

Protótipo de um ambiente virtual tridimensional para utilização no cálculo de impedimento de jogadores de futebol

Diogo Cristofolini

Orientador: Paulo C. Rodacki Gomes

Blumenau, dezembro 2004.

Roteiro da apresentação

- Introdução
- Objetivos
- Fundamentação teórica
- Especificação do protótipo
- Implementação
- Resultados e discussões
- Conclusões

Introdução

- nem sempre os auxiliares e os árbitros de futebol acertam na determinação da posição de impedimento de um atacante
- através de um processo chamado calibração de câmeras, pode-se encontrar a câmera que gerou a imagem, e gerar um modelo matemático da imagem;
- com o modelo gerado, pode-se determinar a distância entre dois pontos no campo, assim determinando ou não a posição de impedimento;

Objetivos

- aperfeiçoar o método de calibração de câmeras implementado por Starosky (2003);
- criar um ambiente virtual tridimensional para reconstruir a cena, na qual o usuário possa interagir visualizando-a de qualquer ângulo.

Fundamentação Teórica

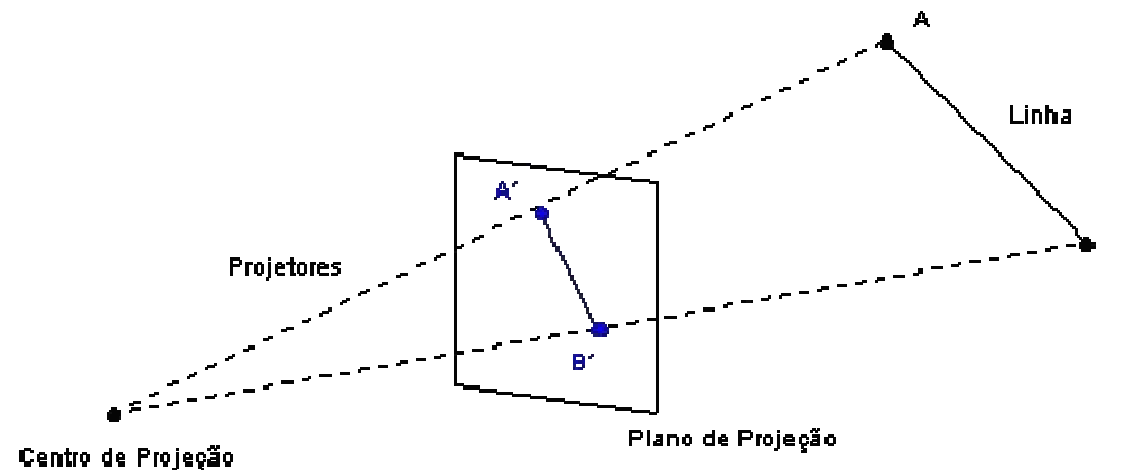
- Sistema de coordenadas 3D
 - Formado por 3 eixos mutuamente ortogonais, denominados de X, Y, Z.
 - Podem ser orientados de duas regras, “mão direita” e “mão esquerda”.



Fundamentação Teórica

● Projeção 3D

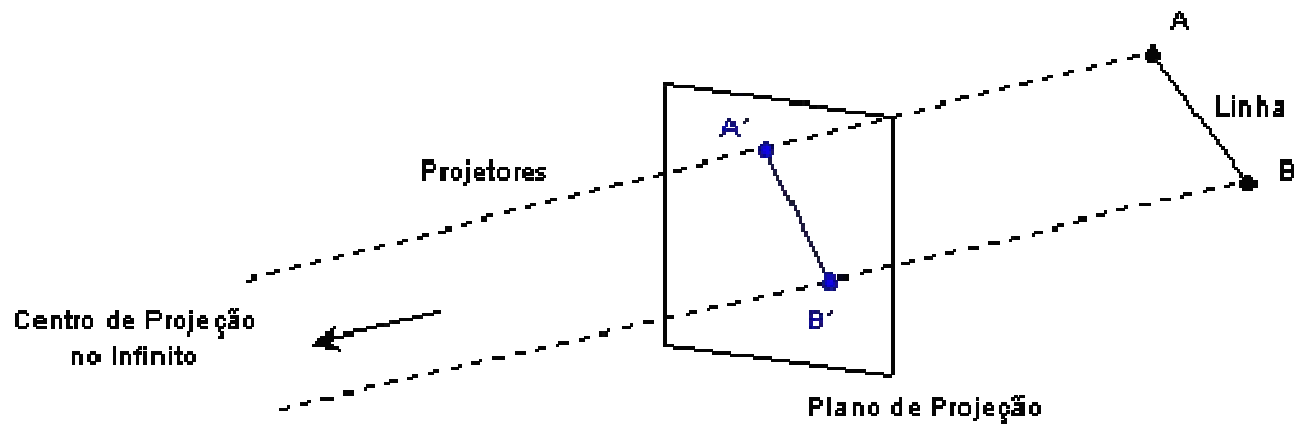
- técnica para projetar uma imagem tridimensional em um plano 2D;
- é definida por raios retos de projeção que são emanados de um centro de projeção que intercepta o plano de projeção 2D, formando nele a imagem;
- divididos em dois grupos, projeção em perspectiva e projeção paralela. Neste trabalho é utilizada a projeção em perspectiva.



Projeção em perspectiva

Fundamentação Teórica

- Projeção 3D



Projeção paralela

Fundamentação Teórica

- Mapeamento de ponto 3D para um ponto 2D

$$(\tilde{u}, \tilde{v}) = \left(f\left(\frac{X}{Z}\right), f\left(\frac{Y}{Z}\right) \right)$$

Fundamentação Teórica

- Calibração de câmeras (homografia)

- método para encontrar os parâmetros que definem a câmera que gerou a imagem;
- conhecendo os parâmetros da câmera, pode-se obter as coordenadas de pontos de imagem em um espaço de 3 dimensões equivalente ao da cena original em 2D.

$$\begin{bmatrix} uS \\ vS \\ S \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Transformação projetiva planar (homografia)

Fundamentação Teórica

- Homografia baseada em pontos próprios
 - encontrar o vetor H que satisfazem as seguintes equações;

$$u_k = \frac{h_{11}x_k + h_{12}y_k + h_{13}w_k}{h_{31}x_k + h_{32}y_k + h_{33}w_k}$$

Equação geral da coordenada u

$$v_k = \frac{h_{21}x_k + h_{22}y_k + h_{23}w_k}{h_{31}x_k + h_{32}y_k + h_{33}w_k}$$

Equação geral da coordenada v

Fundamentação Teórica

- Homografia baseada em pontos próprios

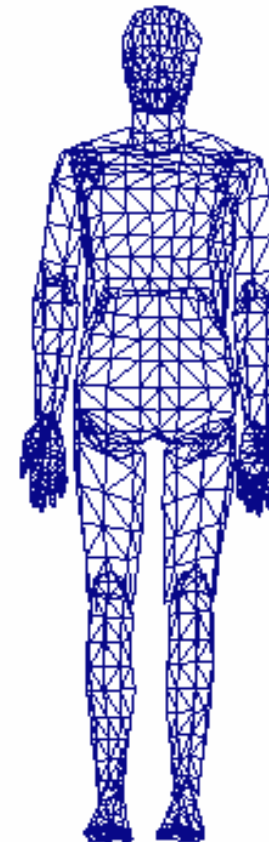
$$M = \begin{bmatrix} x_1 & y_1 & w_1 & 0 & 0 & 0 & -u_1x_1 & -u_1y_1 & -u_1 \\ x_2 & y_2 & w_2 & 0 & 0 & 0 & -u_2x_2 & -u_2y_2 & -u_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & w_n & 0 & 0 & 0 & -u_nx_n & -u_ny_n & -u_n \\ 0 & 0 & 0 & x_1 & y_1 & w_1 & -v_1x_1 & -v_1y_1 & -v_1 \\ 0 & 0 & 0 & x_2 & y_2 & w_2 & -v_2x_2 & -v_2y_2 & -v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & x_n & y_n & w_n & -v_nx_n & -v_ny_n & -v_n \end{bmatrix}$$

$$t^T = [h_{11} \quad h_{12} \quad h_{13} \quad h_{21} \quad h_{22} \quad h_{23} \quad h_{31} \quad h_{32} \quad h_{33}]$$

Fundamentação Teórica

- Objetos 3D

- Forma mais utilizada: especificar a geometria do objeto, seguido pelo mapeamento das faces e aplicação das texturas.
- Modelo de arame (wireframe).



Fundamentação Teórica

- Game Engines

- ganhou popularidade com o desenvolvimento de jogos em primeira pessoa, como Quake (Id Software);
- contém a tecnologia do núcleo do jogo, exceto os componentes do jogo propriamente dito;
- Inclui as regras de funcionamento, módulo de inteligência artificial, detecção de colisão, movimentação de câmeras, etc;
- A partir do mesmo game engine, pode-se criar vários novos jogos, com novos modelos, cenários, sons;
- pode-se fazer uma analogia com o motor de um automóvel;

Fundamentação Teórica

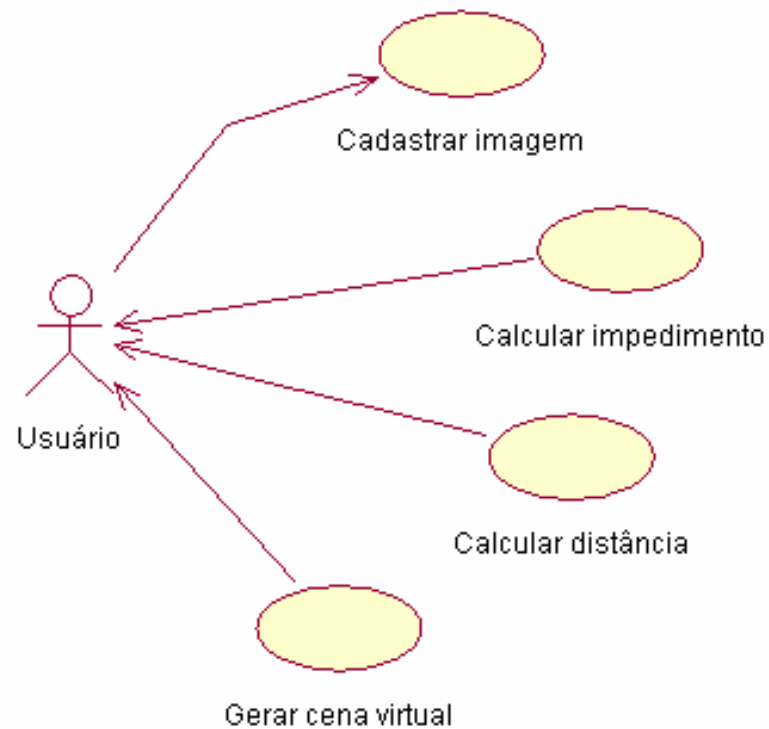
- Game Engines – Irrlicht Engine
 - Código aberto;
 - Multiplataforma;
 - Todos os objetos da cena são tratados como *Nodes*
 - Três modos de renderização:
 - OpenGL
 - Direct3D
 - Algoritmo próprio

Fundamentação Teórica

- Trabalhos correlatos
 - Starosky, Maiko (2003)
 - Szenberg, Flávio (2001)
 - Koser, Everton (2003)

Especificação do Protótipo

- Diagrama de casos de uso



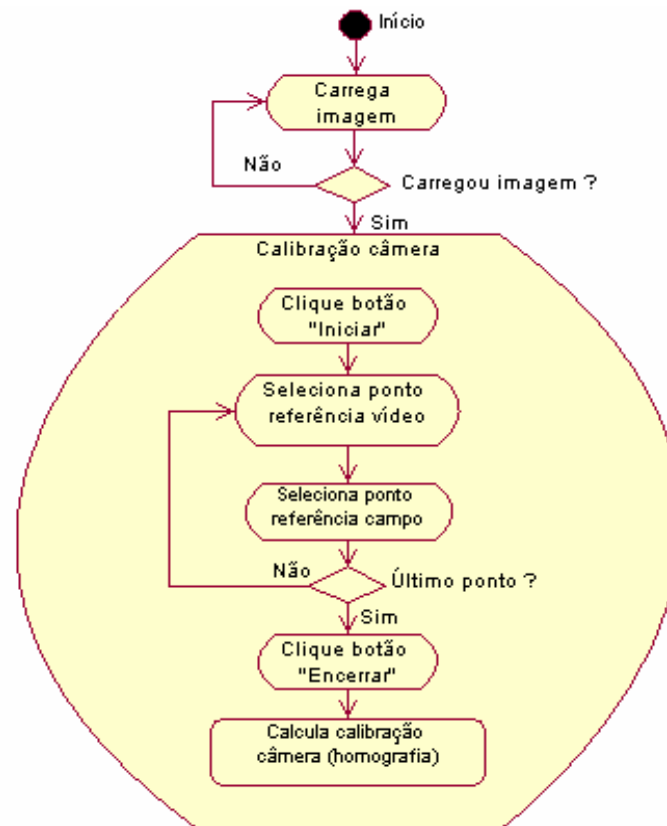
Especificação do Protótipo

- Requisitos do sistema

- Calibrar a câmera a partir de uma imagem estática de uma partida de futebol;
- Retornar a distância real entre dois pontos quaisquer do campo indicados na imagem estática;
- Gerar uma cena virtual tridimensional;
- Utilizar uma biblioteca livre para desenvolvimento de jogos.

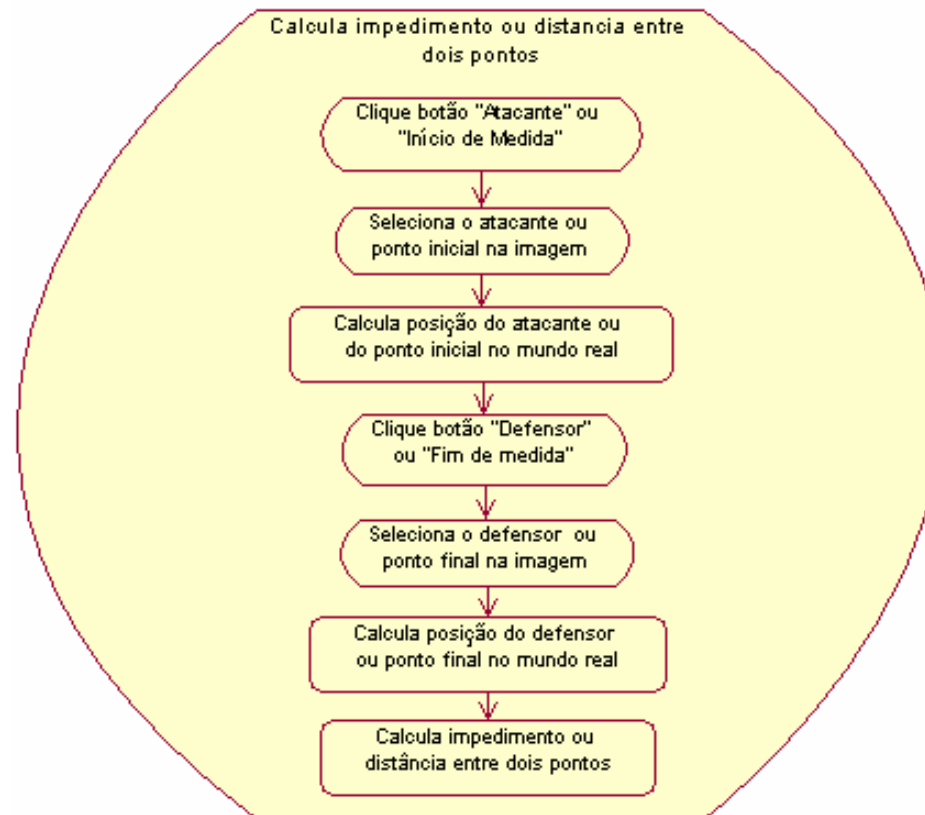
Especificação do Protótipo

- Diagrama de atividades



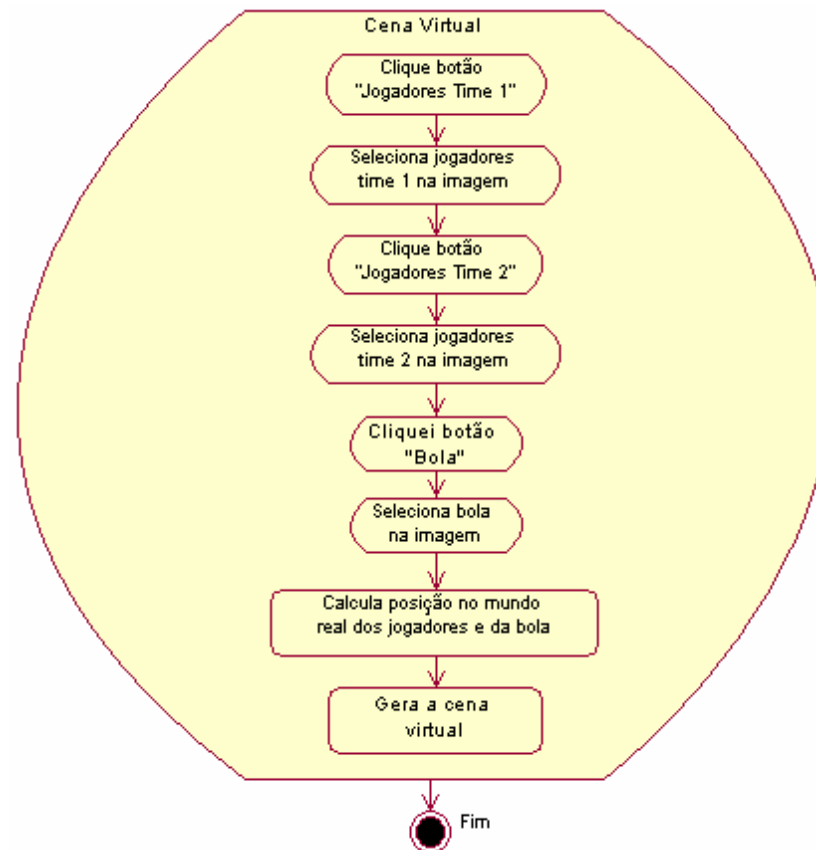
Especificação do Protótipo

- Diagrama de atividades



Especificação do Protótipo

- Diagrama de atividades



Implementação

- Ambiente Microsoft Visual C++ 6.0
- Bibliotecas utilizadas
 - IUP
 - IM
 - CD
 - Irrlicht
- 3D Studio MAX (R3)

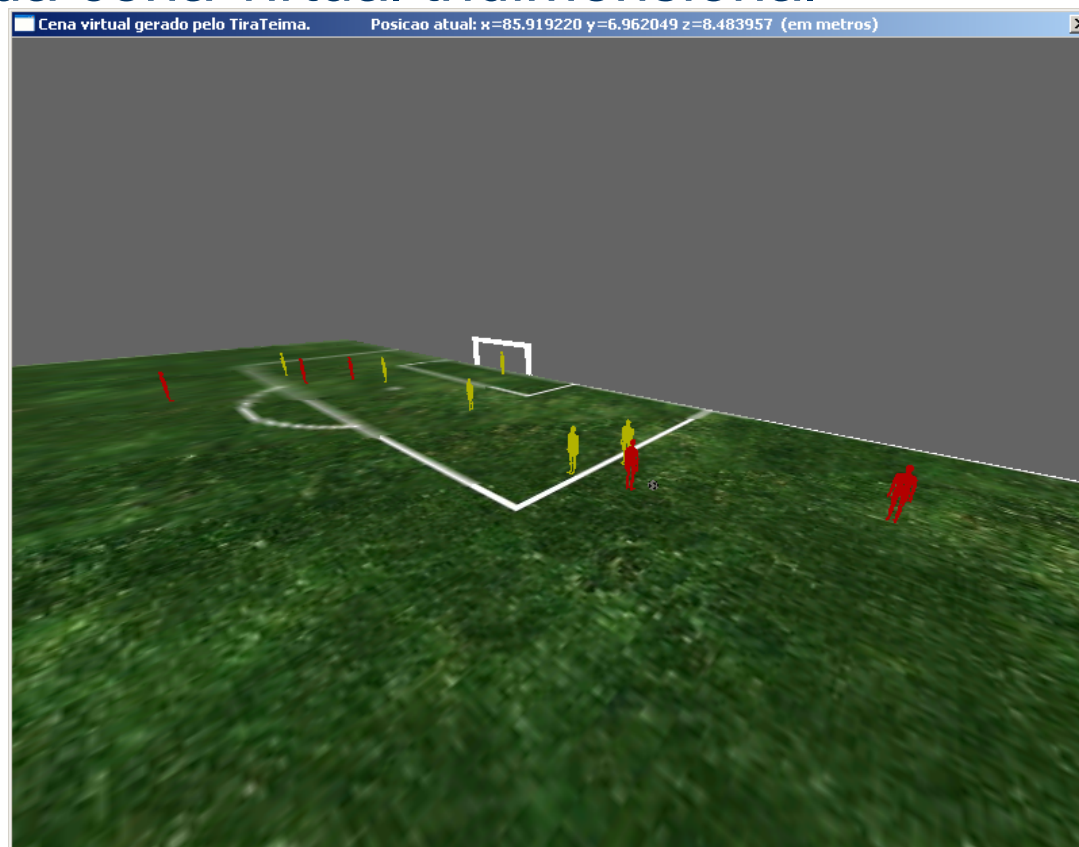
Implementação

- Tela principal do protótipo



Implementação

- Tela da cena virtual tridimensional



Resultados e discussão

Comparativo de resultados

Local comparado	Tira teima original	Tira teima aperfeiçoado	Distância correta
Pênalti	12,12 m	11,11 m	11 m
Grande área	17,18 m	16,17 m	16,5 m
Pequena área	6,6 m	5,54 m	5,5 m

Considerações Finais

- Extensões

- Implementar a segunda etapa do processo de calibração de câmeras definida por Szenberg;
- Aperfeiçoar o ambiente virtual tridimensional;
- Reconhecimento automático das linhas do campo de futebol;
- Configuração do tamanho do campo de futebol.