

Algoritmo de z-Buffer

- Epifanio Dinis Benitez
- Felipe da Motta Varga

Roteiro

- Introdução
- O Algoritmo
- Vantagens
- Desvantagens
- Conclusão
- Referências Bibliográficas

Introdução

- A representação eficiente de cenários complexos necessita de algoritmos de visualização para representar somente as partes visíveis do cenário.
- Alguns algoritmos conhecidos além do z-Buffer são o de Scan-line, Ray-Tracing.. entre outros.
- As rotinas do OpenGL se baseiam neste algoritmo.
- Algumas placas de vídeo o implementam internamente.

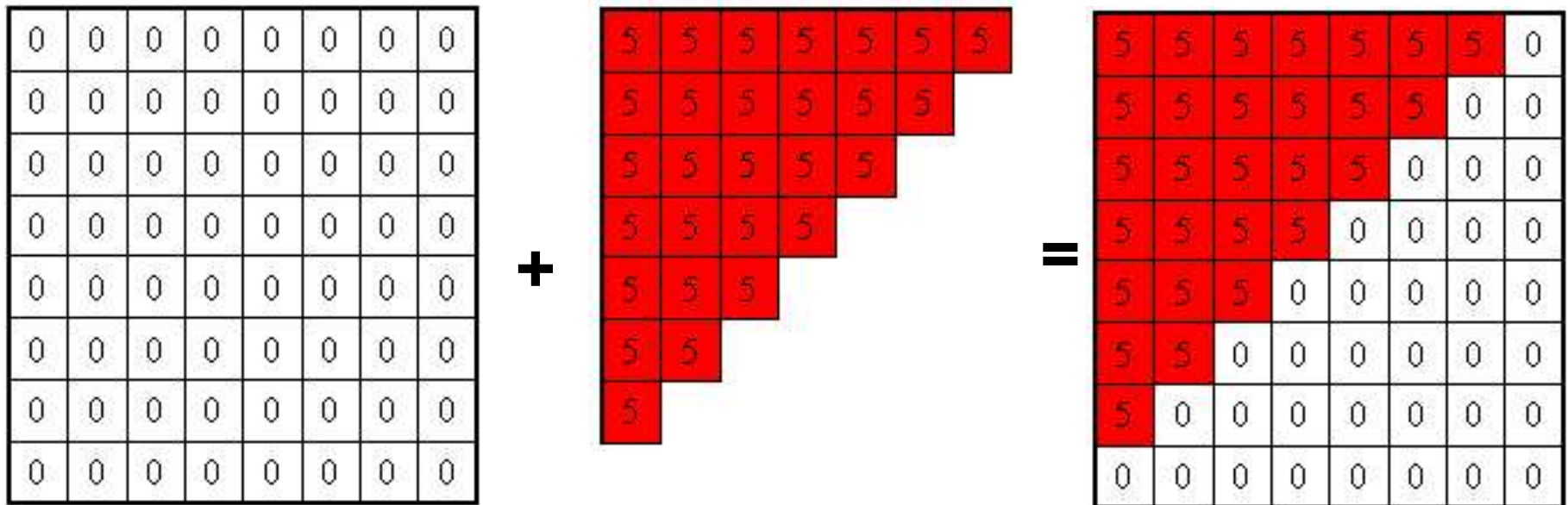
O Algoritmo (1/5)

- Conhecido também como algoritmo de profundidade (*depth-buffer*).
- Sua implementação é simples tanto em hardware como em software.
- Necessita de duas matrizes auxiliares do mesmo tamanho da viewport:
 - De cores
 - z-Buffer

O Algoritmo (2/5)

- Inicialize o plano com a cor de fundo
- Inicialize o z-buffer com valores nulos: $Z(x,y) = 0$;
- Para cada polígono
 - Para pixel projetado do polígono
 - pz = valor de z do polígono nas coords. (x,y) do pixel
 - Se $pz \geq Z(x,y)$
 - Troque $Z(x,y) = pz$
 - Desenhe na tela o pixel (x,y) com a cor do polígono correspondente ao pixel projetado em (x,y)

O Algoritmo (3/5)



O Algoritmo (4/5)

5	5	5	5	5	5	5	0
5	5	5	5	5	5	0	0
5	5	5	5	5	0	0	0
5	5	5	5	0	0	0	0
5	5	5	0	0	0	0	0
5	5	0	0	0	0	0	0
5	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

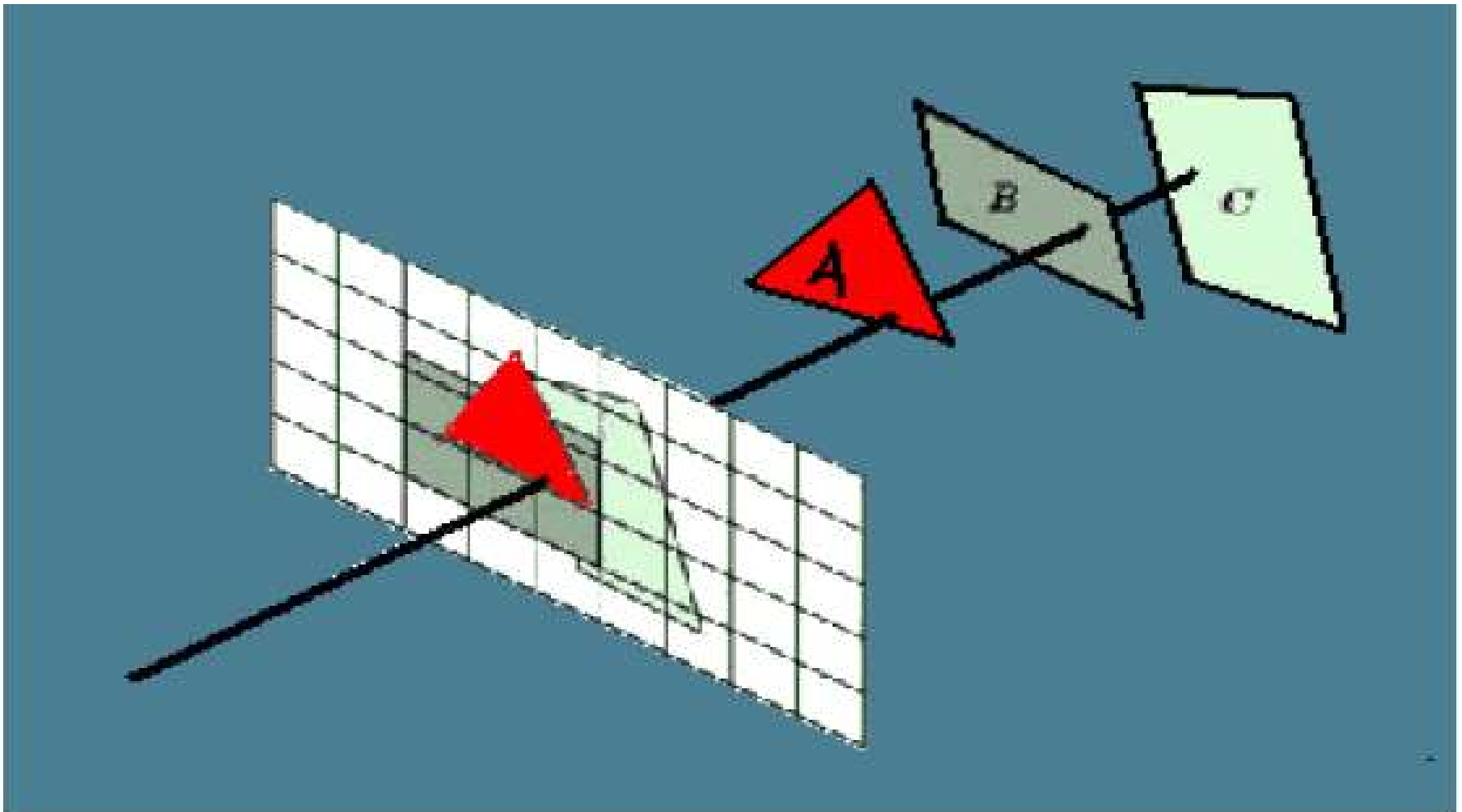
+

3					
4	3				
5	4	3			
6	5	4	3		
7	6	5	4	3	
8	7	6	5	4	3

=

5	5	5	5	5	5	5	0
5	5	5	5	5	5	0	0
5	5	5	5	5	0	0	0
5	5	5	5	0	0	0	0
6	5	5	3	0	0	0	0
7	6	5	4	3	0	0	0
8	7	6	5	4	3	0	0
0	0	0	0	0	0	0	0

O Algoritmo (5/5)



Vantagens

- Fácil implementação
- Permite uma comparação por pixel em cada polígono.
- O tempo do cálculo da superfície visível é aproximadamente independente do número de polígonos do objeto.
- Não necessita ordenação prévia.

Desvantagens

- Necessita de um razoável espaço de memória para armazenamento do z-buffer e da matriz de cores.
- Pode repintar varias vezes o mesmo ponto.