

# KD-Tree para Otimização de Raytracing

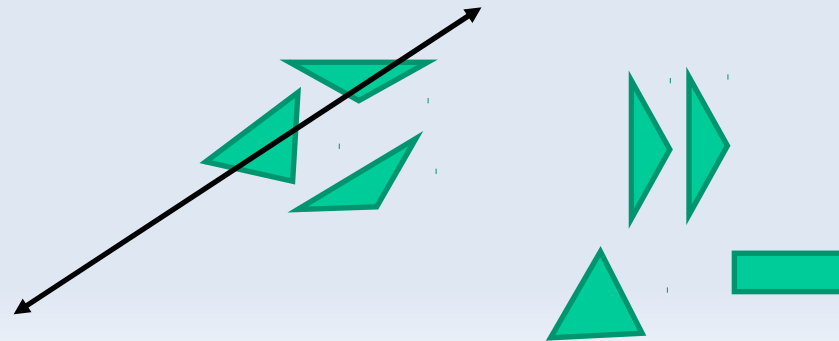
**Vanessa Leite e Ian Medeiros**

# Raytracing

- O conceito de *ray tracing* foi introduzido na Computação Gráfica em 1980 por Whitted
- O termo ray tracing é usado livremente para designar uma infinidade de diferentes abordagens ao problema de *rendering*, desde que baseados no princípio de intersecção de uma semireta (raio) com primitivas geométricas

# Raytracing: Complexidade

- O algoritmo usado para determinar a visibilidade ao longo de um raio requer que cada raio seja intersectado com TODAS as primitivas geométricas da cena.
- O tempo para cada raio é, portanto, linear com o número de primitivas  $N$ :  $T_{\text{ray}} = O(N)$
- No entanto, cada raio não passa próximo da maioria das primitivas

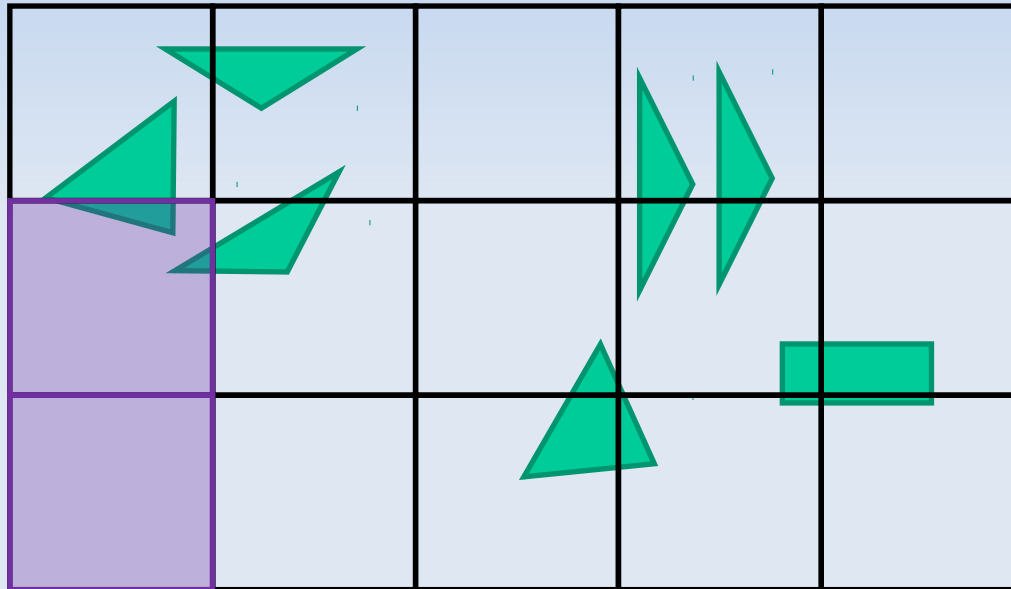


# Raytracing: Estruturas de Aceleração

- O objetivo das estruturas de aceleração é diminuir o número de interseções por raio.
- De duas maneiras:
  1. Permitindo a rejeição rápida e simultânea de grupos de primitivas
  2. Se possível, ordenando o processo de procura (interseções), tal que as primitivas mais próximas da origem do raio sejam processadas primeiro, evitando processar as mais distantes se for encontrada uma intersecção
- Abordagens:
  - **SUBDIVISÃO DO ESPAÇO:** grades regulares, octrees, kd-tree  
Permitem aplicar os critérios 1 e 2
  - **SUBDIVISÃO DOS OBJECTOS:** bounding volume hierarchy (BVH)  
Permite aplicar apenas o critério 1

# Raytracing: Grade Regular

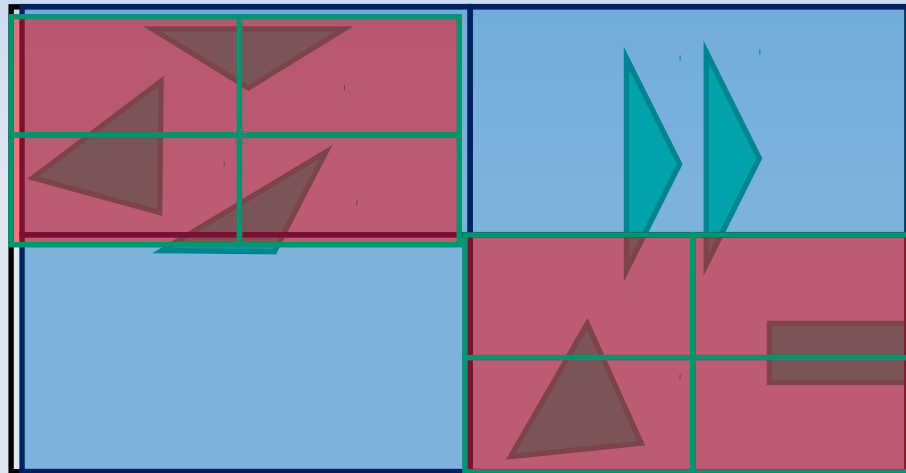
- O espaço 3D é particionado impondo uma grade regular que o subdivide em voxels (*volume elements*). Todos os voxels têm a mesma dimensão.



- Construção muito rápida
- Travessia pouco eficiente devido à má distribuição das primitivas pelos voxels

# Raytracing: Octree

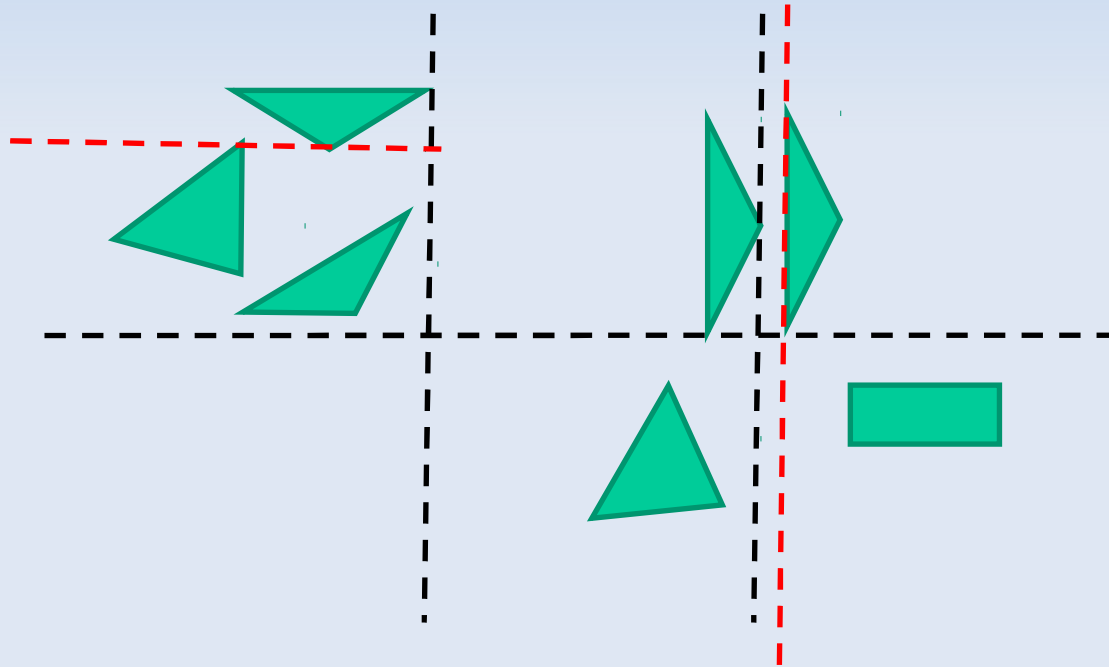
- O espaço é hierarquicamente e adaptativamente subdividido em 8 voxels



- Compromisso entre tempo de construção e eficiência da travessia.

# Raytracing: Kd-tree

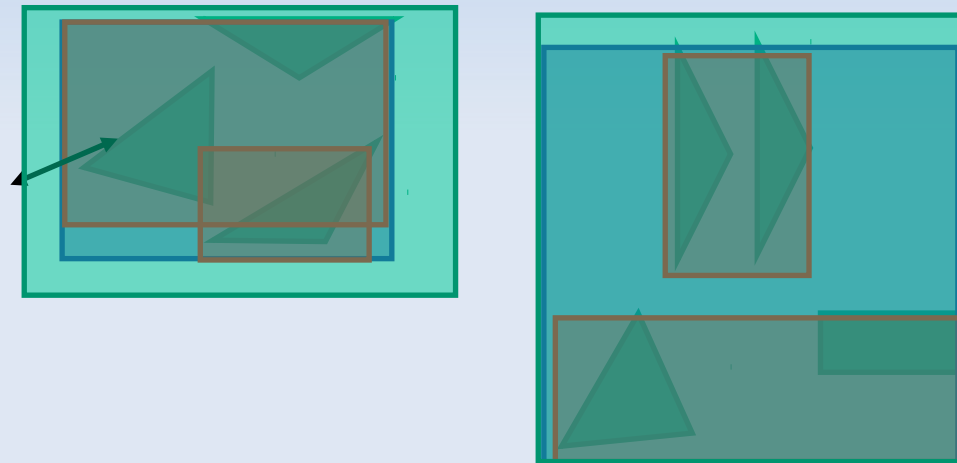
- O espaço é subdividido em 2 por um plano. Cada um dos sub-espacos resultantes é depois subdividido da mesma forma, até atingir um determinado critério de parada



- Travessia mais eficiente se o critério de subdivisão for apropriado (ex: SAH)
- Quanto mais sofisticado for o critério de divisão maior o tempo necessário para construir a árvore.

# Raytracing: Bounding Volume Hierarchy

- Os objetos são agrupados dentro de Bounding Volumes. Cada um destes grupos é depois hierarquicamente subdividido por outros volumes



- Não ordena o espaço
- Construção semelhante à kd-tree
- Travessia ligeiramente inferior à kd-tree



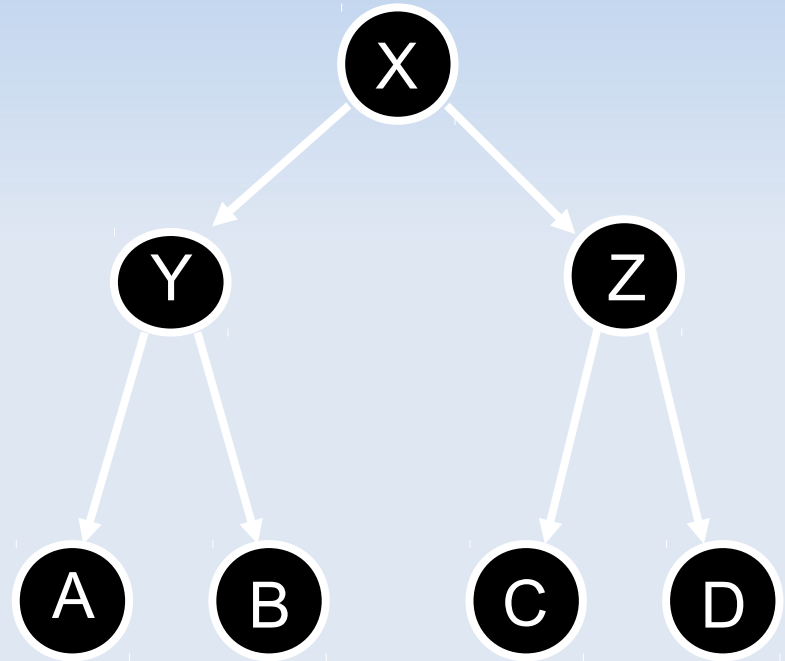
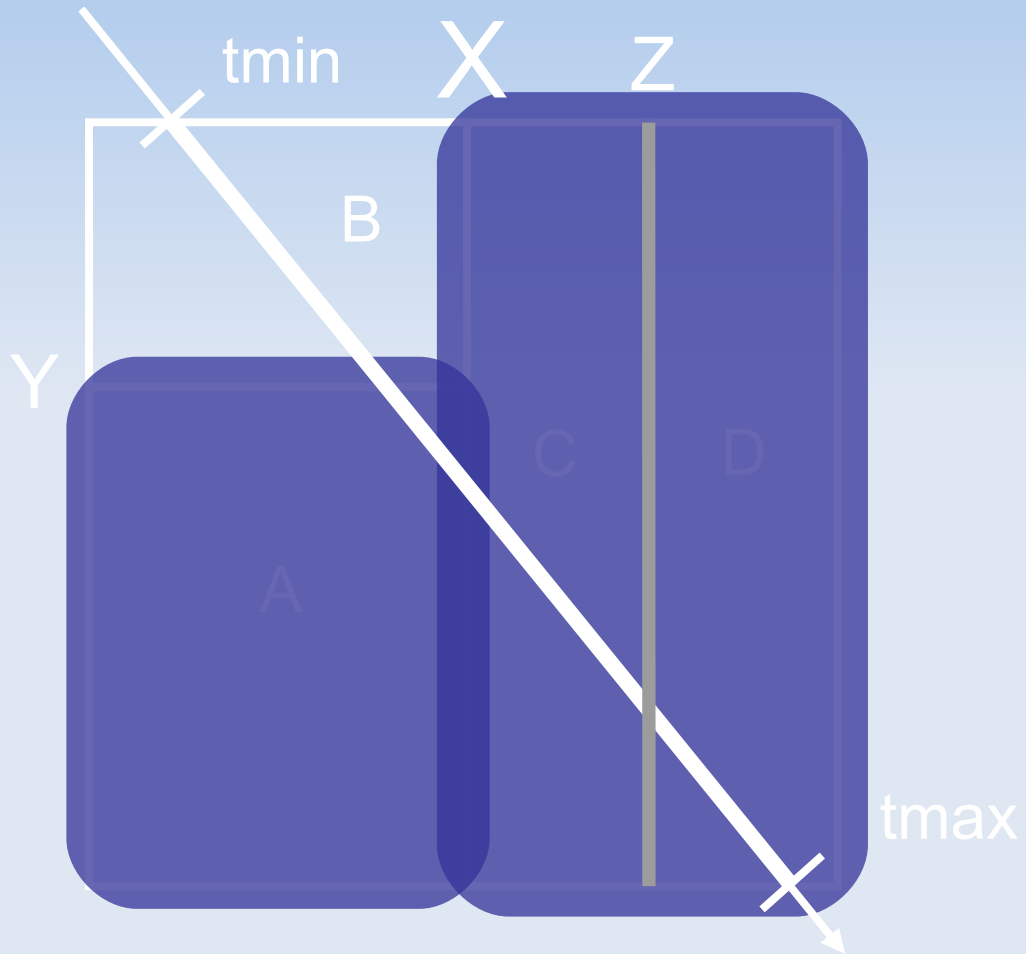
# Raytracing: Complexidade 2

- A complexidade do *ray tracing* com uma estrutura de aceleração apropriada é logarítmica com o número de primitivas geométricas  $N$ :

$$T_{\text{ray}} = O(\log N)$$

- O tempo de construção depende do critério de subdivisão do espaço / agrupamento das primitivas
- Critérios sofisticados (ex: SAH) resultam em travessias eficientes, mas exigem tempos de construção muito elevados
- O tempo de reconstrução/reajustamento de uma estrutura de aceleração pode impedir a sua utilização em contextos interativos.

# KD-Tree



# Referencias

- Highly Parallel Fast KD-tree Construction for Interactive Ray Tracing of Dynamic Scenes - EUROGRAPHICS 2007 / D. Cohen-Or and P. Slavík
- Interactive k-D Tree GPU Raytracing (Daniel Reiter Horn, Jeremy Sugerman, Mike Houston, Pat Hanrahan – Stanford University)
- Construção eficiente de kd-trees para primitivas triangulares (Alessandro Ribeiro da Silva, Wallace Santos Lages - UFMG)
- kD-Tree Traversal Implementations for Ray Tracing on Massive Multiprocessors: a Comparative Study - 21st International Symposium on Computer Architecture and High Performance Computing / Artur Santos, João Marcelo Teixeira, Thiago Farias, Veronica Teichrieb, Judith Kelner
- On fast Construction of SAH-based Bounding Volume Hierarchies / Ingo Wald
- Ray Tracing Interactivo: Estrutura de Aceleração - Ademar Gonçalves - Universidade do Minho, Braga, 2009