

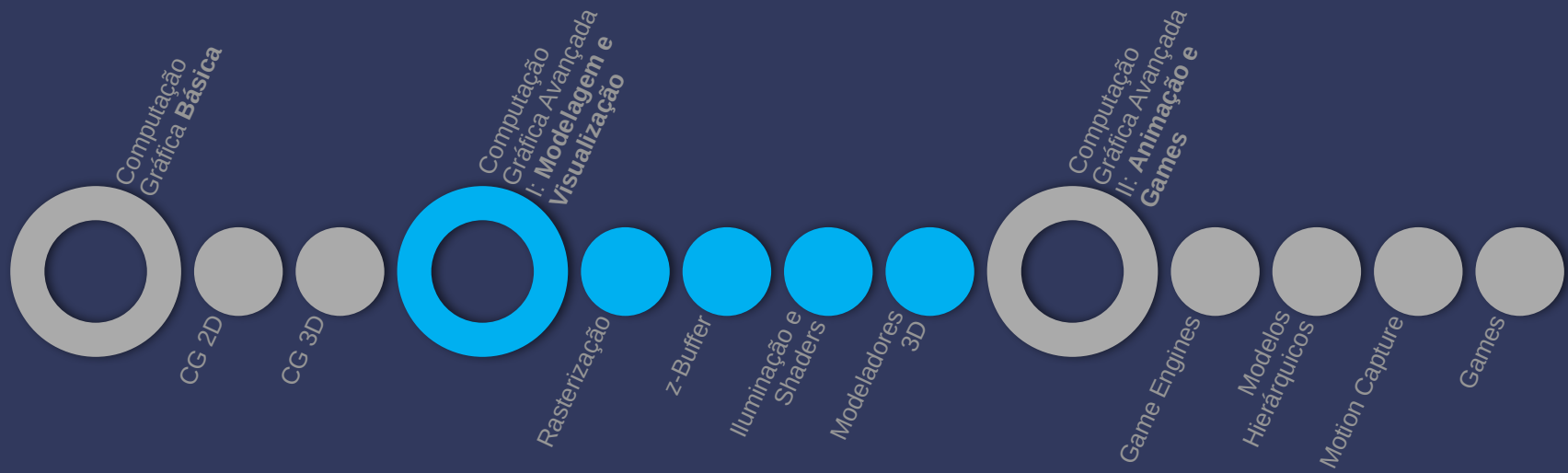
Buffer de Profundidade ou z-Buffer

ine5341 - Computação
Gráfica

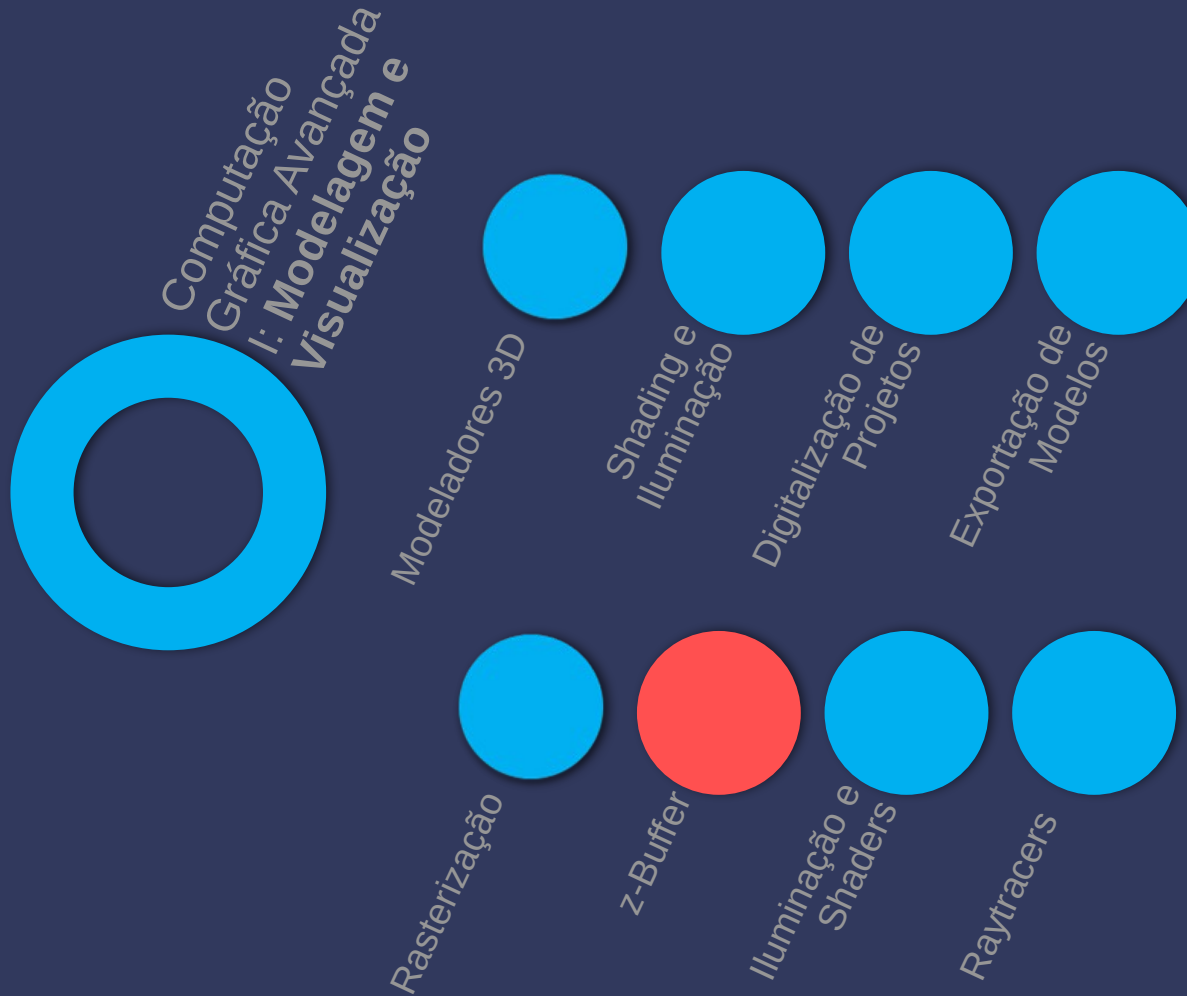
Capítulo 11

Parte 2

Timeline da Disciplina



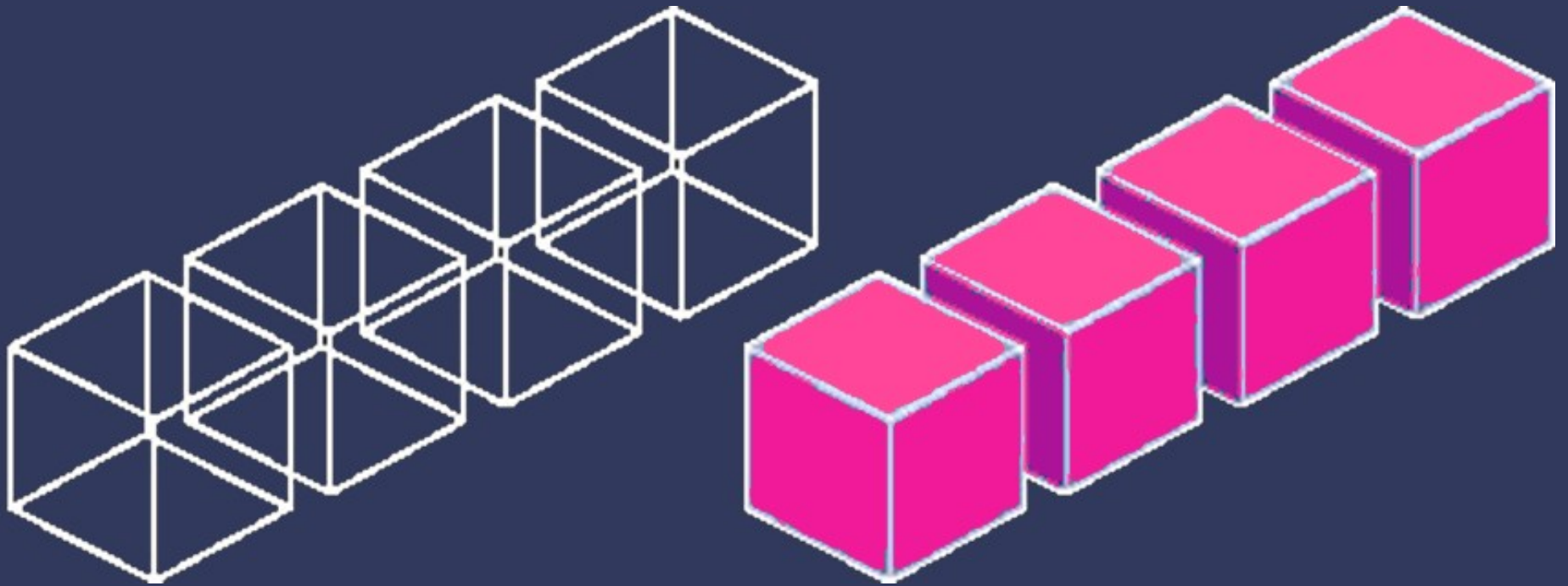
Estratégia dos Módulos Avançados



O Buffer de Profundidade

- Para que possamos realizar a representação acurada e realista de um modelo tridimensional complexo, em uma cena contendo vários objetos diferentes, é necessário que aquelas superfícies normalmente invisíveis de um determinado ângulo ou ponto no mundo sejam também renderizadas invisíveis no computador.
 - Este problema é normalmente chamado de problema de superfície visível (visible-surface problem), quando nosso objetivo é determinar quais superfícies de um objeto são visíveis de um determinado ângulo,
 - ou problema de superfície oculta (hidden-surface problem), quando nosso objetivo é determinar quais superfícies estão invisíveis neste momento.

O Buffer de Profundidade

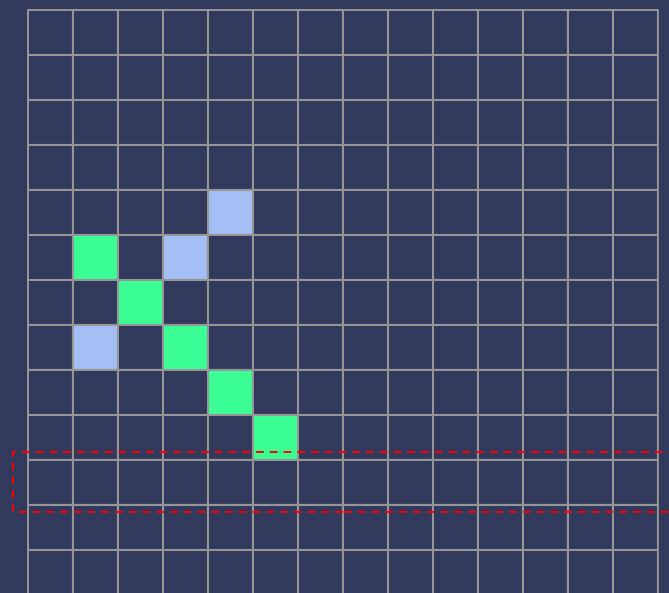


O z-Buffer

- Para todo pixel na Viewport nós mantemos um registro da profundidade (em termos de coordenada z) do objeto na cena que estiver mais próximo e um registro da cor que será utilizada ao se mostrar este ponto na tela do computador.
 - Toda vez que um novo polígono é processado, um valor de z e de intensidade são calculados para cada pixel que estiver dentro dos limites do polígono.
 - Se o valor de coordenada z obtido para aquele polígono for inferior ao valor de z armazenado para aquele pixel no buffer, então este objeto está mais próximo do que algum objeto anteriormente renderizado naquela posição e vai ocultar aquele objeto. Neste caso, substituímos o valor armazenado naquela posição do buffer por este novo valor.

O z-Buffer

- O buffer de profundidade é inicializado atribuindo-se a todos os seus elementos o valor infinito.
 - pode ser representado por algum valor positivo extremamente alto.
- A matriz de cores ou intensidades luminosas (para renderizações em preto e branco) é inicializada atribuindo-se a todos os seus pixels a cor preta (ou alguma outra cor “de fundo” arbitrária).
- Os polígonos a serem renderizados são testados contra este buffer, similarmente à maneira de “encontrar o maior valor em um array”.



Viewport

$-\infty$

$-\infty$

$-\infty$

$-\infty$

$-\infty$

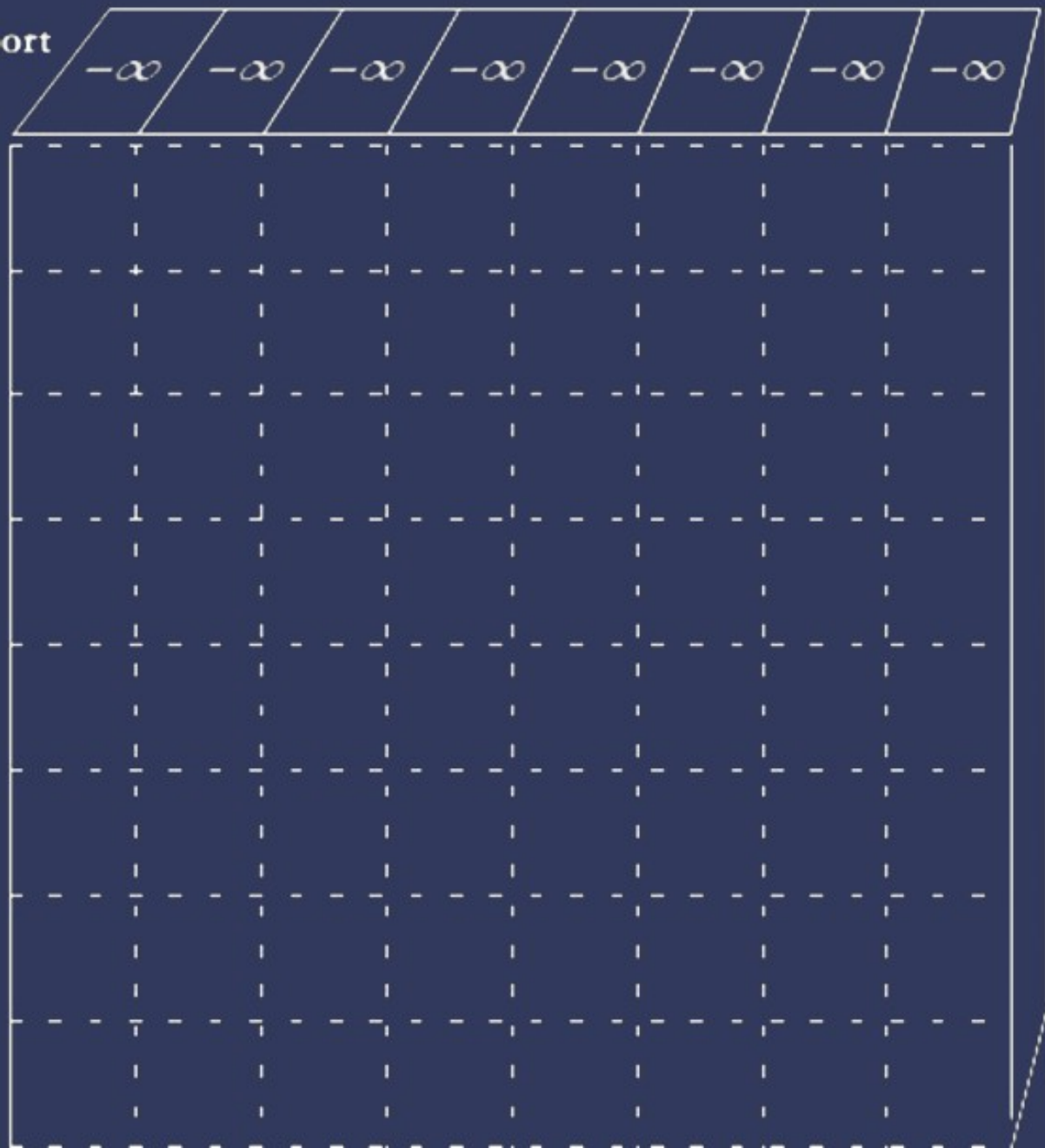
$-\infty$

$-\infty$

$-\infty$

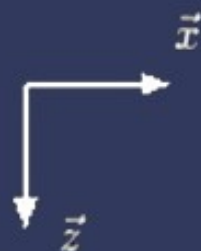
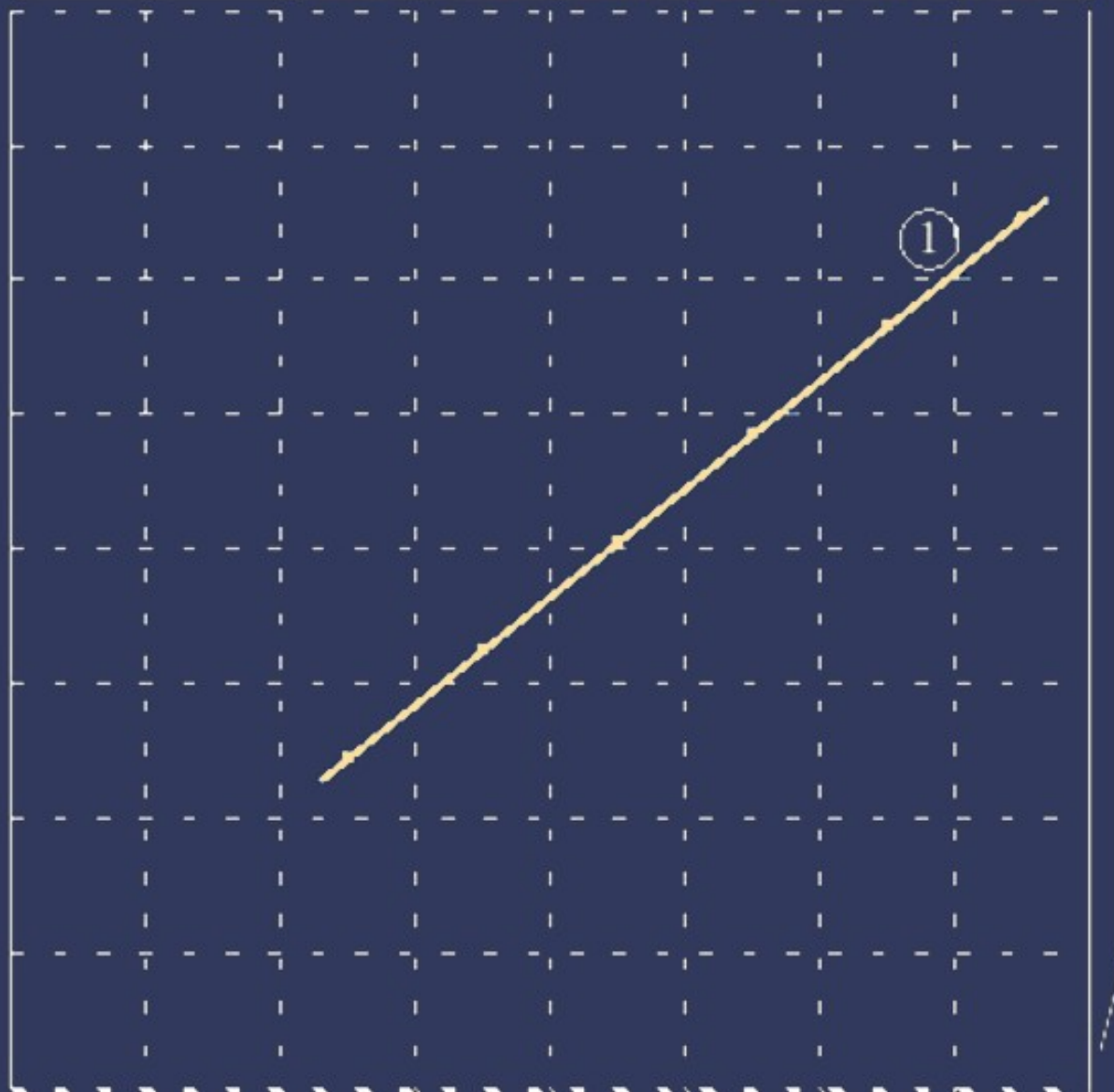
x_1

z_1



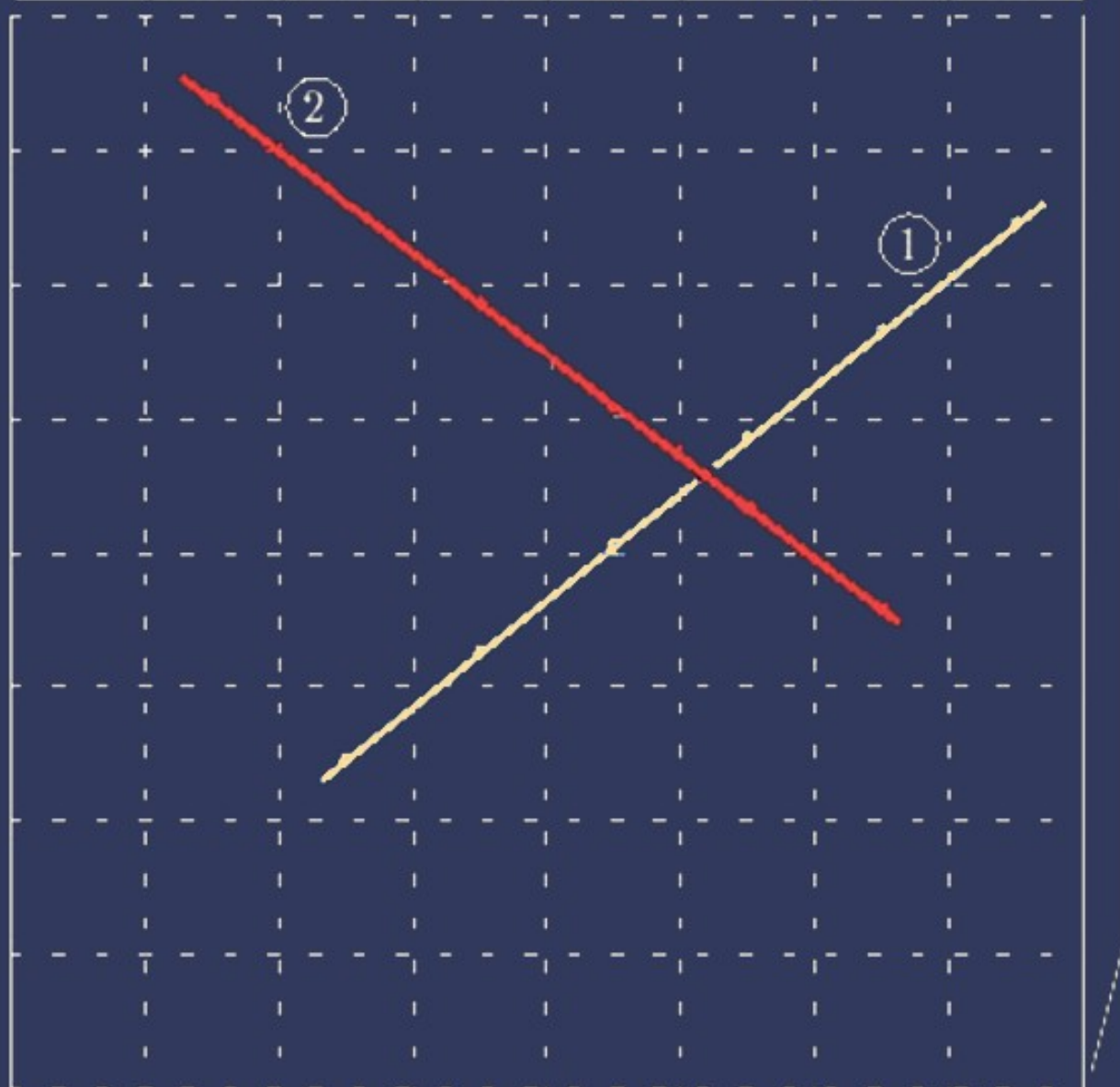
Viewport

$-\infty$	$-\infty$	① 5.5	① 4.7	① 3.9	① 3.1	① 2.3	① 1.5
-----------	-----------	----------	----------	----------	----------	----------	----------



Viewport

$-\infty$	2	2	2	2	1	1	1
	0,6	1,4	2,3	2,9	3,1	2,3	1,5



Algoritmo z-Buffer

Dados:

Lista de polígonos {P1, P2, ..., Pn}

Matriz z-buffer[x,y] inicializado com +infinito

Matriz Intensidade[x,y]

início

para cada polígono P na lista de polígonos faça {

para cada pixel (x,y) que intercepta P faça {

 calcule profundidade-z de P na posição (x,y)

se prof-z < z-buffer[x,y] então {

 Intensidade[x,y] = intensidade de P em (x,y)

 z-buffer[x,y] = prof-z

fim se

fim para

fim para

Desenhe Intensidade

fim

Z-Buffer e Conversão por Varredura

- O algoritmo Z-Buffer se encaixa naturalmente no algoritmo de Conversão por Varredura visto anteriormente.
- Basicamente, dentro do passo de “Preencher cada pixel da linha de varredura”, basta enviar o pixel ao algoritmo de Z-Buffer. O valor do pixel em Z já é conhecido pelas equações 10.4 e 10.3.

Resumindo (Algoritmo de Pixel Shading v.2):

Para cada Polígono Q:

Para cada Trapézio T que forma Q:

Aplique 10.1 para calcular o ponto inicial de ambas bordas de T.

Aplique 10.2 para calcular os incrementos em x e em z entre as linhas de varredura.

Para cada par de pontos extremos das bordas esquerda e direita de T:

Aplique 10.3 e 10.4 para calcular o valor de Z do primeiro pixel da linha e o incremento de z entre os pixels.

Preencha a linha entre os pontos extremos incrementando 1 em x e o valor de 10.3

em z, enviando cada pixel obtido para o algoritmo de Z-buffer

fim-para

Incremente x, y e z dos pontos iniciais para a próxima linha de varredura.

fim-para

Fim-para



Atribuição-Uso Não-Comercial-Compartilhamento pela Licença 2.5 Brasil

Você pode:

- copiar, distribuir, exhibir e executar a obra
- criar obras derivadas

Sob as seguintes condições:

Atribuição — Você deve dar crédito ao autor original, da forma especificada pelo autor ou licenciante.

Uso Não-Comercial — Você não pode utilizar esta obra com finalidades comerciais.

Compartilhamento pela mesma Licença — Se você alterar, transformar, ou criar outra obra com base nesta, você somente poderá distribuir a obra resultante sob uma licença idêntica a esta.

Para ver uma cópia desta licença, visite <http://creativecommons.org/licenses/by-nc-sa/2.5/br/> ou mande uma carta para Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.



UFSC