

Explorando a adaptação do *Sistema Visual Humano* em aplicações interativas através de *Tone Mapping*

Relatório de Andamento do
Trabalho de Conclusão – UNISINOS

Fábio Mierlo

Orientador Marcelo Walter

Dezembro de 2004

Sumário

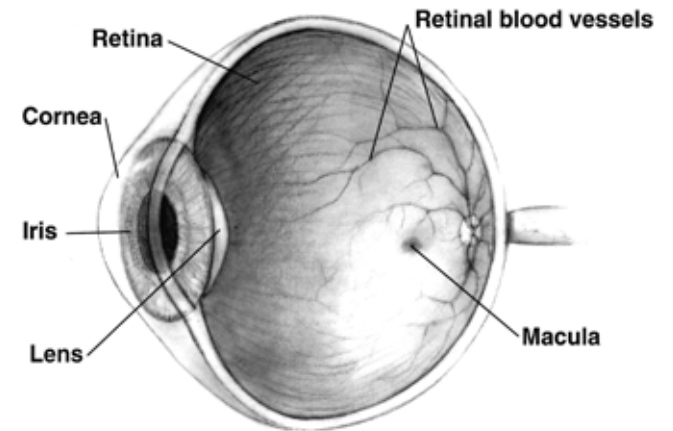
- Introdução
- Motivação
- Objetivos
- Estado da Arte
- Resultados Preliminares
- Conclusões
- Cronograma

Introdução

- HVS
- RGB
- Ray Tracing e Radiosidade
- HDR
- Tone Mapping

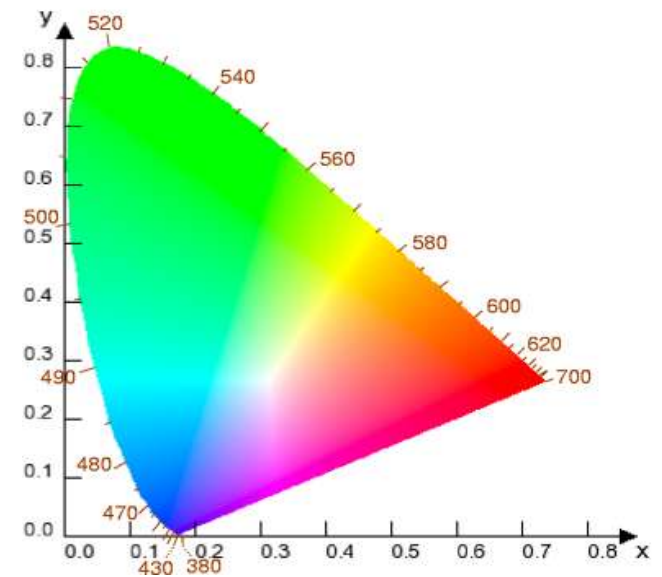
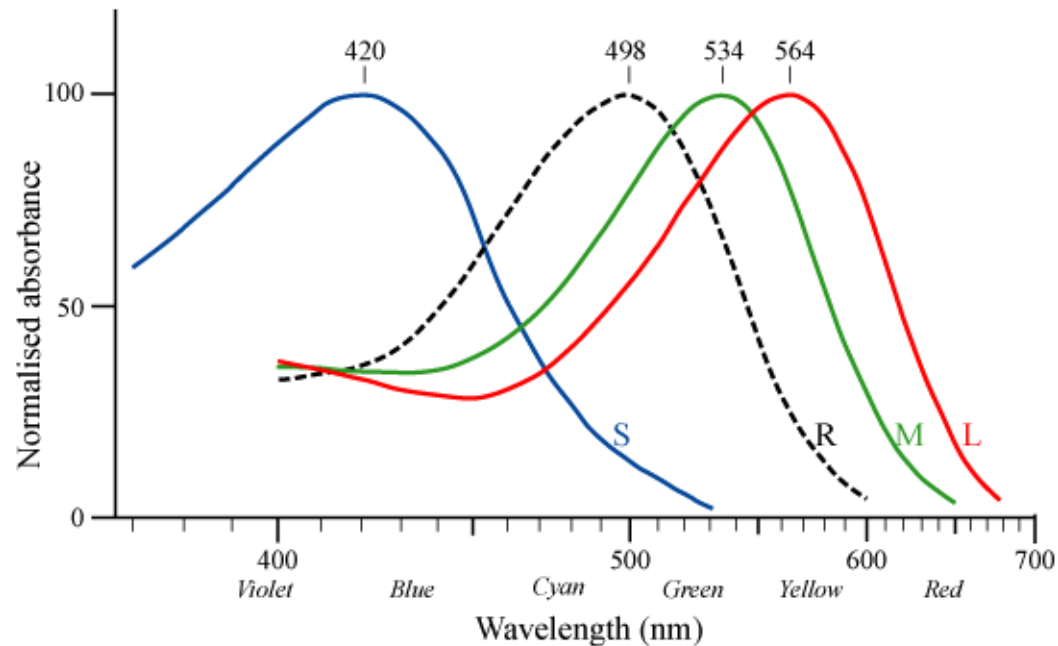
HVS

- Cones e bastonetes
- Função logarítmica
- 1.000.000.000:1 cd/m²
- Adapta-se através de um intervalo dinâmico



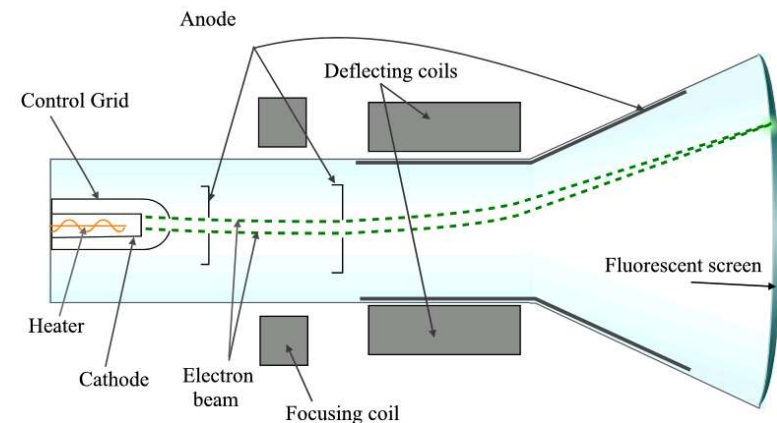
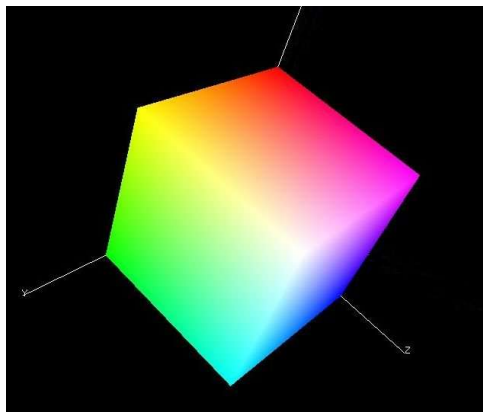
HVS

- Três tipos de cones
- Sensíveis às cores vermelha, verde e azul



RGB

- Representado por um cubo com valores RGB nos eixos ortogonais dentro do intervalo de 0 à 1
- Imagens sintetizadas são visualizadas em monitores CRT



Ray Tracing e Radiosidade

- RGB é limitado por não poder representar muitas cores e a luminância depender da intensidade do monitor
- 100:1 cd/m² e 8 bits por componente
- RT e Radiosidade acumulam iluminação
- Facilmente ultrapassa os 255 valores entre 0 e 1

HDR

- A solução foi aumentar o intervalo dinâmico com números de ponto flutuante
- Diminui erros de arredondamento
- Permitindo maior precisão
- Estende o sistema RGB além do domínio dos números entre 0 e 1

Tone Mapping

- Não é possível visualizar imagens HDR diretamente em monitores CRT
- TM foi desenvolvida para fotografias e televisão
- Comprime o intervalo de luminância
- Imita qualidades perceptuais
- TM é computacionalmente caro
- São classificados como *global* ou *local*
- E podem ser dependentes de tempo

Motivação

- *HDR e Tone Mapping* pertencem ao estado da arte em computação gráfica
- GPUs permitem utilizá-las interativamente
- Aumentar o realismo e a percepção visual em ambientes virtuais através destas tecnologias

Objetivos

- Implementar em GPU um algoritmo de *Tone Mapping* dependente de tempo
- Explorar o potencial uso da adaptação do HVS em aplicações interativas

Estado da Arte

- Debevec
- GPU
- HDR Displays
- Tone Mapping

Debevec

- Em 97 propôs um método para aquisição de imagens com HDR através de fotografias
- Imagens com valor do pixel proporcional ao valor de luminosidade
- Softwares foram estendidos para utilizar estas imagens
- Para estas aplicações executarem interativamente as últimas GPUs foram projetadas para renderizar com HDR

Debevec



Image-Based Lighting, Debevec (98)

GPU

- Pipeline de 128-bits
- Operações aritméticas em ponto flutuante
- Diminui erros de arredondamento
- Processadores de stream
- Suporte limitado a desvio condicional

HDR Displays

- Imagens HDR serão necessárias
- Dois anos como tecnologia emergente
- 40.000:1 cd/m² contra 100:1 cd/m²



HDR Displays, Seetzen (04)

Tone Mapping Global

- Um modelo de TM global simples é o mapeamento linear
- Preservam o contraste relativo, mas removem os detalhes
- Vários modelos foram propostos
- Com características de preservar o brilho, preservar o contraste, se basear na percepção humana, utilizar compressão logarítmica, ajuste de histograma, entre outros

Tone Mapping Local

- Globais são computacionalmente eficientes, porém dificilmente preservam o contraste local
- Modelos locais utilizam um mapeamento espacial variável
- Pode mapear valores diferentes de luminâncias idênticas de acordo com seus vizinhos

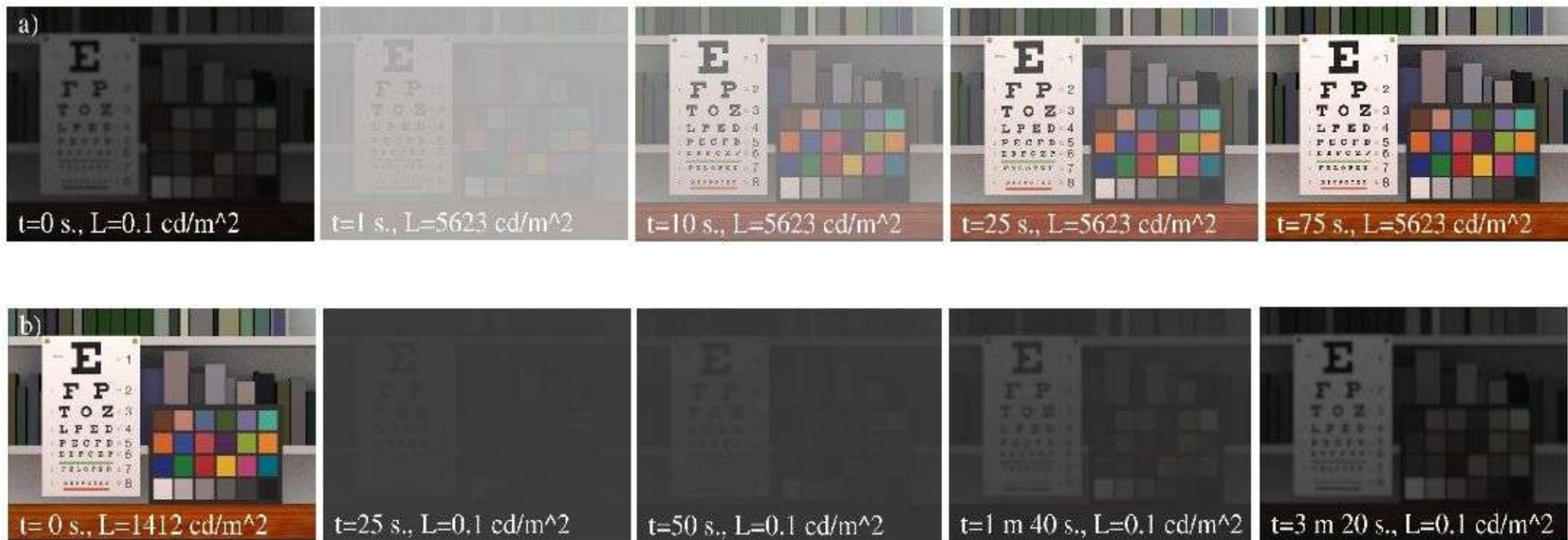
Tone Mapping Local

- Fattal et al. (02), atenuação da magnitude de gradientes
- Reinhard et al. (02), técnica fotográfica chamada “Zone System”
- Durand e Dorsey (02), filtro bilateral para decompor a imagem em duas camadas
- Ashikhmim (02), método de três passos para imagens com alto contraste onde detalhes são reintroduzidos no passo final

Tone Mapping Dependentes de Tempo

- Ferwerda et al. (96), modelo perceptual que considera mudanças em limiares de visibilidade e sensibilidade temporal
- Pattanaik et al. (00), modelo psicofísico através de modelos de aparência
- Durand e Dorsey (00), método interativo simula a adaptação cromática
- Ledda et al. (04), modelo local baseado em filtros bilaterais e dados fisiológicos

Tone Mapping Dependentes de Tempo



Adaptação em função do tempo, Ferwerda (96)

Tone Mapping GPU

- Schell et al. (00), modelo de Ward Larson utilizando texturas em hardware
- Goodnight et al. (03), modelo de Fattal em GPU, porém não interativo
- Goodnight et al. (03), incluiu dependência de tempo no modelo de Reinhard executando interativamente
- Artusi et al. (03), framework híbrido entre CPU e GPU para acelerar a execução

Resultados Preliminares

- Critérios de seleção
- Comparativo entre os modelos
- Seleção do modelo
- Aplicação prática

Critérios de Seleção

- Dependente de tempo
- Local
- Computacionalmente eficiente
- Adaptativo
- Disponibilidade de código fonte (critério não exclusivo)

Comparativos entre os Modelos

- Dependentes de Tempo
 - Ferwerda 96 : Eficiente
 - Pattanaik 00 : Eficiente
 - Durand 00 : Eficiente
 - Ledda 04 : Local



Adaptação local, Ledda (04)

Comparativos entre os Modelos

- Não Dependentes de Tempo
 - Fattal 02 : Local
 - Ashikhmim 02 : Local, Eficiente
 - Durand 02 : Local
 - Reinhard 02 : Local, Eficiente, Adaptativo

Seleção do Modelo

- *Tone Mapping*
 - Reinhard (02) atende a todos os critérios, tem o código fonte disponível e já foi implementado em GPU
- Dependente de Tempo
 - Pattanaik (00) não é limitado a cenas estáticas e não adota uma solução simples em busca de interatividade

Aplicação Prática

- Utilizar a capacidade adaptativa em função do tempo como nível de dificuldade em jogos
- Em um jogo de corrida de carros, ao entrar em um túnel o jogador ficaria temporariamente sem visão até ele estar adaptado

Aplicação Prática



Não adaptado



Adaptado

Conclusões

- Não era de conhecimento que TM podia ser executado interativamente utilizando GPUs
- Nem todos os problemas podem ser mapeados para a arquitetura das GPUs

Conclusões

- Esperava-se gerar os critérios apenas com a comparação entre os modelos de TM, porém alguns foram estabelecidos através do estudo das GPUs
- Nenhum dos modelos atendeu a todos os critérios, sendo necessário achar uma solução para este problema

Conclusões

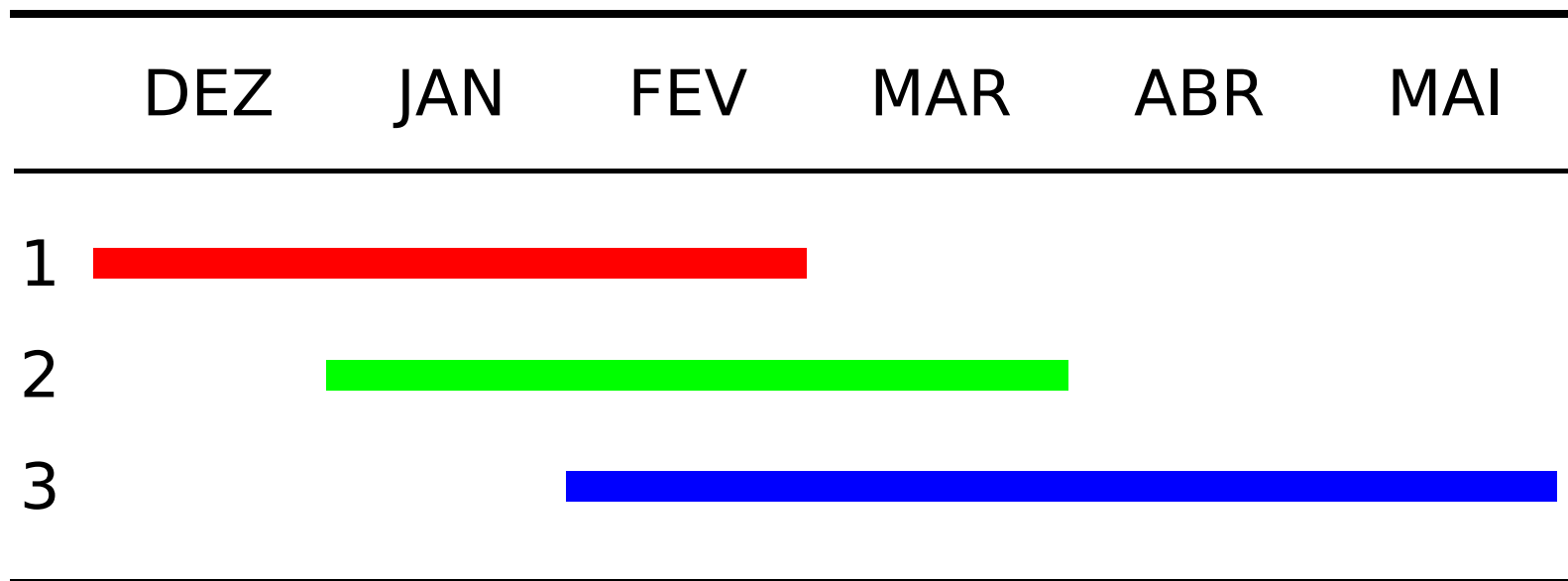
- GPUs: o trabalho deve ser desenvolvido utilizando a API OpenGL
- Arquitetura alvo: placas gráficas compatíveis com o padrão DirectX 9.0
- Talvez seja necessário modificar os modelos selecionados devido as limitações desta categoria de hardware
- O modelo deve permitir controle entre desempenho e qualidade

Discussão

- É necessário utilizar um modelo dependente de tempo que cause uma impressão próxima a realidade em jogos?
- O que é mais importante neste trabalho o modelo de TM ou o modelo de adaptação visual?

Cronograma Atualizado

1. Implementação do protótipo
2. Testes e validações do protótipo
3. Redação da monografia



Zone System

- Zone: é o range de luminância em uma cena. Existe 11 zonas 0 (preto) até X (branco), cada uma dobra a intensidade da anterior
- Middle-grey: zona V, metade do brilho de uma cena. Tipicamente medida com um cartão cinza com 18% de reflexão
- Key: indica se a cena é clara, normal ou escura. Pintura branca seria uma high-key

Zone System

- Dynamic range: razão entre a luminância mais alta e mais baixa. Para fotógrafos é a diferença entre a zona distinguível mais alta e mais baixa
- Dodging-and-burning: retirar e adicionar luz à uma região da cena
- Obs: em fotografia utiliza-se leitura de luz de zonas claras, normais e escuras, para determinar estes parâmetros

Seleção do Modelo

- Reinhard (02)

- Log médio de luminância é o key da imagem

$$\bar{L}_w = \exp \left(\frac{1}{N} \sum_{x,y} \log (\delta + L_w(x,y)) \right)$$

- Log médio é mapeado para o middle-grey zone

$$L(x,y) = \frac{a}{\bar{L}_w} L_w(x,y)$$

- Então é aplicado um TM global simples

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

Seleção do Modelo

- Técnica para melhorar o contraste local
 - Função Center-surround para medir o contraste local em uma escala dada

$$Activity(x, y, s_i) = \frac{V(x, y, s_i) - V(x, y, s_{i+1})}{2^{\phi} a / s_i^2 + V(x, y, s_i)}$$

- Substitui $L(x, y)$ por $V(x, y, s_m)$ no denominador do TM global simples

$$L_d(x, y) = \frac{L(x, y)}{1 + V(x, y, s_m)}$$