

MAC0420/5744

Introdução à Computação Gráfica

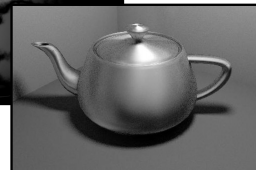
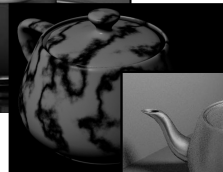
Hitoshi (hitoshi@ime.usp.br)
1o Sem. 2011

Departamento de Ciência da Computação - IME/USP

Renderização



70's: Luz, Textura



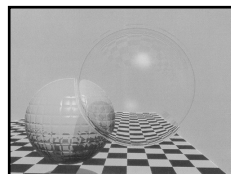
Histórico



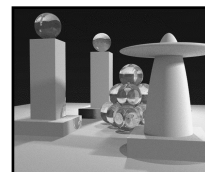
Ivan Sutherland
1963 - Sketch Pad

- Pop-up menus
- Constraint-based drawing
- Hierarchical model

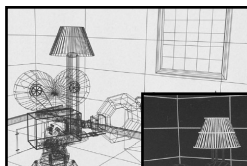
Renderização



80's: Modelos globais
de iluminação



Renderização



60's - Visibilidade

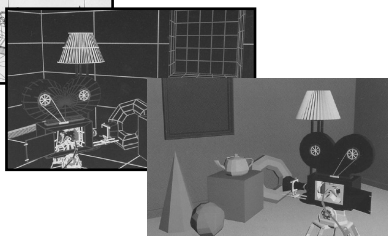
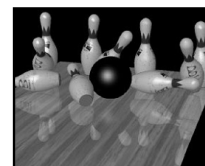
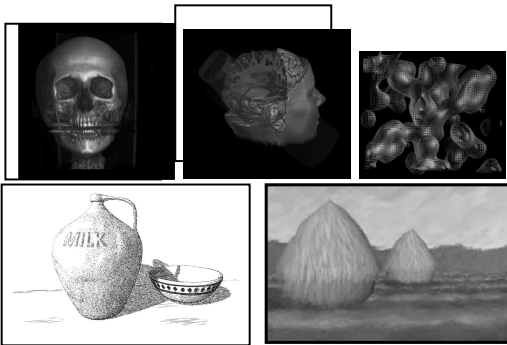


Foto realismo: final 80s



Não realismo: Início dos 90's



Dispositivos Gráficos

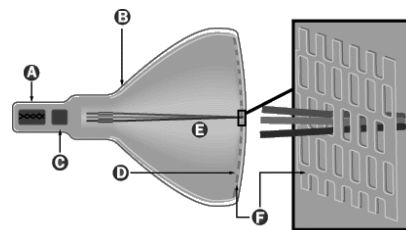
Tubo de Raios Catódicos (CRTs), impressoras, plotters

- Monitores Vetoriais (1963)
 - imagens são desenhadas ponto a ponto
 - apropriados para um número reduzido de linhas
 - linhas "brilhantes" em fundo escuro.
 - vantagem: requer pouca memória para armazenar os pontos (vértices)
 - muitos pontos resultam em flickering
- Storage Tube: Tektronix, fim dos anos 60.
 - permitia permanência da imagem, sem necessidade de HW complexo para *refresh*.
 - não permitia animação, e só permitia linhas

Computação Gráfica

- Surgiu para exibir informações gráficas
 - Inclui a criação, armazenamento e manipulação de modelos e imagens de objetos
- Utiliza os nossos canais visuais, que possui grande banda de transmissão de dados
- Interação: através do uso do teclado, mouse, touch-screen, etc. O estudo desses dispositivos acaba por vezes incluído na área de computação gráfica.

Tubo de raios catódicos



- | | |
|----------------------|-----------------------|
| A - Catodo | D - Fósforo |
| B - Camada condutiva | E - feixe de elétrons |
| C - Anodo | F - Máscara |

Interfaces Gráficas

- Interfaces baseadas em texto predominaram até o final da década de 70
- Interfaces WIMP: Windows, Icons, Menus, and Pointing.
- Aplicações:
 - Ciência (visualização), Engenharia (modelamento), Entretenimento, Educação, etc.
 - Exploração de realidades artificiais (realidade virtual)
 - Extrapolação da realidade (realidade estendida)
 - Tele-presença

Tubo de raios catódicos

- custo relativo baixo
- excelente cor, brilho e contraste
- excelente ângulo para visualização
- consumo médio de energia
- é pesado, difícil de transportar
- ocupa um grande volume (profundidade)

Outros Monitores

- 1990's -> Monitores de Cristal Líquido (LCD) - laptops
- 2000 -> projetores, cinema digital em casa
- Caverna, head mounted displays, autostereoscopic, etc.

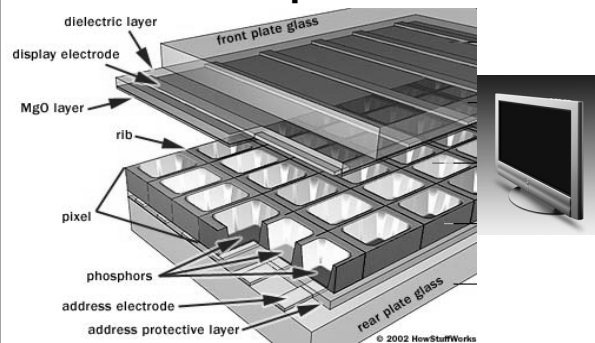
Monitor de cristal líquido

- Consome pouca energia
- Tem boa resolução
- Ocupa um volume pequeno (profundidade)
- É leve, e de fácil transporte
- brilho: inferior aos TRCs, mas pode ser melhorado usando matrizes ativas
- ângulo de visibilidade: pequeno
- baixo custo

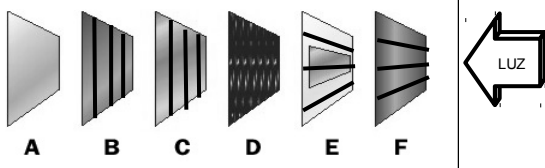
Monitores de Cristal Líquido

- Propriedades do cristal líquido:
 - mudam a polarização da luz em 90 graus
 - porém, quando submetida a um campo elétrico, a luz não muda sua orientação
- Funcionamento:
 - A luz ambiente entra no cristal polarizada. Sem nenhum potencial elétrico, sua orientação muda e passa pelo outro polarizador. Ao ser refletida pelo espelho, faz o caminho inverso.
 - Quando um potencial elétrico é aplicado a um par de fios horizontal x vertical, a luz que passa por aquela coordenada é bloqueada.
 - Através da varredura das linhas e colunas, é possível então exibir uma imagem.

Monitores de plasma



Monitor de cristal líquido



- A - espelho
B - polarizador vert.
C - grade vertical
D - cristal líquido
E - grade horizontal
F - polarizador hor.

Monitores de plasma

- Formado por várias "lâmpadinhas" de neon
- Cada lâmpada:
 - ao receber uma tensão elevada, acende.
 - para mantê-la acesa, pode-se manter a tensão menor que a inicialmente aplicada
 - abaixo de um certo nível de tensão, ela apaga
- Não necessita de refresh buffer
- densidade: de 50 a 150 pontos por polegada
- possui um bom ângulo de visualização, boas dimensões e consumo, mas ainda é caro!

Monitores Raster

- Matriz bidimensional de pixels para representar a imagem
- monocromático (binário e níveis de cinza)
- colorido RGB (Red Green Blue)
 - valores de RGB são combinados para formar cores
- matriz de pixels é armazenada em um *frame buffer* e o HW produz a imagem linha a linha (linhas raster e raster scan)
- taxa de refresh: > 30 vezes por segundo (flickering)
- exige muita memória, só disponível (\$\$) nos 70's.

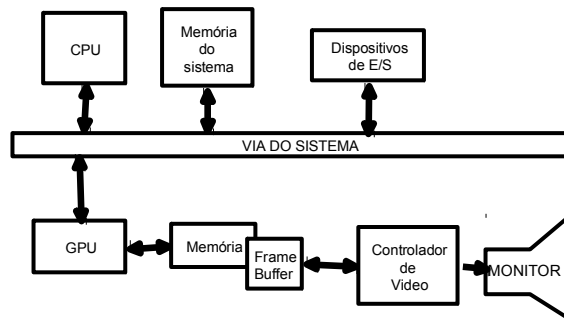
Processador Gráfico (GPU)

- Um processador especial, dedicado a realizar as rotinas de scan conversion e outras rotinas gráficas de alto nível, rapidamente, liberando a CPU.
- Os comandos possíveis variam com a sofisticação do processador gráfico.
 - transformações
 - projeções
 - clipping
 - rasterização
- Pipeline: arquitetura tipicamente utilizada
 - cada processador é responsável por uma operação

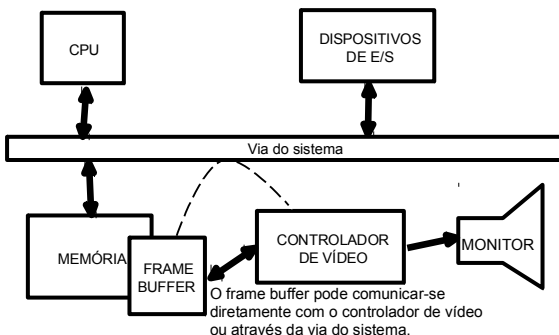
Arquitetura p/ Monitores Raster

- Como as imagens se comunicam entre o processador e o monitor?
- Frame Buffer: armazena a imagem
- A CPU modifica o frame buffer
- Controlador de vídeo: acessa o frame buffer e gera a imagem na tela.
- Rendering (alto nível)
 - processo para transformar um modelo em imagem
- Scan conversion ou Rasterization (baixo nível)
 - processo de conversão de linhas, curvas e interior de superfícies para a forma matricial (no frame buffer).
 - feito através de rotinas prontas (biblioteca gráfica).

Arquitetura raster com GPU



Arquitetura raster simples



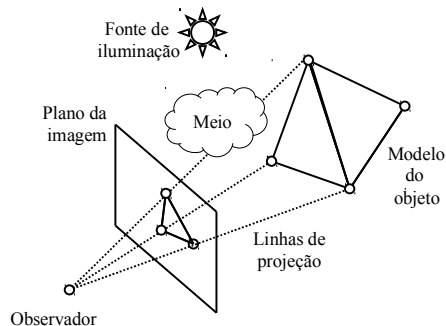
Cor e Imagem

- Percepção de cor e construção de Monitores (RGB)
- Cada componente deve ser representada por um número, ou tripla (R, G, B)
- 24 bits para produzir imagens realistas (high end)
 - Número de bits define a profundidade da imagem
- Exige muita memória para representar uma imagem. Como reduzir?

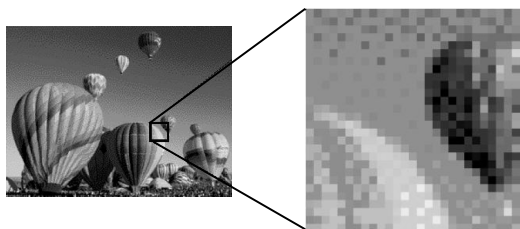
Imagem e Cor

- ✓ Bitmaps: imagens binárias
- ✓ Color maps: tipicamente, 8 bits são usados para indexar uma tabela (lookup table) que armazena uma cor de 24 bits.
- ✓ Componente Alfa: (R, G, B, A)
 - ✓ usada para gerar outros efeitos (opacidade)
- ✓ 256 cores são suficientes?
- ✓ Solução: Digital halftoning. Exemplo: jornais.

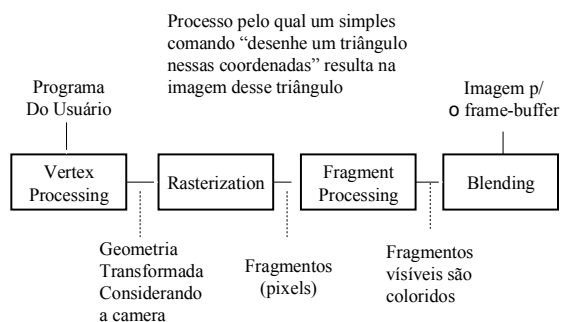
Situação típica de renderização



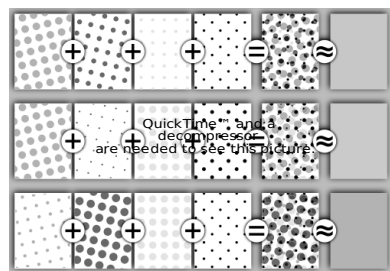
Digital Halftoning



Graphics Pipeline

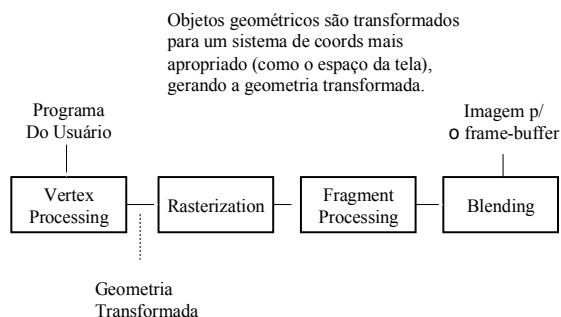


Digital Halftoning



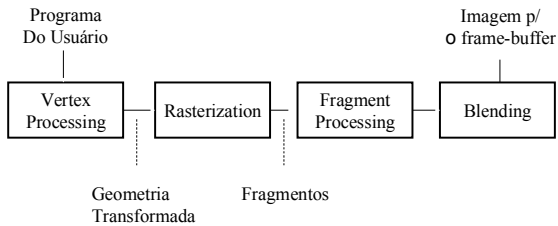
<http://en.wikipedia.org/wiki/File:Halftoningcolor.svg>

Vertex processing

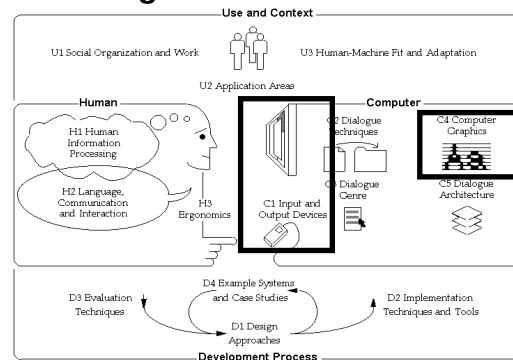


Rasterization

Formas geométricas são convertidas para pixels (ou fragmentos)

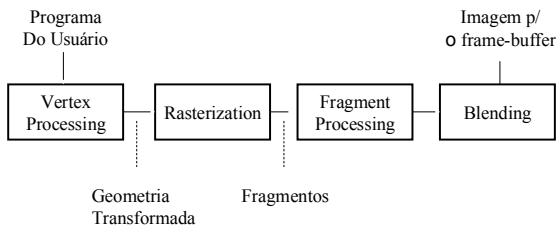


Sistema gráfico interativo e IHC



Fragment processing

Determina-se quais os fragmentos são visíveis. Estes recebem uma "cor".

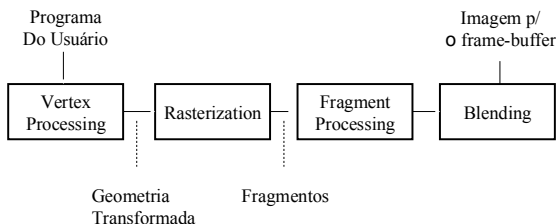


Interfaces WIMP

- Ambiente de trabalho da estação Xerox Star
- + tarde: Macintosh, Windows XX, etc.
- X windows: sistema baseado em janelas, desenvolvido em conjunto por universidades e corporações, com o propósito de ser independente da plataforma, e com acesso remoto (em rede).
- Cliente: programa aplicativo que gera os gráficos, usando os recursos do servidor.
- Servidor: software que controla o display (sistema de janelas, tipo X windows).
- Cliente e servidor podem estar na mesma máquina, ou se comunicam pela rede.

Blending

A cor de um pixel pode ser afetada por vários fragmentos (transparência por exemplo). Essas informações são combinadas para gerar a imagem final.



Graphics API (Applic. Prog. Interface)

- O X-Toolkit fornece ferramentas para a construção de interfaces, mas pouco auxilia com a geração de gráficos.
- Gráficos podem se tornar uma tarefa bastante complicada, principalmente quando envolvem objetos 3D.
- Uma API gráfica é uma biblioteca de funções que fornece rotinas de alto nível para o processamento de gráficos 3D.
- Exemplos: PHIGS, Java3D, e OpenGL.

Eventos e Rotinas Callback

- Programas gráficos interativos são dirigidos por eventos.
- Exemplo de eventos: mouse click, teclado, eventos do X (*display event*, *timer event*)
- A forma convencional de tratar eventos é utilizando rotinas callback.
 - O programa gráfico se "registra" para receber os eventos que deseja, pedindo ao sistema que, quando um certo evento ocorrer, chame uma determinada rotina.

Síntese de Imagens (3D graphics)

- Modelo para visualização
 - o observador possui uma câmera virtual, e a posiciona em relação ao modelo da cena, que é então renderizada.
- Componentes do modelo:
 - Objetos: descrição 3D da cena (estrutura, superfícies, e suas propriedades)
 - Observador: posição, orientação, etc.
 - Fontes de Luz: posição, forma, tipo, etc.
- Balanço: detalhes dos modelos x eficiência

Window Manager

- Ao teclar comandos em um sistema Unix, você interage com um programa chamado shell (bash, csh, ksh, tcsh, etc).
- Ao modificar uma janela (criação, posição, tamanho, etc) você interage com um programa chamado Window Manager (twm, ctwm, fvwm, etc).
- determina onde uma nova janela é colocada
- controla o formato e organização das janelas.

Modelos Geométricos

- Como descrever a cena 3D?
- Modelos baseados em estruturas matemáticas simples são mais populares devido a facilidade de programação.
- As formas primitivas devem ser flexíveis para permitir o modelamento de objetos complexos, através da combinação dessas primitivas simples.
- Modelos poliédricos: um sólido geométrico é descrito através de suas faces 2D.

Características de um WM

- não guarda o conteúdo das janelas escondidas.
- Quando uma janela (ou parte dela) é exposta, o X avisa o programa responsável que esse evento ocorreu, e o programa deve redesenhar a janela (expose event).
- Timer event: permite atualizar o display de forma periódica, o que é muito útil para criar animações.

Objetos poliédricos

- Um objeto pode ser representado pelos seus contornos.
- Contornos podem ser representados por elementos de diversas dimensões.
- vértices [0D]: ponto representado por (x,y,z)
- linhas (edges) [1D]: liga dois vértices
- faces [2D]: polígono plano formado por um conjunto fechado de linhas.

Topologia

- define a forma com que as formas primitivas (vértices, linhas e faces) são compostas para formar uma superfície.
- É importante para manipular as propriedades dos objetos
 - robótica e computação gráfica
- Faces: elementos básicos para renderização de objetos 3D.
 - pode ser definido por um número ilimitado de bordas, e pode conter buracos.

Outras propriedades

- Cor
- Textura
- Acabamento da superfície
- Esses componentes do modelo afetam como a luz é refletida, sendo responsáveis pela aparência de brilho, transparência, aspereza, etc.
- O OpenGL oferece alguns recursos limitados para o modelamento de componentes de acabamento de superfícies.

Polígonos convexos

- Para acelerar o processo de renderização, a maioria dos sistemas gráficos assumem que as faces são compostas por polígonos convexos simples.
- Definição: uma forma é dita CONVEXA se a intersecção de qualquer linha com a forma é um único segmento de linha.
- Polígonos convexos possuem ângulos internos menores que 180 graus, e não contem buracos.

Luz e Fontes de Luz

- A posição das fontes de luz determinam o padrão de sombreado e as sombras no processo de renderização da cena.
- Forma:
 - ponto: Sol
 - região: Luz fluorescente
- Cor:
 - lâmpadas incandescentes: luz avermelhada
 - lâmpadas fluorescente: luz azulada
 - luz solar: a cor depende da hora, dia, mes...

Tesselação

- Como tratar os polígonos não convexos?
- Muitas APIs gráficas fornecem rotinas para dividir polígonos complexos em uma coleção de polígonos convexos, particularmente em triângulos.
- Esse processo é chamado de tesselação ou triangulação.
- O número de faces se torna elevado para um modelo, mas simplifica a renderização.

Modelo de Camera

- A cena 3D deve ser "projetada" no plano 2D de imagem da câmera
- Projeções: paralela e perspectiva
- Projeção paralela: os objetos são projetados ao longo de linhas paralelas
- Projeção perspectiva: os objetos são projetados ao longo de linhas que se encontram em um ponto comum, que define o centro de projeção, ou ponto focal.

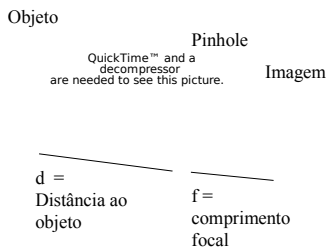
Camera Pin-hole

- Caixa que possui um pequeno buraco em uma face (pin-hole), e a imagem é formada na face oposta.
- A posição do pin-hole define o centro de projeção, e a posição da face oposta define o plano de imagem.
- Geometria de uma câmera pin-hole:
 - eixo z: ao longo do eixo óptico, com o centro de projeção na origem
 - plano de imagem: dimensão {w,h} com $z = -d$

Posição da Câmera (na cena)

- posição: do centro de projeção (CP)
- direção: (vetor) para onde a câmera aponta
- orientação: direção do vetor "up"
- comprimento focal: distância entre CP e o plano de imagem.
- tamanho da imagem: tamanho da região retangular sobre o plano de imagem que corresponde a imagem final.
- FOV e Aspect Ratio (w/h).

Modelo de câmera pinhole



O que você deve saber...

- Dispositivos de HW
 - Tubos de raios catódicos, cristal líquido e plasma
 - Monitores Raster, vetoriais, storage tube
 - Arquiteturas de monitores raster
- Cor e sua representação
 - Lookup Table (paleta de cores)
- Programação Gráfica: XWindows, eventos, callbacks, etc
- Modelos geométricos: Vértices, linhas, faces, tesselação, objetos concavos e convexos, luz e fontes de luz, etc
- Projeção: paralela x perspectiva
- Modelos de câmera: pinhole

... ainda sobre pin-hole

- A transformação não é definida para todo espaço 3D ($z = 0$).
- Problemas quando $z < 0$ e fora do cone de visão ou campo visual (clipping).
- O processo de projeção inverte a imagem.
 - matematicamente, podemos colocar o "filme" da camera na frente da lente, tornando $d > 0$.
- Obs: é um modelo simples...