



COMPUTAÇÃO GRÁFICA

Dispositivos Gráficos

CURSO DE CIÊNCIA DA COMPUTAÇÃO
UNIPAC BARBACENA
PROFESSOR NAIRON NERI SILVA

Resolução de imagens

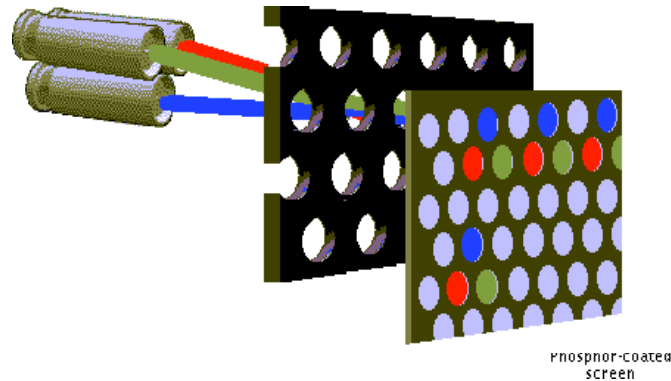
- A resolução está associada a quantidade e a qualidade de informação que um dispositivo apresenta.
 - A resolução pode ser medida em pixels (*pictures elements*), no caso de monitores ou em DPI (impressoras).
 - O pixel representa uma unidade que pode ser controlada individualmente e que contém informações sobre cores e brilho.
 - O tamanho do pixel vai depender de como a resolução da tela foi configurada.

Resolução de imagens

- Relação pixels X bits

- Cada pixel requer uma quantidade de bits, de acordo com o seu modelo de cores.

- Ex: **Sistema RGB**



Cada cor primária - 256 níveis (8 bits)

Cada pixel – 3 cores (RGB) X 8 bits = 24 bits (true color) => 16 M de cores

Cores de 32 bits => 8 bits para o canal alfa, com 256 níveis de transparência

Cores Preto & Branco – 1 bit (Ex: 0 Branco, 1 Preto)

Para uma resolução de 800 X 600, 32 bits -> 1,87 MB

1.024 X 768, 24 bits -> 2,30 MB

Resolução de imagens

dpi (pontos por polegada)

- Uma medida da resolução da impressora em pontos por polegada. Impressoras a laser/ink jet típicas de mesa imprimem em 300 dpi.
- Fotocompositoras imprimem em 1270 ou 2540 dpi. Impressoras com capacidades maiores de dpi produzem uma saída mais suave e mais limpa.
- O termo dpi também é usado para medir resolução de digitalização e para indicar resolução de bitmap.
- Imagens para visualização em monitores não precisam ter mais que 100 dpi (porém existem telas com dpi bem mais alto).
- Imagens a serem impressas, o mínimo recomendado para uma boa resolução são 300 dpi.

Resolução de imagens

- O número de pixels por polegada de uma imagem em formato bitmap, é medida em ppi (pixels por polegada) ou dpi (pontos por polegada).
- Resoluções baixas podem resultar em aparência granulada na imagem em formato bitmap, e resoluções altas podem produzir imagens mais suaves, mas resultam em arquivos maiores.



Resolução Gráfica

- Virtualmente todos os dispositivos de I/O gráficos usam uma malha retangular de posições endereçáveis - a qual é denominada "retângulo de visualização".
- A "resolução gráfica" de um dispositivo é o número de posições (ou pontos, ou pixels) horizontais e verticais que ele pode distinguir. Existem 4 parâmetros que definem a resolução.

- 1. ndh** - número de posições endereçáveis horizontalmente.
- 2. ndv** - número de posições endereçáveis verticalmente.
- 3. width** - a largura do retângulo de visualização em mm.
- 4. height** - a altura do retângulo de visualização em mm.

- A partir desses 4 parâmetros, vários números interessantes podem ser calculados.

Resolução Gráfica

1. resolução horizontal:

$$\text{horiz_res} = \frac{ndh}{width}$$

2. tamanho ponto horizontal:

$$\text{horiz_dot_size} = \frac{width}{ndh}$$

3. resolução vertical:

$$\text{vert_res} = \frac{ndv}{height}$$

4. tamanho ponto vertical:

$$\text{vert_dot_size} = \frac{height}{ndv}$$

5. total pontos endereçáveis:

$$\text{total_nr_dots} = ndh.ndv$$

6. resolução de área:

$$\text{area_res} = \frac{\text{total_nr_dots}}{(width.height)}$$

7. razão de aspecto (gráfica):

$$\text{aspect_ratio} = \frac{\text{vert_dot_size}}{\text{horiz_dot_size}}$$

8. razão de aspecto (física):

$$\text{physical_aspect_ratio} = \frac{height}{width}$$

- Note que *horiz_res* , *vert_res* e *area_res* definem resoluções físicas, enquanto que *ndh*, *ndv* e *total_nr_dots* definem resoluções gráficas. Dispositivos de visualização podem ter a mesma resolução gráfica, com resoluções físicas muito diferentes.

- O ideal seria ter um *aspect ratio* igual ou próximo de 1.

Arquitetura Gráfica

Frame-buffer

- O acesso ao dispositivo gráfico é mais lento que o acesso à memória.
 - Isso acontece devido à grande quantidade de componentes que atuam para que um *pixel* seja mostrado na tela.
 - O acesso ao controlador gráfico é feito via protocolos de hardware e software que são mais complicados que enviar um byte para memória.
- Assim, existem ocasiões que a velocidade é um fator importante:
 - Rasterização on-line, re-display de imagens, jogos, animações etc.
- É necessário o desenvolvimento de técnicas de construção de imagens em memória.
 - Uma das principais é a criação do "*frame -buffer*".

Arquitetura Gráfica

Frame-buffer

- O frame-buffer é composto de uma região da memória que armazenará a imagem e por um grupo de rotinas de acesso à imagem.
- A memória é considerada como um “espaço” monodimensional, onde cada byte é endereçado por um único valor.
- Já uma imagem é um objeto gerado em espaço bidimensional (ou 3D), e isso acarretará:
 - Um pixel da imagem será representado por um número de bytes, dependendo de quantos bits serão necessários para compor a cor correspondente.
 - A quantidade de bytes na memória deverá ser suficiente para comportar todos os pixels a serem representados.
 - Deverá ter uma correspondência aritmética entre a posição (x, y) do pixel e seu endereço no frame-buffer.

Arquitetura Gráfica

Frame-buffer

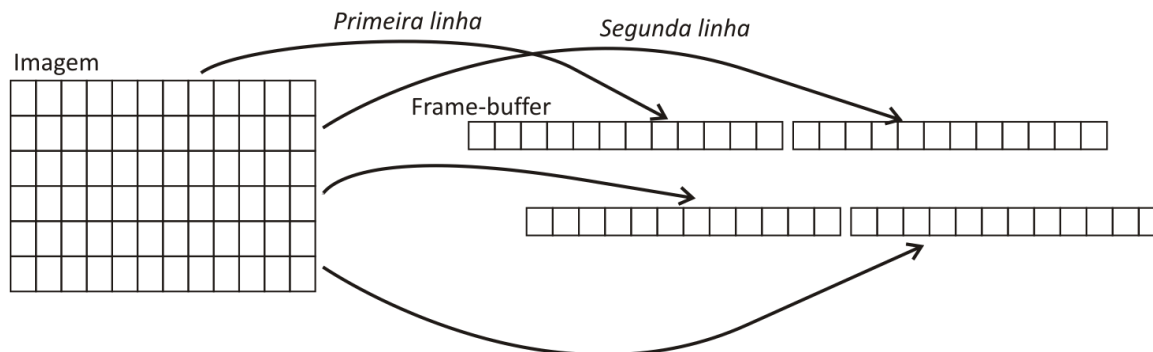
- Essa condição - de correspondência entre posições do pixel e do endereço – pode ser contornada se for utilizada uma linguagem de alto nível que aceite a definição de variáveis do tipo *arrays* ou matrizes bidimensionais.
- Por exemplo, seja uma imagem *true color* de 800 x 600 pixels de tamanho.
 - Cada pixel da imagem é representado por 32 bits (4 bytes).
- O tamanho total do frame-buffer para essa imagem será então 800 x 600 x 4 bytes = 1.920.000 bytes (1,9 MB).
- Em C, a declaração dessa área seria:

```
unsigned long int FB [800][600];
```

Arquitetura Gráfica

Frame-buffer

- Ao longo do programa, o acesso ao pixel (x, y) pode ser feito por:
$$FB[X][Y]$$
- Caso seja usada uma linguagem que não suporte variáveis do tipo matrizes, deve-se implementar uma arquitetura de memória e uma correspondente conversão de endereços.
- Uma boa (e simples) representação do frame-buffer é considerar cada linha como uma sequência de bytes que se agrupam em linhas como sequencias consecutivas.



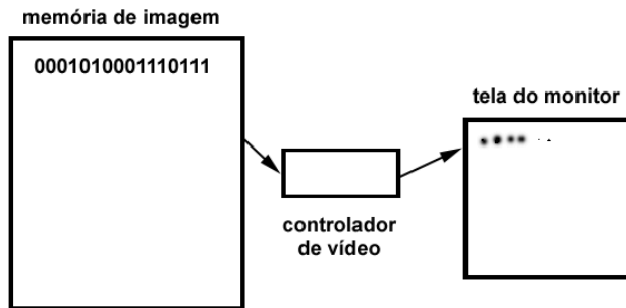
Arquitetura Gráfica

Frame-buffer

- Seguindo a imagem exemplo (800x600, true color) e estabelecendo que o primeiro byte do frame-buffer é designado pelo endereço FB , o segundo pixel está no endereço $FB+4$ (cada pixel ocupa 4 bytes).
 - O 3º pixel está em $FB+2*4$ e o último byte da primeira linha está em $FB+799*4$.
 - O 1º pixel da segunda linha está em $FB+800*4$. O 2º em $FB+800*4+4$ e o 3º em $FB+800*4+2*4$.
- De forma genérica, um pixel qualquer na posição (x,y) na imagem exemplo é dada por : $FB+[800*y+x]*4$.
- Para um caso qualquer, o endereço de um pixel no frame-buffer é dado por $m = FB+[x+Ly]*b$, onde m é o endereço do início do pixel no frame-buffer, FB é o endereço inicial do frame-buffer, L é o nº de pixels numa linha da imagem e b o tamanho que um pixel ocupa em bytes.

Arquitetura Gráfica

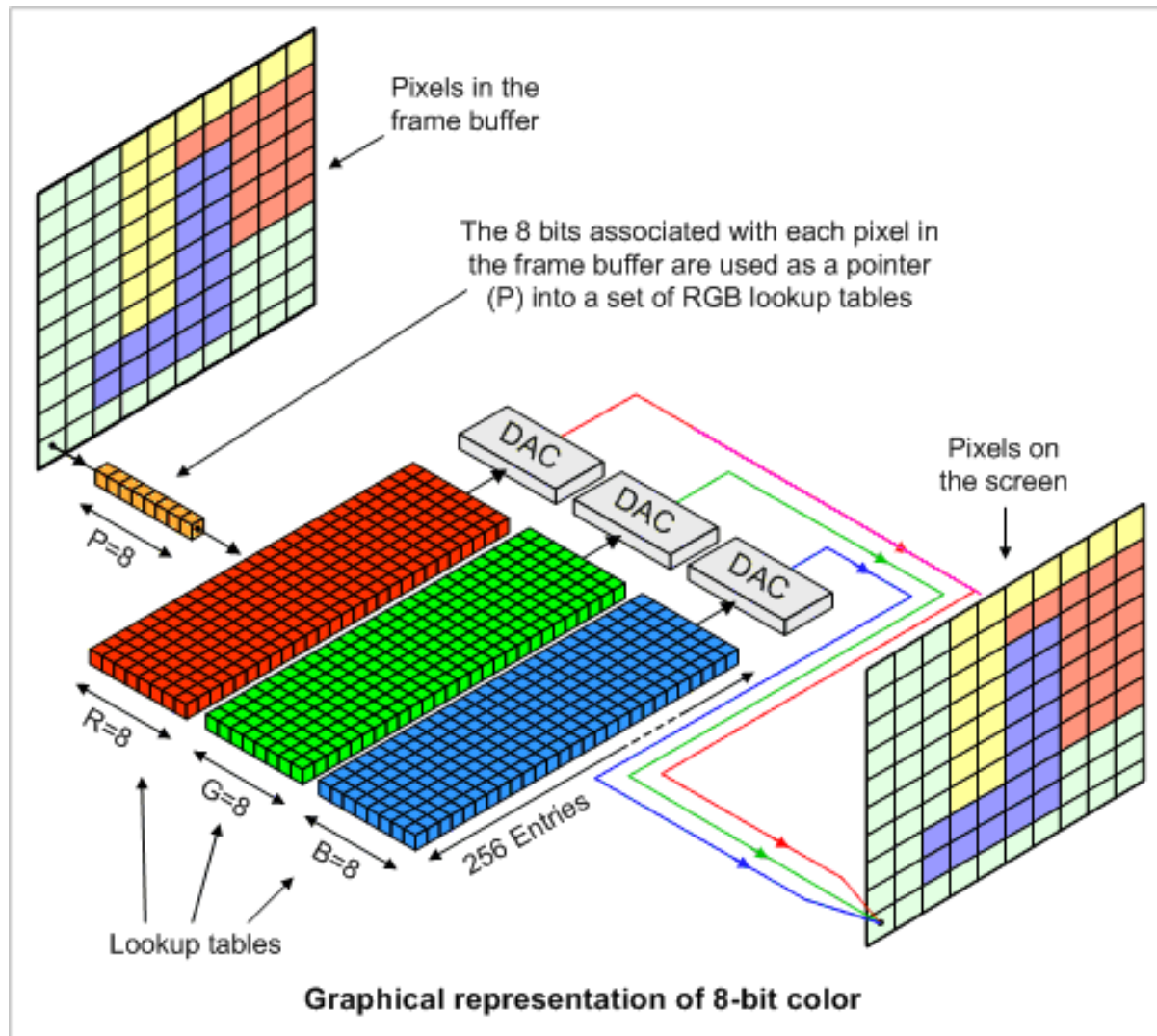
- A tecnologia utilizada atualmente na grande maioria dos terminais de vídeo gráficos é a mesma dos aparelhos de TV. Um terminal gráfico simples requer:



Uma sequência de bits na memória de imagem é convertida para uma sequência de pixels na tela.

1. Uma memória digital (frame buffer), na qual a imagem a ser visualizada é armazenada como uma matriz de pixels (cada posição na matriz contém a intensidade associada ao pixel correspondente na tela).
2. O monitor.
3. Um controlador de vídeo (display controller), que consiste de uma interface que transfere o conteúdo do frame buffer para o monitor.
 - Os dados devem ser transferidos repetidamente, pelo menos 15 vezes por segundo, de modo a manter uma imagem estável na tela, reduzindo o *flickering*. Note que processo de transferência implica numa conversão digital-analógica (DAC).

Arquitetura Gráfica



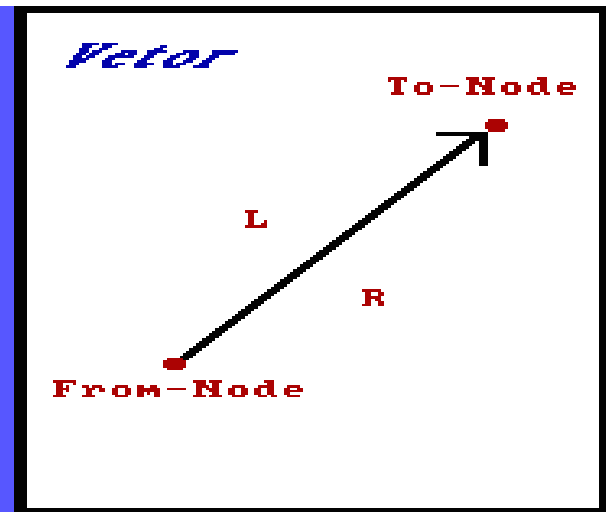
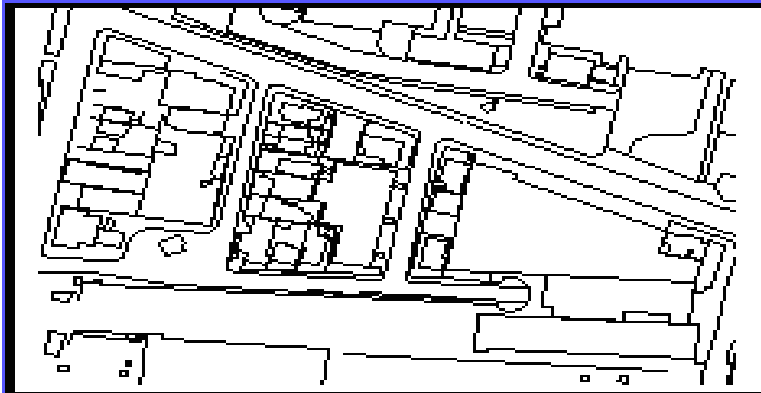
Tipos de Imagens

- As imagens podem ser classificadas como Vetorial e Matricial;
- A representação vetorial é empregada para a definição e modelagens de objetos que serão representados pela imagem;
- Na representação vetorial são usados como elementos básicos os pontos, as linhas, as curvas, etc.;
- Esses elementos básicos são chamados primitivas vetoriais;
- Cada primitiva vetorial possui um conjunto de atributos que define sua aparência e um conjunto de dados que define sua geometria.



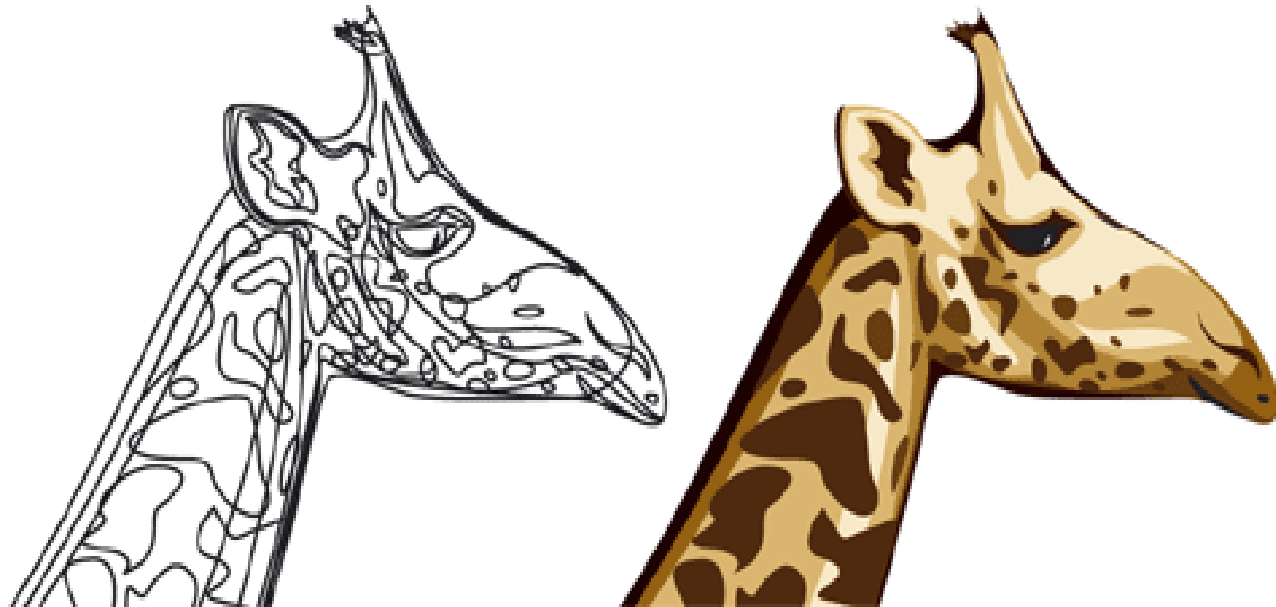
Imagens Vetoriais

- Vantagens das imagens vetoriais:
 - Facilidade de armazenamento dos elementos geométricos;
 - Facilidade de manipulação (escala, rotação, etc.);
 - Alteração simples;
- Desvantagem das imagens vetoriais
 - Requer dispositivos de saída específicos para ter bons resultados;
 - Reconstrução mais lenta.



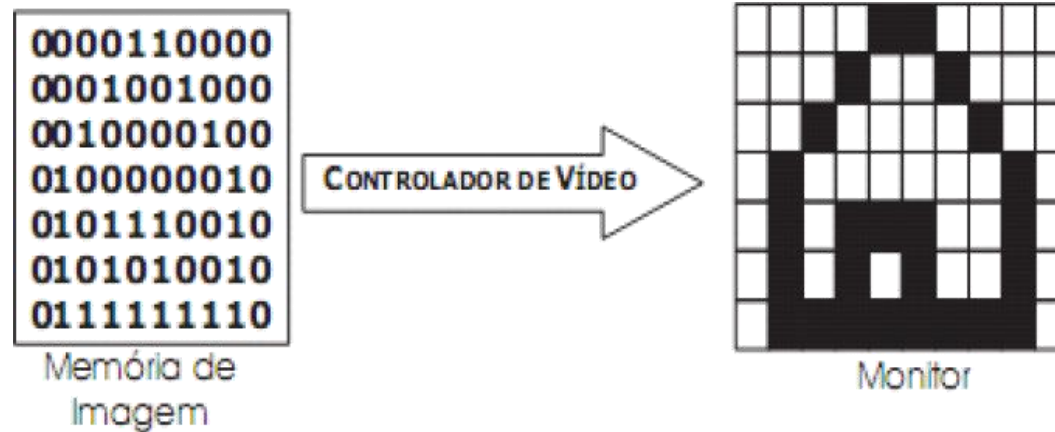
Imagens Vetoriais

- Ilustração vetorial com e sem preenchimento de cor.



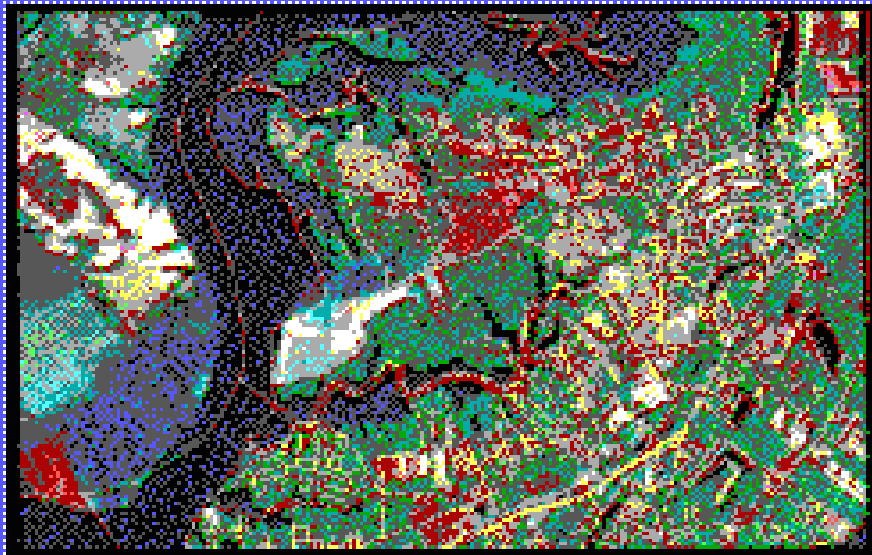
Imagens Matriciais

- Na representação matricial, a imagem é descrita por um conjunto de células em um arranjo espacial bidimensional, uma matriz;
- Cada célula representa os pixels da imagem;
- Os objetos são formados usando adequadamente esses pixels;
- As imagens matriciais são também conhecidas como bitmaps;
- A representação matricial é usada para formar a imagem na memória e nas telas de computador.

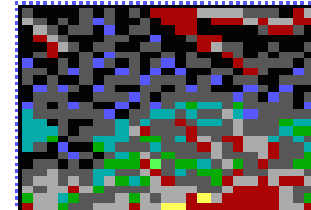


Imagens Matriciais

- Bitmaps ou Mapa de Bits



PIXEL

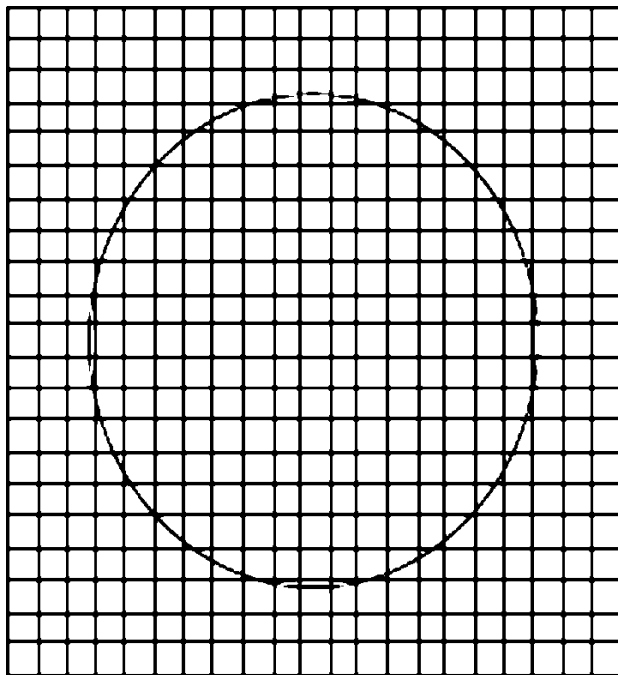


(C,L,9999999)

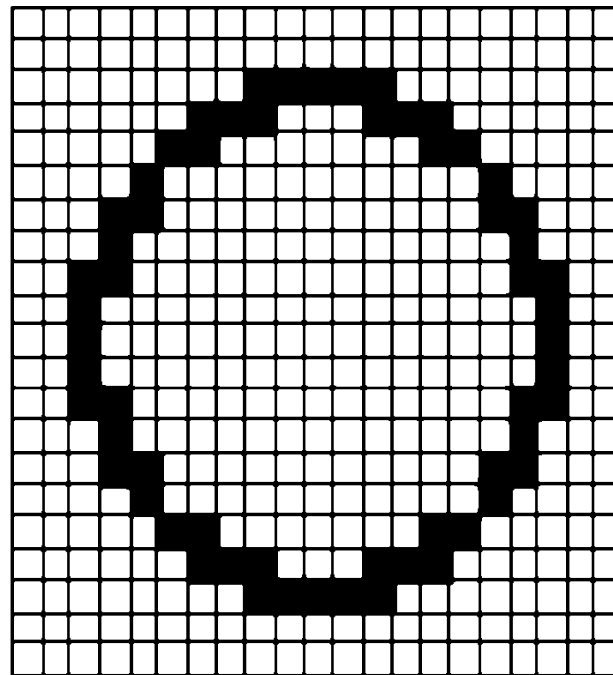
Imagens Matriciais

- Vantagens das imagens matriciais
 - Fácil tradução para dispositivos baseados em pontos (monitores, impressoras, etc.);
 - Fácil armazenamento e leitura;
 - Valores dos pixels podem ser alterados individualmente ou em grupo;
- Desvantagens das imagens matriciais
 - Imagens podem ser muito grandes;
 - Dificuldade em realizar operações de escala;

Representação de Imagens



Objeto Vetorial



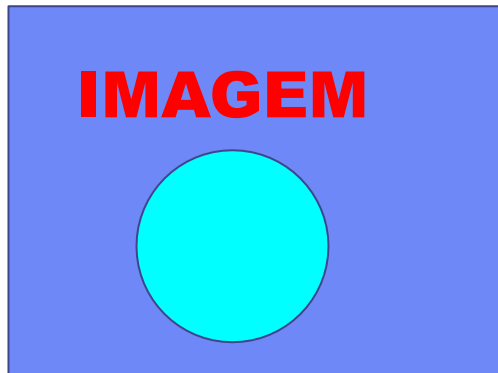
Objeto Matricial

Conversão de formatos

- Bitmap p/ Bitmap
 - Melhores resultados
 - Reajuste na informação de cor
 - Problemas com diferenças no tamanho da paleta de cor
- Vetorial p/ Vetorial
 - Problemas com diferenças entre o número e o tipo de objetos disponíveis
 - Problemas com interpretação de medidas e com a aparência dos elementos de imagem e das primitivas

Conversão de formatos

- Vetorial p/ Bitmap
 - Imagem vetorial é decomposta em pixels e colocada numa matriz
 - Qualidade depende do tamanho da matriz
 - Problemas de serrilhado



Conversão de formatos

- Bitmap para vetorial
 - Conversão mais difícil, com altos índices de falha
 - Algoritmos e heurísticas de detecção de formas
 - Resultados bons para formas geométricas, ruins para imagens reais
 - Normalmente resulta na perda de cores



Conversão de formatos

- Outros fatores que influenciam a conversão
 - Formatos proprietários ou específicos de uma aplicação
 - Tamanho da paleta de cores
 - Formato da compressão

Dispositivos Gráficos

- Toda imagem criada através de recursos computacionais deve ser representada em algum dispositivo físico que permita a sua visualização.
- Diversas tecnologias e diferentes tipos de dispositivos são utilizados para gerar representações visuais, sendo que o desenvolvimento dessas tecnologias teve um papel fundamental na evolução da CG.
- Tanto para o usuário como para o implementador de sistemas gráficos é importante conhecer as características de cada uma dessas tecnologias para sua melhor utilização.
- Vamos discutir alguns aspectos da arquitetura e organização dos tipos mais comuns dos dispositivos de exibição gráfica, sem entrar em detalhes técnicos.

Dispositivos Gráficos

- É possível classificar os dispositivos de exibição (traçadores, impressoras e terminais de vídeo) em duas principais categorias, segundo a forma pela qual as imagens são geradas:
dispositivos vetoriais e dispositivos matriciais.
 - Os dispositivos gráficos vetoriais conseguem traçar segmentos de reta perfeitos entre dois pontos da malha finita de pontos definida por suas superfícies de exibição.
 - Os dispositivos matriciais, por outro lado, apenas conseguem traçar pontos, também em uma malha finita. Assim, segmentos de reta são traçados como sequências de pontos próximos.
- Através dos dispositivos gráficos interagimos com o sistema na busca de uma extensão dos limites do nosso corpo e uma melhor comunicação com a máquina.
- Os dispositivos podem ser classificados em:
 - **Dispositivos gráficos de entrada**
 - **Dispositivos gráficos de saída**

Dispositivos gráficos de entrada

- Joysticks
 - Alavancas de comando que determinam a direção e velocidade do cursor na tela;
 - Usados geralmente nos jogos de videogame, estações de realidade virtual e estações industriais de controle de robôs.



Dispositivos gráficos de entrada

- Telas sensíveis ao toque
 - Provavelmente, o dispositivo gráfico de entrada mais utilizado ultimamente;
 - Utilizado em Smartphones, Tablets, Notebook...



Dispositivos gráficos de entrada

- Mesa Digitalizadora
 - Consiste em uma mesa e um apontador;
 - Ao tocar a mesa o computador recebe as coordenadas do referido ponto;



Dispositivos gráficos de entrada

- Digitalizador 3D
 - Uma espécie de braço mecânico com um sensor de toque na ponta.



Dispositivos gráficos de entrada

- Luvas
 - Através de sensores detectam e medem as flexões e pressões dos dedos;



Dispositivos gráficos de entrada

- Scanners Tridimensionais
 - Utilizam câmeras digitais acopladas a uma mesa especial que fornece as coordenadas para os sistemas;



Dispositivos gráficos de entrada

- Capacetes
 - Estereoscópicos ou monoscópicos;
 - Binoculares ou monoculares;



Dispositivos gráficos de entrada

- Roupa de RV
 - Permite a interação com o mundo virtual;
 - Usadas para gerar informações de movimento humano.



Dispositivos gráficos de saída

- Plotters
 - Produzem desenhos em grande dimensões, com elevada qualidade e rigor.



Dispositivos gráficos saída

- Cave
 - Usam a ideia de colocar o usuário em uma sala com paredes que são na verdade telas para projeção de imagens.



Dispositivos gráficos de saída

- Óculos com display
 - Óculos inteligente composto por um visor que entrega diversas informações ao usuário



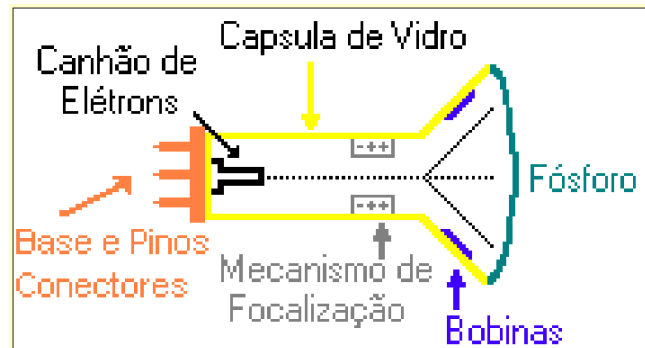
Dispositivos gráficos de saída

- Óculos de VR
 - Conhecidos como “óculos de realidade virtual”;
 - Operam exibindo duas telas de imagens de uma cena virtual;



Dispositivos gráficos de saída - Monitores

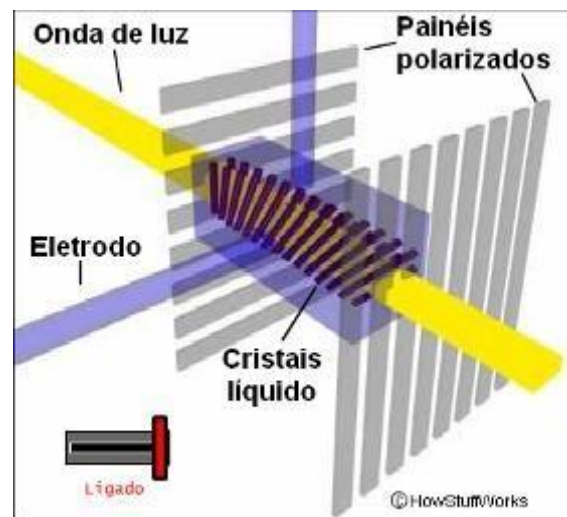
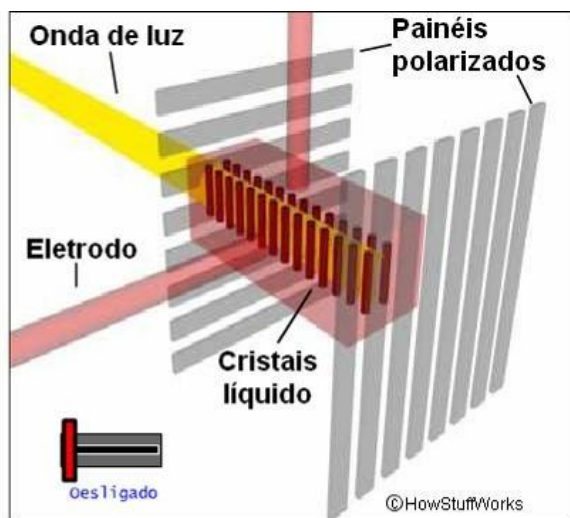
- **CRT (Tubo de Raios Catódicos)** - Consiste basicamente de uma superfície de exibição, quase plana, recoberta internamente de material à base de fósforo, um canhão emissor de elétrons e um sistema de deflexão.



- O canhão emite um fino feixe de elétrons que, acelerados, chocam-se contra a superfície fosforescente da tela.
- Sob a ação dos elétrons, o material fosforescente incandesce, emitindo luz no ponto da tela atingido pelo feixe. A função do sistema de deflexão é dirigir o feixe de elétrons para um determinado ponto da tela.

Dispositivos gráficos de saída - Monitores

- **LCD** (Displays de Cristal Líquido) – A tecnologia de monitores de cristal líquido funciona através do bloqueio da luz. Um monitor LCD é formado por duas peças de vidro polarizado (também chamado de substrato) preenchidas com material de cristal líquido.
- Uma luz de fundo passa pelo primeiro substrato. Ao mesmo tempo, correntes elétricas fazem com que as moléculas de cristal líquido se alinhem para formar as variações de luz que passam para o segundo substrato e assim, formem as cores e imagens que você vê.



Dispositivos gráficos de saída - Monitores



Fonte dos Slides

- *Slide do material do Prof. Jorge Cavalcanti - UNIVASF*