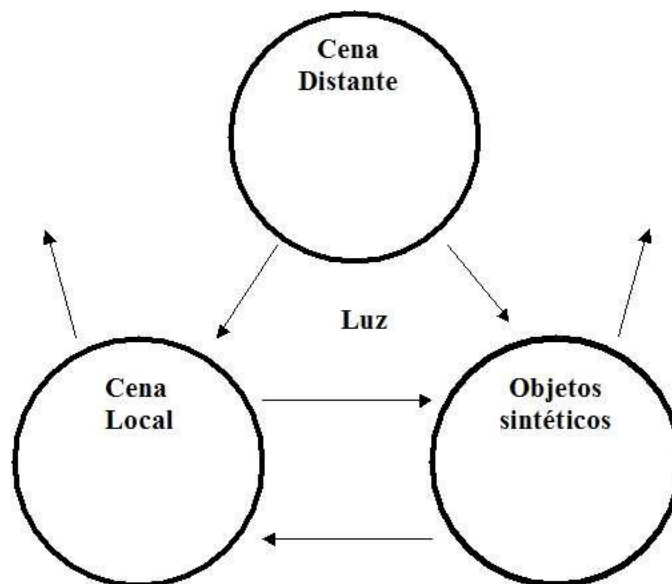


Figura 3.2: Método para adição de objetos sintéticos

### 3.2.2

#### **Captura de iluminação ambiente**

Como descrito anteriormente, a cena distante deverá ser corretamente refletida nos objetos sintéticos. Como a luz incide de todas as direções no objeto, é necessário que um mapa de radiância HDR omnidirecional seja criado.

A técnica mais difundida para captura das condições de iluminação omnidirecionais de um ambiente é a utilização de esferas perfeitamente espelhadas, como uma bola de aço polido. A esfera é posicionada perto da localização onde se deseja adicionar os objetos sintéticos.

Um conjunto de fotografias com diferentes tempos de exposição são tiradas para a construção do mapa de radiância como descrito na seção 3.1.2.

As medidas de radiância observadas na esfera são mapeadas na geometria utilizada para representar a cena distante. Na maioria dos casos esse modelo pode ser bastante simples. Geralmente volumes em forma de cubo ou esfera são posicionados de forma a englobar toda a geometria da cena. A imagem gerada por uma única esfera pode exibir artefatos visíveis. Primeiro, a camera, e possivelmente o fotógrafo serão visíveis. A esfera ao capturar a cena, interage com ela, pode aparecer em reflexões de outros objetos na cena, gerar sombras e refletir luz para outras superfícies. Tomando um certo cuidado no posicionamento da esfera e da câmera, tais efeitos podem ser minimizados e não serem perceptíveis na renderização final. Caso os artefatos sejam significativos, estes podem ser retirados com programas de processamento de imagens como o Adobe Photoshop. Outra maneira de reduzir artefatos é combinar imagens da esfera em diferentes posições e selecionar as áreas melhor representadas. Uma descrição mais detalhada desse processo é apresentada no Capítulo 5.

### 3.2.3

#### **Mapeando mapas de iluminação à cena**

O próximo passo é determinar o mapeamento entre as coordenadas na imagem e os raios do mundo que as originaram. Para essa determinação, é necessário que a posição da bola no mundo seja conhecida, o tamanho da bola, e os parâmetros da câmera, como posição e distância focal. Com essa informação, é possível traçar raios partindo do centro da câmera em direção ao pixels da imagem e refletir raios partindo da esfera para o ambiente. A informação gerada pelo traçado de raios representa a iluminação fornecida

pela cena distante na solução de iluminação global a ser computada. Esse mapeamento depende do tipo mapa de ambiente utilizado, esféricos ou *cube maps*. Mapas esféricos são gerados automaticamente pelo processo descrito anteriormente, porém apresentam problemas de resolução nos polos. *Cube maps* precisam ser gerados a partir do mapa esférico, mas possuem melhor precisão para qualquer direção do mundo a ser mapeada, e apresenta suporte de hardware gráfico, o que é especialmente importante para as aplicações em tempo real.

### 3.2.4

#### Renderização da cena

A partir da geração dos mapas de iluminação e do mapeamento em uma superfície representante do ambiente, esse dados são enviados ao sistema de renderização com suporte a iluminação global. Tipicamente, tais sistemas são baseados no algoritmo de *ray-tracing* e iluminação global. Os valores de iluminação para cada pixel é derivado pelo vetor de reflexão da câmera em um ponto  $p$  da superfície do objeto a ser renderizado. A forma como algoritmos de iluminação global computam a solução de iluminação é mostrada no Capítulo 5. Após a renderização dos objetos sintéticos, a cena pode ser processada para adicionar efeitos de *motion blur*, *Bloom*, *Depth of field*.

### 3.3

#### Codificação de imagens HDR

Uma das questões mais importantes relacionadas à imagem digital, é a forma de armazenagem. Tal fato é especialmente verdadeiro para imagens HDR, que devem gravar valores muito além das capacidades da codificação de 24 bits padrão. Uma boa codificação de imagens HDR é particularmente importante, pois é muito fácil fazer com que tais imagens ocupem uma grande quantidade de memória, muitas vezes comprometendo o espaço disponível em discos rígidos, fato corriqueiro para edição de vídeo HDR. Felizmente, um conjunto de formatos para lidar com essas imagens tem sido desenvolvido pela comunidade de computação gráfica. Uma codificação diz respeito à representação dos bits componentes de um pixel, por exemplo, oito bits por canal de cor com três canais de cor por pixel. Um formato inclui qualquer informação, que associada ao conjunto de dados da codificação dos pixels, permite que uma imagem seja lida e posteriormente exibida.

Nessa sessão, os tipos mais comuns de codificação de imagens HDR serão apresentados, junto com seus formatos associados.

### 3.3.1

#### Características de codificação HDR

A quantidade de bits é um dos fatores mais marcantes em relação as diferenças entre codificações de imagens LDR em relação à imagens HDR. Porém, tal fato não é o único. Especificamente, uma imagem LDR tradicional, com 24 bits por pixel, é normalmente classificada como *output-referred*, uma vez que suas cores são associadas à algum dispositivo de visualização. Por sua vez, imagens HDR são definidas como *scene-referred*, devido ao fato que seus pixels possuem relação direta com a radiância exibida nas cenas. Tal fato é coerente com a fato que dispositivos de exibição são tipicamente LDR, enquanto as cenas do mundo são HDR. Valores LDR apresentam pixels cujos valores foram saturados e conseqüentemente não podem ser utilizados para referir ao contexto de radiância numa cena. De outra maneira, imagens HDR podem ser associadas a um sistema dedicado a um dispositivo de exibição em particular, porém isso não faz muito sentido, uma vez que valores HDR sempre podem ser mapeados para exibição em um dispositivo em particular. Porém a conversão inversa não é possível.

A armazenagem de imagens HDR deve ser feita de forma a permitir que os valores sejam armazenados sem que haja perda significativa de precisão. Existem diferentes formatos para armazenagem para tais imagens. Todos os formatos que serão apresentados a seguir possuem em comum o fato de cada pixel ser representado em ponto flutuante. Tal representação permite uma maior precisão para valores de pixels, permitindo um maior domínio de valores, tipicamente  $1.17549435e^{-38}$  a  $1.17549435e^{38}$  para 32 bits por canal em formato IEEE 754. Outra importante questão associada a codificação HDR se refere a quantização dos valores de pixels. Caso a distância entre valores de cor adjacentes seja muito grande, artefatos são gerados durante a visualização, degradando o resultado final. Dessa forma, a precisão dos formatos deve ser utilizada não apenas para permitir o maior espectro de valores possível, mas também permitir bons níveis de quantização dos dados.

### 3.3.2

#### Formatos

Os três formatos apresentados nessa são o formato IEEE RGB, o formato RGBE e o formato OpenEXR. Suas características, finalidades, vantagens e desvantagens serão descritos a seguir.

#### 3.3.2.1

##### IEEE 96-bit TIFF

O formato Tagged Image File Format (.tiff) inclui uma codificação com 32 bits por canal em ponto flutuante IEEE [10], a cerca de dez anos. Essa formato é de certa forma, o mais poderoso formato para representação de imagens HDR. Consegue cobrir cerca de 79 ordens de magnitude com variações minúsculas, praticamente eliminando qualquer artefato resultante de quantização.

O maior problema desse formato está no espaço necessário para armazenar imagens. Com 32 bits por canal de cor, utilizando três canais de cor, cada pixel ocupa 96 bits. Devido ao fato dos 12 últimos bits da mantissa contém ruído aleatório, mesmo algoritmos de compressão entrópica avançada conseguem apenas cerca de 10% de compressão. A biblioteca padrão TIFF nem ao menos tenta comprimir esse formato.

Mesmo assim, esse formato apresenta vantagens como fácil leitura e escrita de arquivos. Esse formato é muito bom para leitura de *framebuffers* em ponto flutuante e posterior manipulação sem perdas. Também, esse formato serve de comparação para outros formatos de codificação que utilizem menos bits por pixel.

#### 3.3.2.2

##### Formato RGBE

O formato Red-Green-Blue-Exponent (RGBE)[27] ou HDR, foi introduzido como parte do sistema de simulação de iluminação e renderização Radiance[32]. Essa iniciativa de Greg Ward Larsson para desenvolver um formato de imagem que armazenasse valores reais de radiância para ser utilizado no sistema e que não tivesse um grande custo de armazenagem. O arquivo consiste de um pequeno cabeçalho ASCII, seguido por uma *string* que define o tamanho da imagem, seguido pelos dados da imagem, tipicamente utilizando um algoritmo de *Run length encoding* para comprimir os dados. O formato RGBE se utiliza do

conceito de expoente compartilhado associado aos canais de cor. As duas codificações principais dos pixels são 4 bytes representando os valores RGB, ou XYZ e o expoente compartilhado.



Figura 3.3: Formato RGBE

Os componentes RGBE são gerados pelos componentes RGB de uma imagem LDR através de:

$$E = \lceil \log_2 (\max(R, G, B)) + 128 \rceil$$

$$R_{hdr} = \left\lfloor \frac{256R}{2^{E-128}} \right\rfloor$$

$$G_{hdr} = \left\lfloor \frac{256G}{2^{E-128}} \right\rfloor$$

$$B_{hdr} = \left\lfloor \frac{256B}{2^{E-128}} \right\rfloor$$

A conversão reversa, de valores RGBE para pixels LDR é:

$$R = \frac{R_{hdr} + 0.5}{256} 2^{E-128}$$

$$G = \frac{G_{hdr} + 0.5}{256} 2^{E-128}$$

$$B = \frac{B_{hdr} + 0.5}{256} 2^{E-128}$$

A conversão de valores RGBE para XYZE e vice-versa são as mesmas conversões de valores LDR tradicionais. Ao utilizar a codificação XYZE, todo o espectro de cores visível é suportado. O *dynamic range* dessa codificação é grande (cerca de 76 ordens de magnitude) e a precisão é suficiente para a grande maioria das aplicações. O formato RGBE utiliza apenas 32 bits por pixel para representar uma imagem, de forma que seu custo é apenas um terço maior que imagens RGB tradicionais, que apresentam 24 bits por pixel. Porém, devido a codificação mantíssa-expoente, é necessário um gasto maior de processamento para a leitura

dessas imagens. Esse formato permite cerca de 25% de compressão ao utilizar métodos de *run length encoding* [42]. O que permite um tamanho final compatível com uma imagem RGB com nenhuma forma de compressão. Além disso, devido a facilidade de leitura e escrita desse formato o torna um dos formatos preferidos para aplicações de renderização em tempo real.

### 3.3.3

#### Formato OpenEXR

O formato OpenEXR[11], significa EXtended Range (.exr) e foi desenvolvido pela Industrial Light and Magic inicialmente para melhorar a qualidade das imagens em filmes. O formato se tornou público em 2002 através de uma biblioteca C++, permitindo fácil integração a outros programas. É um dos formatos mais explorados para renderizações baseadas em HDR, seu formato interno de representação é compatível com o tipo half, utilizado pelas arquitetura de placas gráficas modernas com suporte a linguagem Cg [43]. Esse formato é também conhecido como S5E10, que representa um bit para sinal, cinco para expoente e dez para a mantissa. A biblioteca do formato OpenEXR também suporta 32 bits por canal (96 bits por pixel) e 24 bits por canal (72 bits por pixel) em ponto flutuante. O formato de 32 bits por canal é o mesmo formato IEEE TIFF, e o formato de 24 bits, introduzido pela Pixar ainda não é muito difundido.

A codificação padrão do formato OpenEXR utiliza floats de 16 bits para representar cada um dos canais de cor e mais um canal de transparência (RGBA), resultando em 48 bits para os canais. A organização dos bits para essa codificação é esquematizada na Figura 3.4

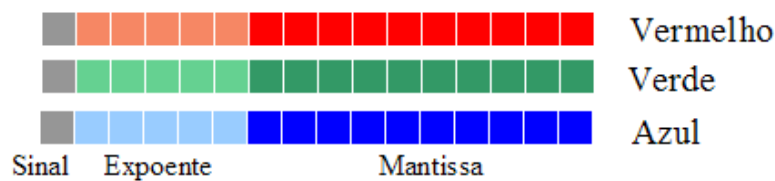


Figura 3.4: Codificação dos bits no formato OpenEXR. um bit para sinal, cinco para expoente e dez para mantissa

Esse formato consegue representar cerca de 10.7 ordens de magnitude, porém seu passo de quantização é de apenas 0.1%, muito abaixo do limiar da visão humana, permitindo assim extensiva manipulação dessas imagens antes que qualquer artefato se torne evidente. Isso significa que o OpenEXR consegue representar valores de cor adjacentes com um precisão muito

grande, ou seja, uma variação muito pequena entre seus valores. Esse fato é um dos maiores motivadores da utilização desse formato para geração de imagens de alta qualidade para a indústria cinematográfica. Esse formato permite ainda compressão de até 40% utilizando-se wavelets[42].

### 3.3.4

#### **Formato para aplicações tempo real sem suporte a texturas em ponto flutuante**

Em aplicações tempo real que não possuem suporte à texturas em ponto flutuante um esquema diferente pode ser utilizado. Com essa técnica[51], imagens de 8 bits são utilizadas para representar uma parte do valor de radiância do pixel. Durante a renderização as imagens podem ser combinadas utilizando técnicas de register combiners ou pixel shaders. Essa técnica também pode ser utilizada com outros formatos inteiros de maior precisão, por exemplo o formato HILO que apresenta 16 bits por canal de cor, com 2 canais de cor por pixel. Como esse formato apresenta maior precisão, os resultados finais são melhores.