

Dispositivos de Saída e Formação de Imagens

SCC0250 - Computação Gráfica

Prof. Rosane Minghim

[https://sites.google.com/site/computacaograficaicmc2017t2/
rminghim@icmc.usp.br](https://sites.google.com/site/computacaograficaicmc2017t2/rminghim@icmc.usp.br)

P.A.E. Nicolas Roque *nrsantos@usp.br*

Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP)

baseado no material de anos anteriores, vários autores

19 de março de 2017



Sumário

1 Dispositivos de Saída (Exibição)

2 Formação de Imagens

Sistema Computacional

- Um **sistema computacional** normalmente é composto por
 - Processador
 - Memória
 - **Dispositivos de saída (sistema gráfico)**
 - Dispositivos de entrada

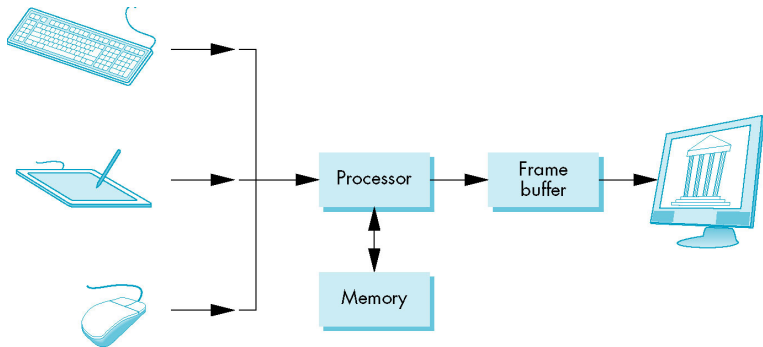
Computação Gráfica: Dispositivos de Exibição

- De forma geral existem dois tipos de **dispositivo de exibição**: dispositivo **vetorial** e dispositivo **matricial**

- Dispositivos vetoriais (“vector”)
 - natureza analógica
 - imagens formadas por segmentos de reta
 - geradas a partir de “display files”

- Dispositivos matriciais (“raster”)
 - natureza digital
 - imagens formadas pelo preenchimento de matriz de “pixels”
 - geradas a partir de “frame-buffers”

Sistema Gráfico Matricial (Raster)



Geração da Imagem em Dispositivo Matricial

- Descrição da **cena** mantida no **frame-buffer**, que contém uma posição associada a cada pixel da tela
- Para cada **pixel**, o valor armazenado na posição correspondente define a **intensidade (ou cor)** com que o pixel será traçado
- Todos os objetos são pixels

Frame Buffer

- Resolução: número de pixels
- Implementado c/ memória VRAM/DRAM/DDR/GDDR
 - Video random-access memory
 - Dynamic random-access memory
 - Acesso rápido para re-exibição e restauro
- frame buffer pode armazenar outras informações além da cor do pixel
 - múltiplas camadas, ou múltiplos buffers

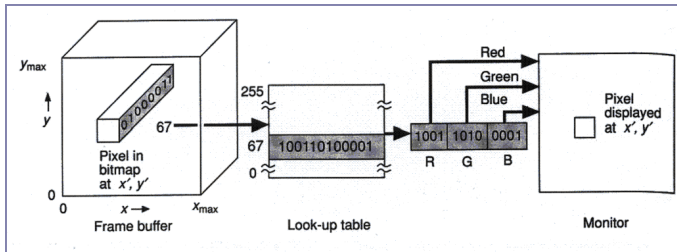
Frame Buffer

- Profundidade do frame buffer (depth):
 - Número de bits p/ cada pixel, determina o número de cores que o sistema consegue exibir
 - 1 bit = 2 cores
 - 8 bits = $2^8 = 256$ cores
 - 24 bits = $2^{24} = 16777216$ cores => sistema true color
 - d bits = 2^d cores possíveis (reais)
- Sistema RGB: grupos de bits associados a cada uma de 3 cores primárias: Red, Green, Blue

Frame Buffer

- A quantidade de **cores** representadas no frame buffer é **limitada**
 - **Paleta de cor** pode ser usada para permitir escolher qual faixa de cor utilizar
- O valor armazenado em uma posição do frame buffer pode ser usado como **índice de uma tabela de cor**
 - Clut: Color Lookup Table

Video Look-up Table

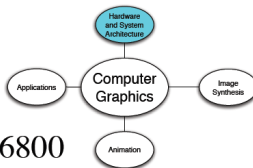


- Define a paleta de cores
 - Nesse exemplo uma paleta contém 256 cores das 4096 possíveis

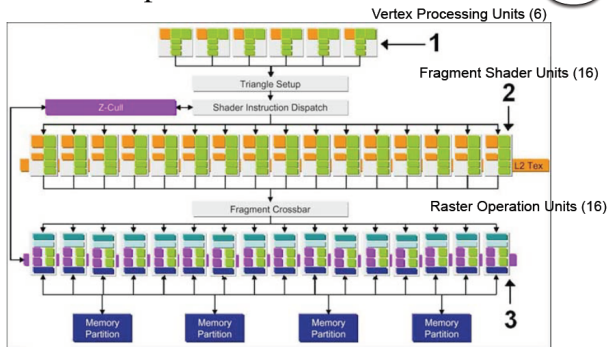
Processador

- O **processador** é responsável por transformar as primitivas gráficas (linhas, círculos, polígonos, etc.) em pixels no f.b.
 - Essa conversão é conhecida como **rasterização** ou **conversão matricial**

Hardware



- Example: NVIDIA GeForce 6800



© Torsten Möller

Sumário

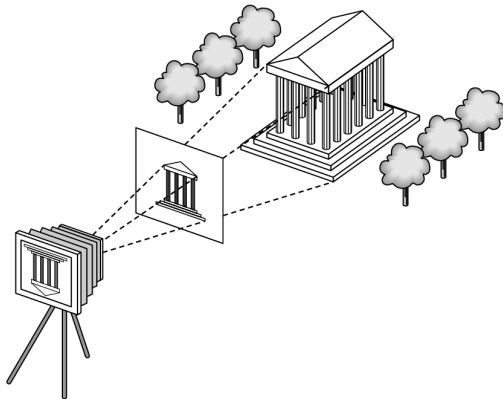
1 Dispositivos de Saída (Exibição)

2 Formação de Imagens

Formação de Imagens

- Em computação gráfica as imagens são formadas usando um **processo análogo** ao aplicado por sistemas físicos de imageamento
 - Câmeras
 - Microscópios
 - Telescópios

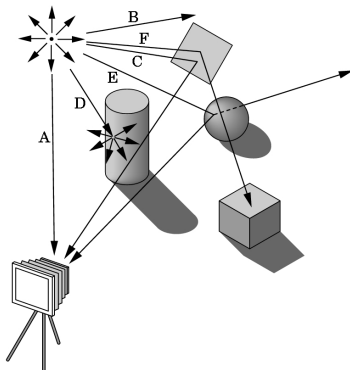
Modelo de Câmera Sintética



- (1) Objetos, (2) observador e (3) fonte(s) de luz

Ray Tracing

- Uma forma de implementação desse modelo de câmera é utilizando a abordagem de *ray tracing*



Pipeline Gráfico

- Outra implementação seria considerando uma arquitetura *Pipeline*, onde os objetos são **processados**, **um de cada vez**, na **ordem** que eles foram gerados pela aplicação
 - Só considera efeitos locais de luz
 - Abordagem empregada pela OpenGL



Pipeline Gráfico

- Muito do trabalho nesse pipeline está em **converter** representações de objetos de um **sistema de coordenada** para outro
 - Coordenadas dos objetos
 - Coordenadas de câmera (olho)
 - Coordenadas da tela



Processamento de Vértices

Processamento de Vértices

- **Projeção** é o processo que combina a visão 3D de um observador com objetos 3D para produzir imagens 2D
 - **Projeções perspectivas**: os raios de projeção se encontram no centro da projeção
 - **Projeção paralela**: os raios de projeção são paralelos, o centro da projeção é substituído por uma direção de projeção
- O processamento de vértices também calcula **cor**



Montagem de Primitivas

Montagem de Primitivas

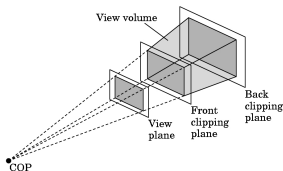
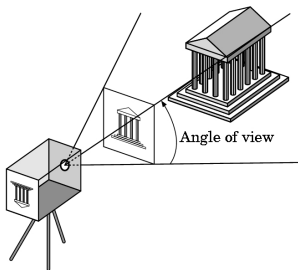
- Vértices devem compor objetos geométricos antes de acontecer o recorte e a rasterização
 - Segmentos de linha
 - Polígonos
 - Superfícies e curvas



Recorte

Recorte

- Assim como nenhuma câmera real pode “ver” completamente o mundo, a **câmera virtual** pode somente **ver parte mundo** ou espaço de objetos
 - Objetos que não estão dentro desse volume são “recortados” da cena



Rasterização

Rasterização

- Os objetos que não foram “recortados” **geram pixels** que são coloridos no frame buffer
 - Vértices e seus atributos são **interpolados** sobre os objetos pelo rasterizador
- A rasterização produz um conjunto de fragmentos (“pixels em potencial”) para cada objeto
 - Tem uma localização no frame buffer
 - Tem atributos de cor e profundidade



Processamento de Fragmentos

Processamento de Fragmentos

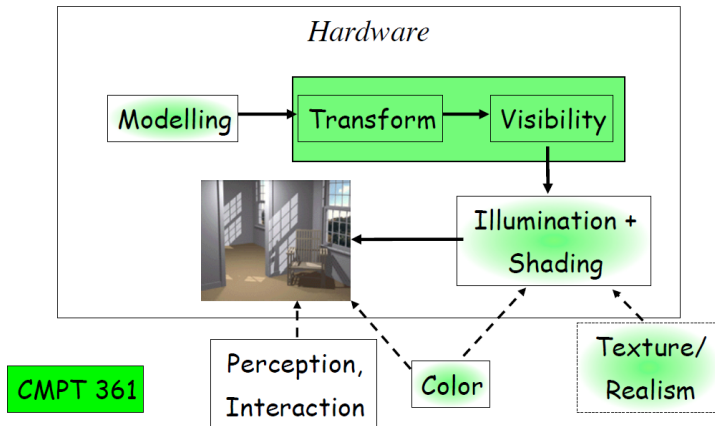
- Fragmentos são processados para **determinar a cor** dos pixels correspondentes no frame buffer
 - Mapeamento de textura ou interpolação das cores dos vértices
- **Fragmentos** podem ser **bloqueados** por outros fragmentos mais próximos da câmera
 - Remoção de superfície oculta



Pipeline Programável

- A maioria das placas gráficas implementam esse Pipeline Gráfico com funcionalidades fixas
 - Processamento dos vértices
 - Processamento dos fragmentos
- Atualmente o processamento dos vértices e fragmentos é programável
 - Muitas técnicas que não podiam ser implementadas em **tempo real agora são possíveis**
 - Isso é *GLSL (OpenGL Shading Language)*

Graphics Pipeline



Créditos

- Slides em inglês desta aula atribuídos a
 - Torsten Moller
 - <http://www.cs.sfu.ca/~torsten/Teaching/Cmpt361/>