Equipe abnTEX2

Modelo Canônico de Trabalho Acadêmico com abnTEX2

Brasil

2014, v-1.9.2

Equipe abnTEX2

Modelo Canônico de Trabalho Acadêmico com abnTEX2

Modelo canônico de trabalho monográfico acadêmico em conformidade com as normas ABNT apresentado à comunidade de usuários LATEX.

Universidade do Brasil – UBr

Faculdade de Arquitetura da Informação

Programa de Pós-Graduação

Orientador: Lauro César Araujo

Coorientador: Equipe abnTEX2

Brasil

2014, v-1.9.2

Ficha de identificação da obra

A ficha de identificação é elaborada pelo próprio autor.

Orientações em:

http://portalbu.ufsc.br/ficha

Equipe abnTEX2

Modelo Canônico de Trabalho Acadêmico com abnTEX2

Modelo canônico de trabalho monográfico acadêmico em conformidade com as normas ABNT apresentado à comunidade de usuários LATEX.

Trabalho aprovado. Brasil, 24 de novembro de 2012:

Lauro César Araujo
Orientador

Professor
Convidado 1

Professor
Convidado 2

Brasil

2014, v-1.9.2

Este trabalho é dedicado às crianças adultas que, quando pequenas, sonharam em se tornar cientistas.

Agradecimentos

Os agradecimentos principais são direcionados à Gerald Weber, Miguel Frasson, Leslie H. Watter, Bruno Parente Lima, Flávio de Vasconcellos Corrêa, Otavio Real Salvador, Renato Machnievscz[1] e todos aqueles que contribuíram para que a produção de trabalhos acadêmicos conforme as normas ABNT com LaTeX fosse possível.

Agradecimentos especiais são direcionados ao Centro de Pesquisa em Arquitetura da Informação[2] da Universidade de Brasília (CPAI), ao grupo de usuários *latex-br*[3] e aos novos voluntários do grupo *abnTeX2*[4] que contribuíram e que ainda contribuirão para a evolução do abnTeX2.

1 Os nomes dos integrantes do primeiro projeto abnTeX foram extraídos de <http://codigolivre.org.br/projects/abntex/>
2 <http://www.cpai.unb.br/>
3 <http://groups.google.com/group/latex-br>
4 <http://groups.google.com/group/abntex2> e <http://abntex2.googlecode.com/>

"Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12, 2)

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. Arvores. Geometria Computacional.

Abstract

A study of the most common tecniques for optimizing rendering on windows. Beign awere 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 ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ABNT	Associação Brasileira de Normas Técnicas
abnTeX	ABsurdas Normas para TeX

Lista de símbolos

Γ	Letra grega Gama
Λ	Lambda
ζ	Letra grega minúscula zeta
$\in$	Pertence

Sumário

Parte I

Preparação da pesquisa

1 Introdução

A contagem de polígonos em artefatos gráficos em filmes, jogos, pesquisas medicas e cientificas seguem em crescimento vertiginosa e apesar do Hardware acompanhar este crescimento, existem claras restrições com relação a V-RAM e a 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 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.

[1]

Parte II

Referenciais teóricos

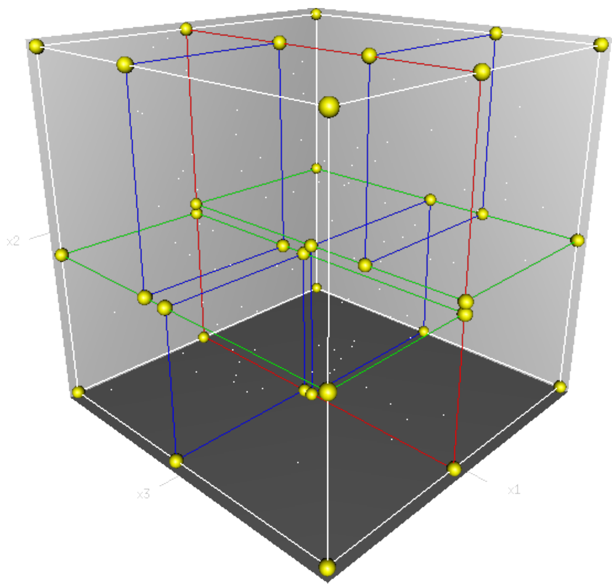
2 Estruturas de busca de pontos

Neste capitulo 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. Sao os algoritmos e estruturas de dados base para a construção de um controle de janela para acesso rapido de poligonos em janela.

2.1 Arvores KD

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

Figura 1 – Arvore k*dimensional* - 3D



Fonte: GPL

2.1.1 Construção da Árvore

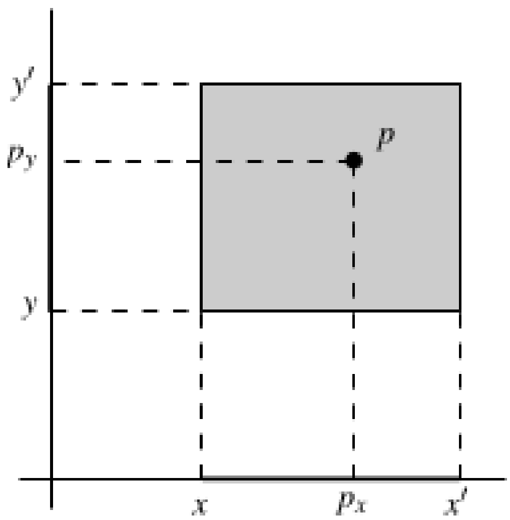
A termos didáticos segue a construção da arvore KD de 2 dimensões. Seja P o conjunto de n pontos em um plano.

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

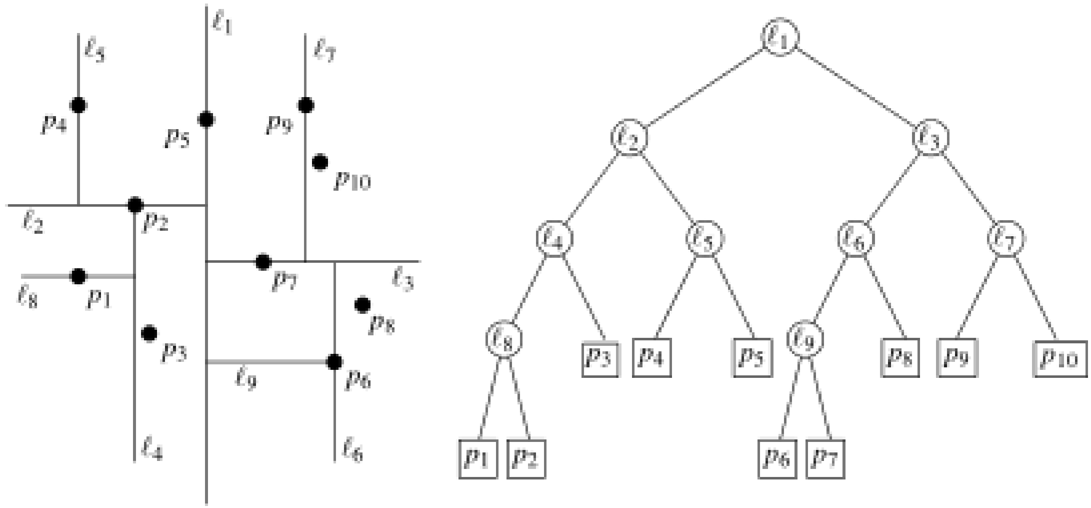


Fonte: Computational Geometry

Na construção de uma arvore 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 arvore, 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

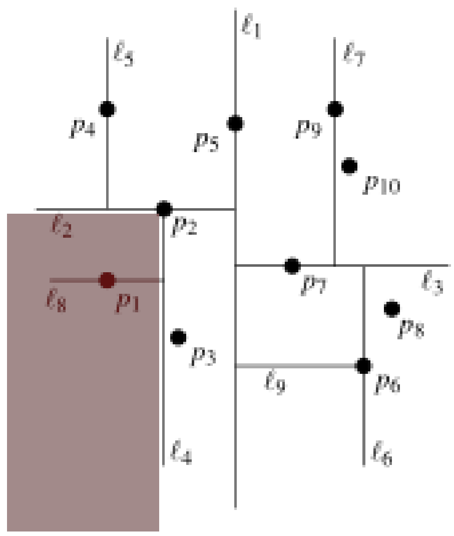


2.1.2 Busca com alcance

Agora retomamos para o algoritmo de busca. Podemos imaginar que os pontos na subárvore à esquerda da raiz, estão limitados à direita pela reta com o eixo x com valor de $x \leq l_1$. Enquanto os pontos na subárvore à direita do nodo l_1 estão limitados com o eixo $x > l_1$.

A exemplo: o nodo l_4, a região correspondente de l_4 é limitada à esquerda de l_1 e abaixo de y do nodo l_2.

Figura 4 – Área respectiva de um nodo



Denotaremos esta área de um nodo v como $regiao(v)$. A região da raiz é sim-

Algorithm 1 The Bellman-Kalaba algorithm

```
procedure BellmanKalaba(G, u, l, p)
    for all v ∈ V(G) do
        l(v) ← ∞
    end for
    l(u) ← 0
    repeat
        for i ← 1, n do
            min ← l(v_i)
            for j ← 1, n do
                if min > e(v_i, v_j) + l(v_j) then
                    min ← e(v_i, v_j) + l(v_j)
                    p(i) ← v_j
                end if
            end for
            l′(i) ← min
        end for
        changed ← l ≠ l′
        l ← l′
    until ¬changed
end procedure

procedure FindPathBK(v, u, p)
    if v = u then
        Write v
    else
        w ← v
        while w ≠ u do
            Write w
            w ← p(w)
        end while
    end if
end procedure
```

plesmente (no caso de uma arvore 2D) o plano inteiro. Portanto o algoritmo buscará a subárvore de v somente se o retângulo de busca intersectar a $regiao(v)$. O algoritmo de busca funciona descendo a arvore mas visitando somente os nodos que a $regiao(v)$ intersecta o retângulo da busca. Quando uma $regiao(v)$ esta contido no retângulo de busca retornamos todos os pontos na subárvore. Quando chegarmos nos nodos folhas temos de checar se o nodo esta dentro da busca, se tiver, retorna-o.

Segue o algoritmo que recebe como parâmetros a raiz da arvore-KD e o retângulo de busca R. Usa-se uma chamada $RetornaSubarvore(v)$ que atravessa a arvore de nodo v e retorna todos os pontos nas suas folhas. Segue como notação $fe(v)$ sendo o filho da esquerda e $fd(v)$ o filho da direita do nodo v.

A principal comparação realizada é checar se a área de $Busca$ intersecta a região de um nodo v. Para isso precisamos computar $regiao(v)$ para todos os nodos v durante a fase de construção da arvore. Uma alternativa é manter a região salva nas chamadas recursivas usando as linhas guardadas nos nodos internos. Por exemplo a região correspondente ao filho esquerda de um nodo v em uma profundidade par (no exemplo 2D, analisamos no eixo x) pode ser calculado com:

$$regiao(fe(v)) = regiao(v) \cap l(v)^{esquerda}$$

, onde $l(v)$ é a linha que divide o eixo salvo em v, e $l(v)^{esquerda}$ é a metade esquerda do plano.

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.

[2-3]

Parte III

Resultados

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.

Conclusão

[31-33]

[title=REFERÊNCIAS]

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.