



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

Lucas Rodrigo da Silva Suppes

Algoritmos e Estruturas de dados para visualização de polígonos

Florianópolis
2020

Lucas Rodrigo da Silva Suppes

Algoritmos e Estruturas de dados para visualização de polígonos

Tese submetida ao Programa de Pós-Graduação em Ciências da Computação da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Ciências da Computação.

Orientador: Prof. Álvaro Junio Pereira Franco , Dr.

Florianópolis
2020

Ficha de identificação da obra

A ficha de identificação é elaborada pelo próprio autor.

Orientações em:

<http://portalbu.ufsc.br/ficha>

Lucas Rodrigo da Silva Suppes

Algoritmos e Estruturas de dados para visualização de polígonos

O presente trabalho em nível de Bacharelado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof.(a) xxxx, Dr(a).

Instituição xxxx

Prof.(a) xxxx, Dr(a).

Instituição xxxx

Prof.(a) xxxx, Dr(a).

Instituição xxxx

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de Bacharel em Ciencias da Computação.

Coordenação do Programa de
Pós-Graduação

Prof. Álvaro Junio Pereira Franco , Dr.
Orientador

Florianópolis, 2020.

Este trabalho é dedicado aos meus colegas de classe e
aos meus queridos pais.

AGRADECIMENTOS

Inserir os agradecimentos aos colaboradores à execução do trabalho.

[illegible]

*“Texto da Epígrafe.
Citação relativa ao tema do trabalho.
É opcional. A epígrafe pode também aparecer
na abertura de cada seção ou capítulo.
Deve ser elaborada de acordo com a NBR 10520.”
(SOBRENOME do autor da epígrafe, ano)*

RESUMO

Um estudo das estruturas e técnicas comuns para otimização para renderização em janelas. Tendo em mente o constante crescimento de polígonos dos objetos a serem renderizados, há uma necessidade de que os objetos sejam acessados e "encontrados" de forma rápida e eficiente.

Palavras-chave: Estruturas de Dados. Árvores. Geometria Computacional.

ABSTRACT

A study of the most common techniques for optimizing rendering on windows. Beign aware of the constant growth of the polygon count and the size of graphical applications, and the necessity for easy and quick access to the objects.

Keywords: Data Structure. Tree. Computational Geometry.

LISTA DE FIGURAS

Figura 1 – Arvore <i>kdimensional</i> - 3D	15
Figura 2 – Busca em alcance <i>dimensional</i> - 2D	16
Figura 3 – Arvore 2D	17

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

RAM	Memória de Acesso Aleatório
V-RAM	Video RAM
VSD	Visible-Surface Determination

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVOS	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	ARVORES KD	15
2.1.1	Construção da Árvore	15
2.1.2	Busca com alcance	17
2.2	ARVORE DE INTERVALOS	17
2.3	ARVORE DE SEGMENTOS	17
3	SEÇÃO	18
4	CONCLUSÃO	19
	APÊNDICE A – DESCRIÇÃO 1	20
	ANEXO A – DESCRIÇÃO 2	21

1 INTRODUÇÃO

A contagem de polígonos em artefatos gráficos em filmes, jogos, pesquisas médicas e científicas seguem em crescimento vertiginosa e apesar do Hardware acompanhar este crescimento, existem claras restrições com relação a Video RAM (V-RAM) e a Memória de Acesso Aleatório (RAM). Com essas restrições em mente, são necessárias estruturas de dados que permitam o acesso rápido dessas figuras, consultas e armazenamento. A seleção e determinação de faces visíveis Visible-Surface Determination (VSD)

1.1 OBJETIVOS

O objetivo deste trabalho visa a busca rápida de estruturas geométricas. Visando aplicações gráficas em tempo real. Quando a câmera da aplicação precisa saber quais figuras geométricas precisam ser desenhadas tendo apenas a informação da posição da câmera e das coordenadas do mundo, este artigo visa o estudo de algoritmos para a solução deste tipo de problema.

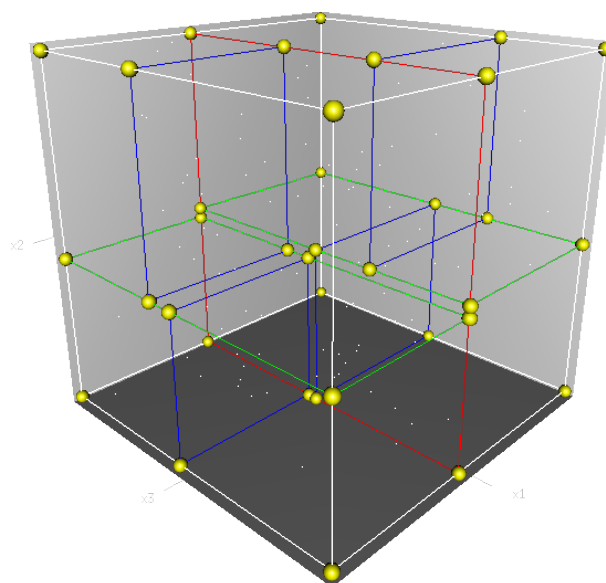
2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo apresenta-se uma visão de como funcionam os algoritmos e estruturas de dados base para o desenvolvimento de algoritmos para renderização em janela acesso rápido para estruturas geométricas. São os algoritmos e estruturas de dados base para a construção de um controle de janela para acesso rápido de polígonos em janela.

2.1 ARVORES KD

Uma Árvore KD é uma árvore binária onde cada folha é um ponto *k-dimensional*. E cada nó não-folha é um corte do espaço, representando implicitamente um hiperplano. Pontos à esquerda desse hiperplano estão na subárvore da esquerda, e respectivamente para o lado direito. Cada nó é associado com uma das *k* dimensões. Então, a citar um exemplo, se dado nó divide o eixo *x*, a subárvore à esquerda contém os pontos com o eixo *x* menor que o ponto de corte.

Figura 1 – Árvore *kdimensional* - 3D



Fonte – GPL

2.1.1 Construção da Árvore

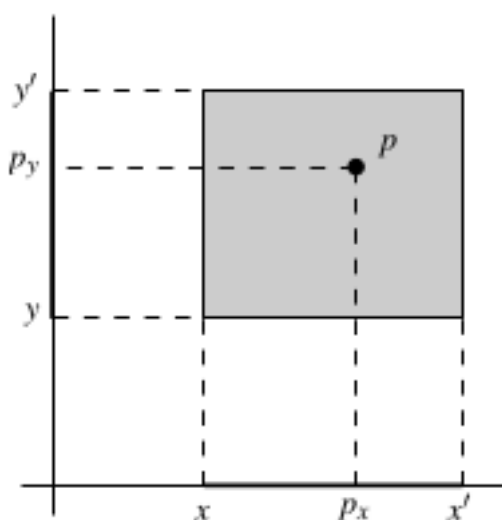
A termos didáticos segue a construção da árvore KD de 2 dimensões. Seja *P* o conjunto de *n* pontos em um plano.

Uma busca de alcance 2-dimensional em P é uma busca de quais pontos da busca estão entre o retângulo de busca $[x, x'] \times [y, y']$. Um ponto $p := (p_x, p_y)$ está dentro do retângulo de busca se e somente se:

$$p_x \in [x, x'] \text{ e } p_y \in [y, y']$$

Podemos dizer que uma busca 2-dimensional é composta de duas sub-buscas 1-dimensional, uma no eixo x -coordenada de um dos pontos e um na y -coordenada.

Figura 2 – Busca em alcance *dimensional* - 2D

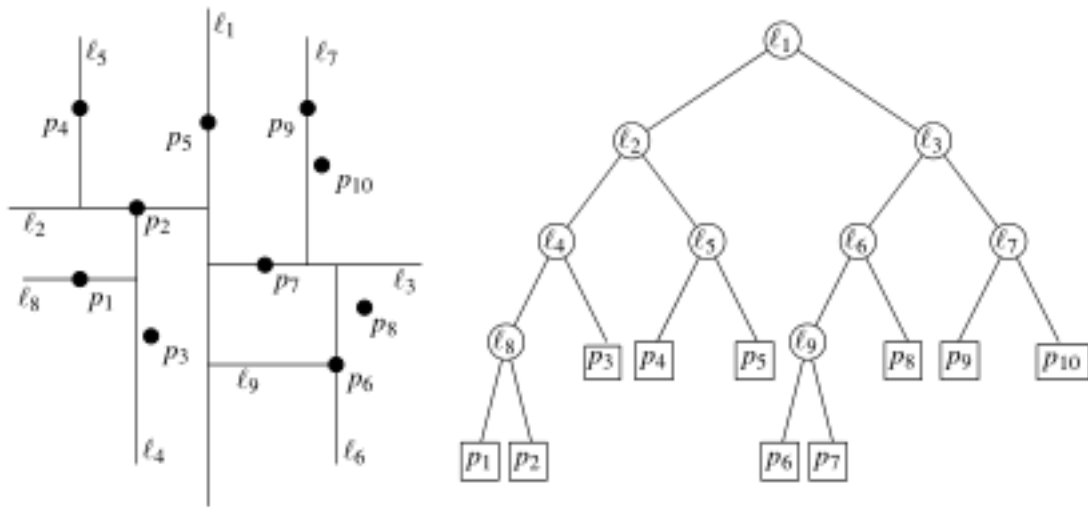


Fonte – Computational Geometry

Na construção de uma árvore para 2 dimensões, cada ponto tem uma x -coordenada e uma y -coordenada. Seguimos então escolhendo um eixo inicial e salvando o valor de corte deste eixo que divide os pontos deste eixo em dois conjuntos, e no próximo nível da árvore, alterna-se o eixo e repete-se o processo recursivamente.

Na raiz ordena-se todos os pontos e divide o conjunto de pontos P com uma linha vertical l que divide os pontos pelo eixo x . Guarda-se o valor de x_v no nodo e alterna-se o eixo. Agora, recursivamente repete o processo para os dois subconjuntos de pontos: Ordena-se os pontos pelo eixo y e divide o conjunto e encontra-se a reta horizontal que subdivide os pontos e guarda no nodo do valor de corte y . A condição de parada é até quando o conjunto de pontos restantes contiver apenas um ponto. Este sendo então o nodo folha.

Figura 3 – Arvore 2D

**Algorithm 1** ConstroiArvoreKD(*Pontos*, *profundidade*)

```

if P contem apenas um ponto then
  return P
else

  if profundidade é par then
    Divide P em dois subconjuntos com um linha vertical l pela mediana da x –
    coordenada dos pontos em P. Seja P1 o conjunto dos pontos à esquerda de l e
    seja P2 o conjunto de pontos à direita de l.
  else
    Divide P em dois subconjuntos com um linha horizontal l pela mediana da
    y – coordenada dos pontos em P. Seja P1 o conjunto dos pontos acima de l e
    seja P2 o conjunto de pontos à abaixo de l.
  end if
end if

  vesquerda ← ConstroiArvoreKD(P1, profundidade + 1)
  vdireita ← ConstroiArvoreKD(P2, profundidade + 1) Cria um nodo v guardando l e
  fazendo vesquerda o nodo da esquerda de v e fazendo vdireita nodo da direita de v.
return v

```

2.1.2 Busca com alcance**2.2 ARVORE DE INTERVALOS****2.3 ARVORE DE SEGMENTOS**

3 SEÇÃO

Este *template* contém algumas seções criadas na tentativa de facilitar seu uso. No entanto, não há um limite máximo ou mínimo de seção a ser utilizado no trabalho. Cabe a cada autor definir a quantidade que melhor atenda à sua necessidade.

4 CONCLUSÃO

As conclusões devem responder às questões da pesquisa, em relação aos objetivos e às hipóteses. Devem ser breves, podendo apresentar recomendações e sugestões para trabalhos futuros.

APÊNDICE A – DESCRIÇÃO 1

Textos elaborados pelo autor, a fim de completar a sua argumentação. Deve ser precedido da palavra APÊNDICE, identificada por letras maiúsculas consecutivas, travessão e pelo respectivo título. Utilizam-se letras maiúsculas dobradas quando esgotadas as letras do alfabeto.

ANEXO A – DESCRIÇÃO 2

São documentos não elaborados pelo autor que servem como fundamentação (mapas, leis, estatutos). Deve ser precedido da palavra ANEXO, identificada por letras maiúsculas consecutivas, travessão e pelo respectivo título. Utilizam-se letras maiúsculas dobradas quando esgotadas as letras do alfabeto.