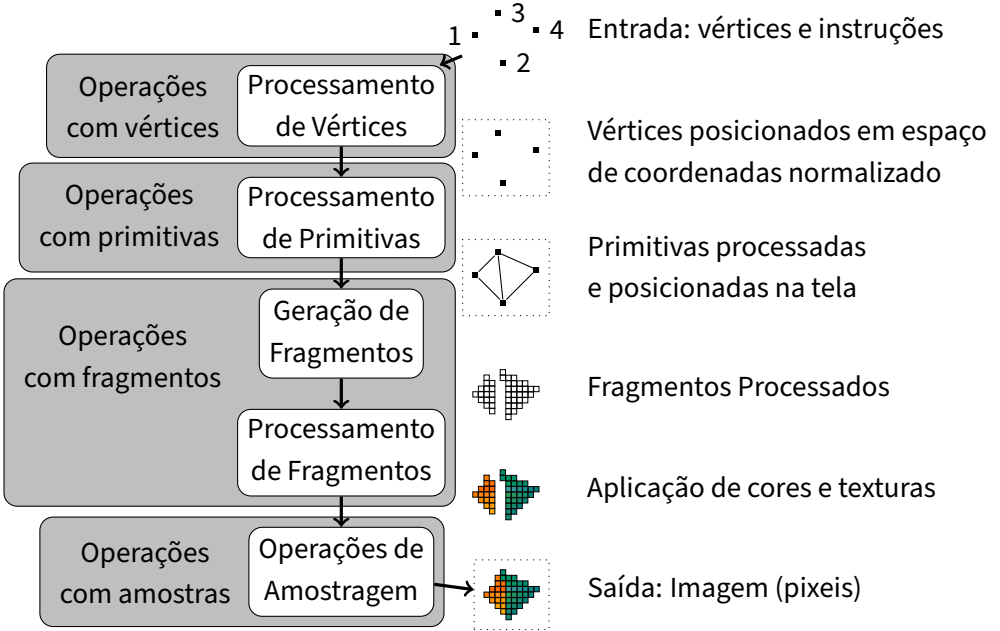
Computação Gráfica

Rasterização

Antialiasing

Prof. Alaor Cervati Neto

2023/1


1
2
3
4
Entrada: vértices e instruções
Operações com vértices
Processamento de Vértices
Vértices posicionados em espaço de coordenadas normalizado
Operações com primitivas
Processamento de Primitivas
Primitivas processadas e posicionadas na tela
Operações com fragmentos
Geração de Fragmentos
Fragmentos Processados
Processamento de Fragmentos
Aplicação de cores e texturas
Operações com amostras
Operações de Amostragem
Saída: Imagem (pixeis)

Sistema Gráfico

Sistema Gráfico

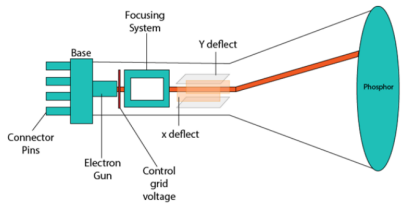
Partes do Sistema Gráfico:

- Dispositivos de Entrada.
- Processador.
- Memória.
- *Frame Buffer*.
- Dispositivos de Saída.

Sistema Gráfico

Monitor de Vídeo (dispositivo de saída):

- CRT (*Cathode Ray Tube*) foi usado por muito tempo.
- Feixe de elétrons em material fosforescente.



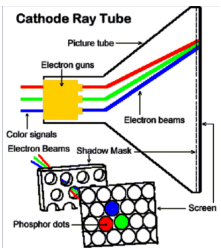
Cathode Ray Tube

Sistema Gráfico

Monitor de Vídeo (dispositivo de saída):

- CRT coloridos.
- Intensidade dos feixes corresponde à cor dos pixeis.
- Primeiros tinham 8 cores.

Valores			Valor Binário	COR
R	G	B		
0	0	0	0	BLACK
0	0	1	1	BLUE
0	1	0	2	GREEN
0	1	1	3	CYAN
1	0	0	4	RED
1	0	1	5	MAGENTA
1	1	0	6	YELLOW
1	1	1	7	WHITE



Sistema Gráfico

Monitor de Vídeo (dispositivo de saída):

- LCD (*Liquid Cristal Display*): tela é composta por cristais que são polarizados (via corrente elétrica) para gerar as cores.
- Plasma: tela composta por minúsculas cápsulas de vidro e gás (plasma) e uma capa de fósforo.
  - As cápsulas são os pixeis e cada uma é composta de 3 subpixeis que correspondem às cores RGB.
  - Corrente elétrica faz com que o plasma emita raios ultravioleta.
  - Essa luz faz o fósforo brilhar na cor apropriada.

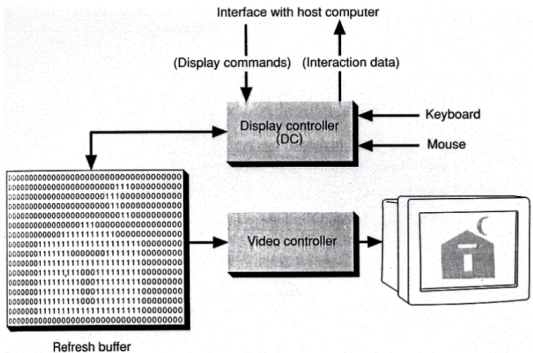
Sistema Gráfico

Monitor de Vídeo (dispositivo de saída):

- LED (*Light-Emitting Diode*): Uso de diodos (*Red*, *Green*, *Blue*) para emissão da luz. Pontos são iluminados de forma separada, melhorando a definição, cores, e contraste.
- Outros dispositivos: projetores, hologramas, etc.

Sistema Gráfico

Arquitetura simplificada de um dispositivo de exibição (matricial):



Sistema Gráfico

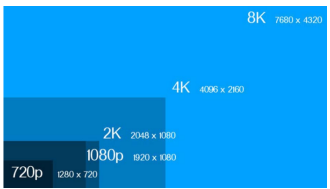
Placas Gráficas:

- Hardware responsável por receber os comandos do processador e controlar o monitor de vídeo.
- *Drawing front end* (*drawing engine*): recebe os comandos do processador com quais pixeis a serem traçados e o sua intensidade (cores) e atualiza o "*bitmap*" do *frame-buffer*.
- *Video back-end*: interpreta o *bitmap* do *frame-buffer* e traduz para o monitor de vídeo.

Sistema Gráfico

Imagem no dispositivo matricial:

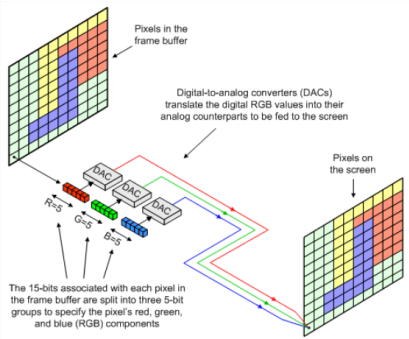
- Cada imagem é mantida no *frame-buffer* (*fb*), que contém uma posição associada a cada pixel da tela.
- Cada pixel tem um valor de intensidade (ou cor).
- Resolução é o número de pixeis, que é igual à memória do *fb*.



Conversão Matricial

Conversão Matricial

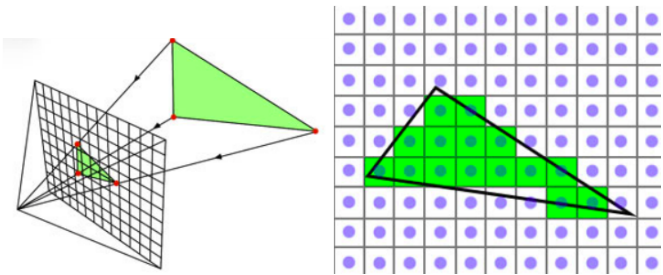
Frame-buffer contém representação matricial discreta da imagem (cena).



Graphical representation of 15-bit color

Conversão Matricial

Objetos geométricos (vetoriais) precisam ser convertidos para a representação matricial (pixeis).



Resulta em serrilhamento (*aliasing*)

Conversão Matricial

Problema:

- Traçar primitivas geométricas (segmentos de reta, polígonos, circunferências, elipses, curvas, etc.) no dispositivo matricial.
- Rasterização: vetorial para matricial.
- Na prática, deseja-se converter do sistema de coordenadas reais (pontos flutuantes) para o sistema de coordenadas inteiras (discreto).

Conversão Matricial

Características Desejáveis:

- Linearidade.
- Precisão.
- Espessura (Densidade Uniforme).
- Intensidade independente de inclinação.
- Continuidade.
- Rapidez no traçado.

Conversão Matricial

Considerando o caso mais simples, segmentos de reta:

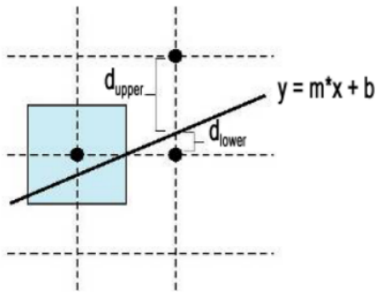
- Equação explícita da reta: $y = m \cdot x + b$.
- Inclinação da reta: $m = \frac{y_f - y_i}{x_f - x_i}$.
- Interseção do eixo y: $b = y_i - m \cdot x_i$.

Algoritmo de Bresenham

- Trabalha somente com inteiros.
- Incrementa x em intervalos unitários.
- Calcula o y correspondente.
- Partindo do ponto (x_k, y_k):
  - Considere o caminho da esquerda para direita.
  - Assuma $0 < |m| < 1$.
  - Próximo ponto pode ser (x_{k+1}, y_k) ou (x_{k+1}, y_{k+1}).

Algoritmo de Bresenham

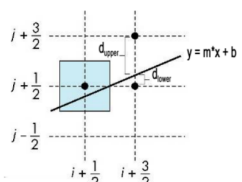
Regra de decisão: $\begin{cases} (d_{\text{lower}} - d_{\text{upper}}) \geq 0 \implies \text{usar o pixel superior} \\ (d_{\text{lower}} - d_{\text{upper}}) < 0 \implies \text{usar o pixel inferior} \end{cases}$



Algoritmo de Bresenham

Calculando d_{lower} e d_{upper}:

- Com base na equação da reta ($y = m \cdot x + b$), na posição $x_k + 1$, a coordenada y é calculada como $y = m \cdot (x_k + 1) + b$.
- Então, $d_{\text{lower}} = y - y_k = m \cdot (x_k + 1) + b - y_k$ e $d_{\text{upper}} = (y_k + 1) - y = y_k + 1 - m \cdot (x_k + 1) + b$.



Algoritmo de Bresenham

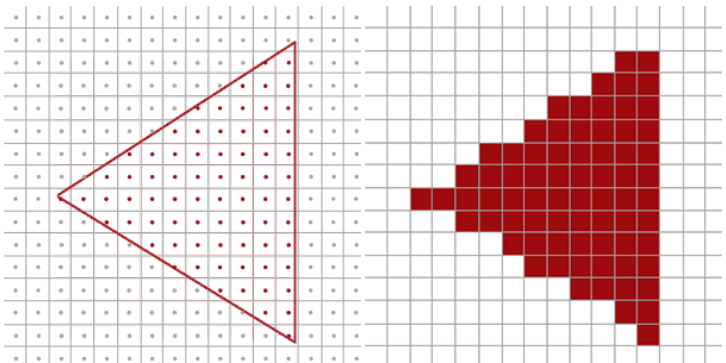
Regra de decisão:

- Um teste rápido para saber a proximidade:
  $p_k = d_{\text{lower}} - d_{\text{upper}} = 2m\,(x_k + 1) - 2y_k + 2b - 1.$
- Assim: $\begin{cases} p_k \geq 0 \implies \text{pixel superior} \\ p_k < 0 \implies \text{pixel inferior} \end{cases}$
- Porém, calcular m envolve operações de ponto flutuante: $m = \frac{y_{\text{end}} - y_0}{x_{\text{end}} - x_0} = \frac{\Delta y}{\Delta x}$.
- Então, substituindo m por $\frac{\Delta y}{\Delta x}$ e multiplicando tudo por Δx, tem-se:
  $p_k = \Delta x\,(d_{\text{lower}} - d_{\text{upper}}) = 2\Delta y \cdot x_k - 2\Delta x \cdot y_k + c$, onde $c = 2\Delta y + \Delta x\,(2b - 1)$ é um parâmetro constante independente da posição do pixel.

Antialiasing

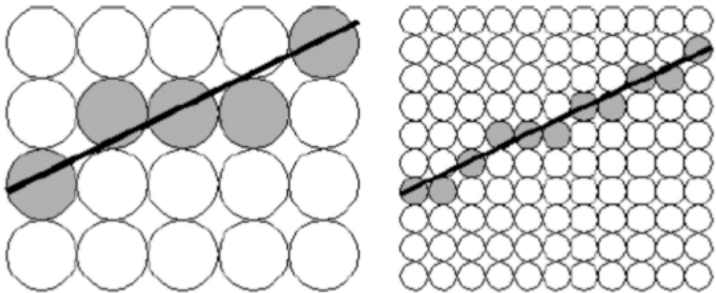
Correção de Traçado (*Antialiasing*)

Voltando ao problema de serrilhado na conversão matricial:



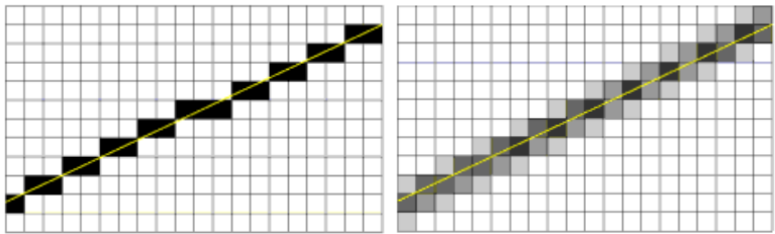
Correção de Traçado (*Antialiasing*)

A forma mais simples de resolver esse problema é aumentando a resolução.



Correção de Traçado (*Antialiasing*)

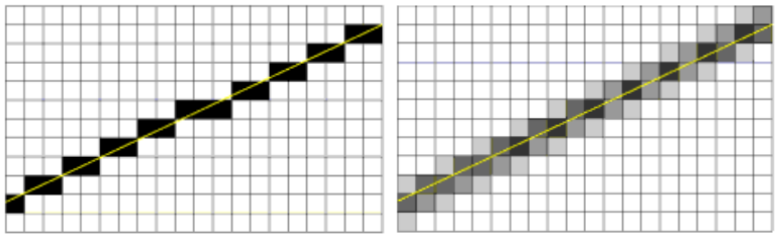
A estratégia mais simples é modificar a intensidade do pixel proporcionalmente.



Baseado em área: modifica a intensidade do pixel proporcional à cobertura do traçado.

Correção de Traçado (*Antialiasing*)

A estratégia mais simples é modificar a intensidade do pixel proporcionalmente.



Funciona bem para gráficos simples (linhas e curvas), mas é ineficiente para cenários 3D.

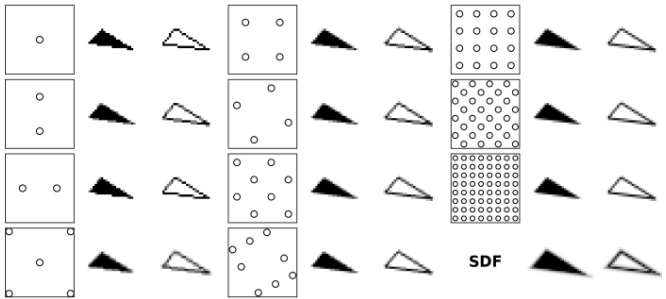
Correção de Traçado (*Antialiasing*)

Técnica *Supersampling*:

- Simula um monitor com maior resolução (mais pixeis).
- Na prática, cria (sub) pixeis de tamanhos menores.
- Converte a imagem para esse monitor simulado.
- A intensidade do pixel real (monitor real) é definida com base na quantidade de subpixeis cobertos.

Correção de Traçado (*Antialiasing*)

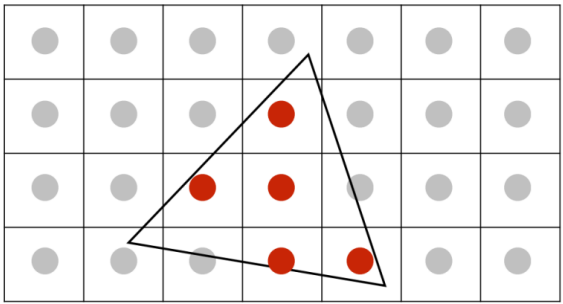
Técnica *Supersampling*:



Antialiasing com diferentes quantidades de subpixeis.

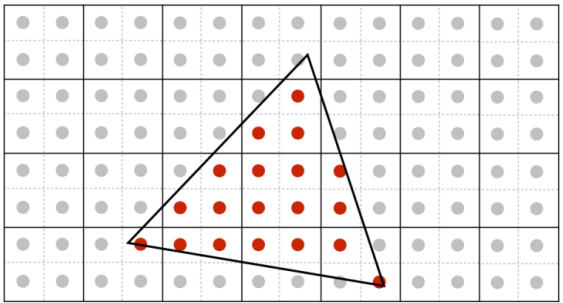
Correção de Traçado (*Antialiasing*)

Técnica *Supersampling*:



Correção de Traçado (*Antialiasing*)

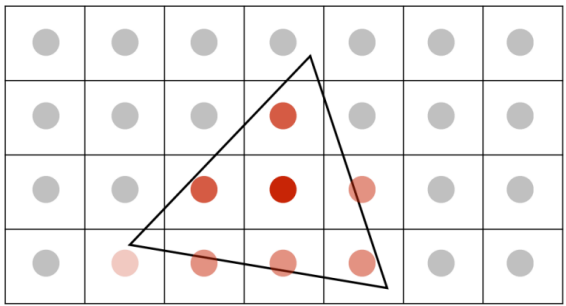
Técnica *Supersampling*:



Supersampling 2×2

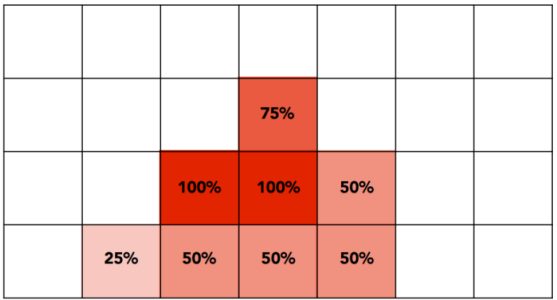
Correção de Traçado (*Antialiasing*)

Técnica *Supersampling*:



Correção de Traçado (*Antialiasing*)

Técnica *Supersampling*:



Correção de Traçado (*Antialiasing*)

Técnica *Supersampling*:

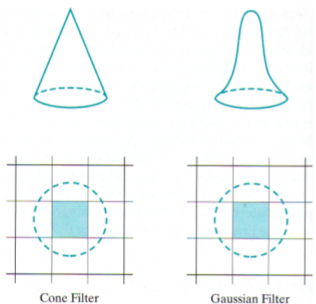
- Alguns autores defendem que o *supersampling* atinge qualidades próximas do ideal.
- No entanto, custos computacional e de memória inviabilizam seu uso para algumas aplicações (em tempo real).

Uma estratégia alternativa é *Multi-Sampling*:

- Técnicas para selecionar regiões de pixeis que precisam de correção (por exemplo, regiões de fronteira dos objetos).
- Implementada em muitas GPU.

Correção de Traçado (*Antialiasing*)

Técnica de Filtragem: uma superfície contínua de ponderação é usada para determinar a cobertura do pixel.



Correção de Traçado (*Antialiasing*)

DLSS (*Deep Learning Super Sampling*):

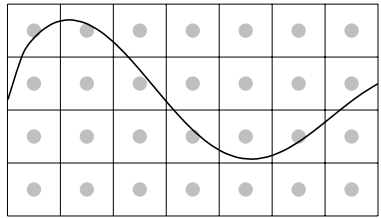
Material de base para a aula

- Introduction to Computer Graphics. Version 1.2, January 2018, David J. Eck. Source: `http://math.hws.edu/graphicsbook/`.
- Imagem Rasterização: `https://www.techpowerup.com/review/nvidia-geforce-turing-geforce-rtx-architecture/5.html`.
- Imagem *Antialiasing*: `https://learnopengl.com/Advanced-OpenGL/Anti-Aliasing`.
- Computação Gráfica: Aula 11. Slides de Ricardo M. Marcacini. Disciplina SCC0250/0650, ICMC/USP, 2021.

Exercícios

Exercícios I

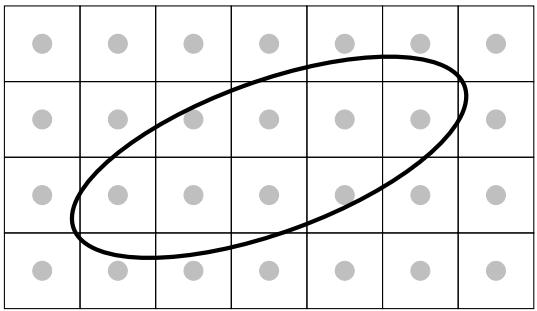
As respostas podem ser dadas na forma de figuras ou matrizes.



1. Apresente o resultado da rasterização simples para esta figura.
2. Suavize o traçado fazendo uma amostragem da área em torno da linha.

Exercícios II

3. Considerando a figura e matriz de conversão a seguir:



Apresente o resultado de uma operação de *supersampling* 2×2 para este caso.