INSERÇÃO DE PUBLICIDADE VIRTUAL EM IMAGENS ESTÁTICAS DE JOGOS DE FUTEBOL

Diogo Zanella
diogoz@gmail.com

Orientador – Paulo Cesar Rodacki Gomes
rodacki@inf.furb.br

ROTEIRO

- Introdução
- Objetivos
- Fundamentação Teórica
  - Conceitos básicos
  - Contexto atual do tema
- Desenvolvimento do Trabalho
  - Requisitos principais
  - Especificação
    - Técnicas e ferramentas
    - Apresentação da especificação
    - Operacionalidade
  - Resultados e Discussão
- Colclusão
  - Extensões

INTRODUÇÃO

- Publicidade Virtual
  - Vantagens
  - Recursos
    - Hardware
    - Software (Visão computacional)

Fonte: Gomes (1999)

Inserção de Publicidade Virtual

INTRODUÇÃO

- Trabalhos Existentes
  - Hagen (2005)
    - Calibração automática
  - Cristofolini (2004)
    - Homografia

OBJETIVOS

- Objetivo
  - Localizar pontos do campo
  - Encontrar matriz de transformação
  - Selecionar local de inserção
  - Inserir publicidade

FUNDAMENTAÇÃO

- Publicidade/Inserção Virtual
  - Estática
  - Animada

  - Bidimensinal
  - Tridimensional

Fonte: Gomes (1999)

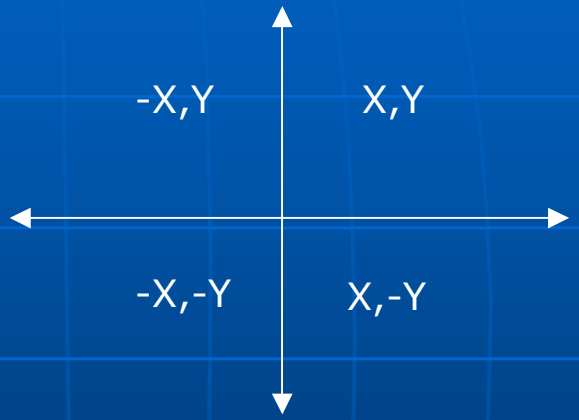
Estática/Bidimensinal Animada/Tridimensional

FUNDAMENTAÇÃO

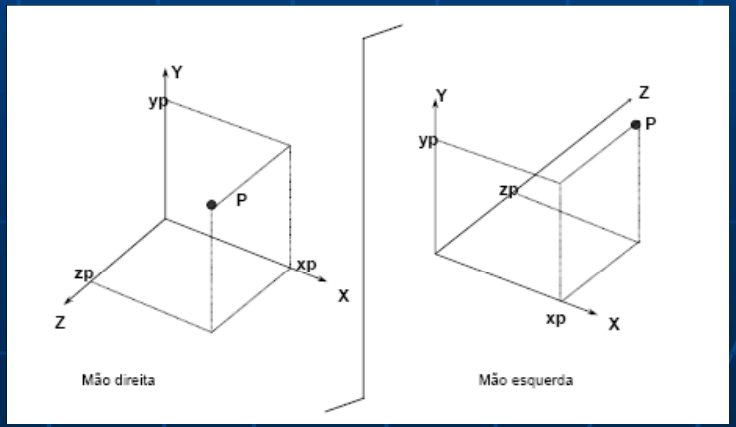
- Calibração de Câmeras
  - Sistemas de coordenadas/Parâmetros
  - Associação de pontos
  - Cena real x Cena virtual
  - Imagem estática x Vídeo
  - Algoritmo

FUNDAMENTAÇÃO

- Sistemas de coordenadas
  - Sistema 2D

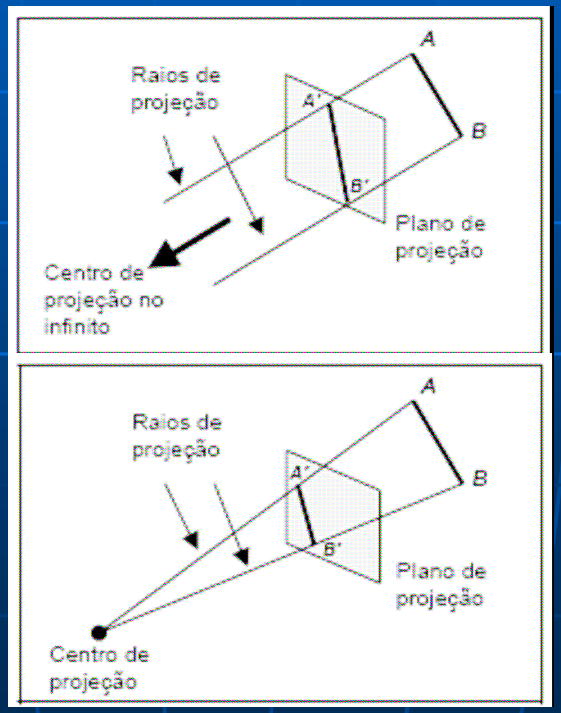


  - Sistema 3D



FUNDAMENTAÇÃO

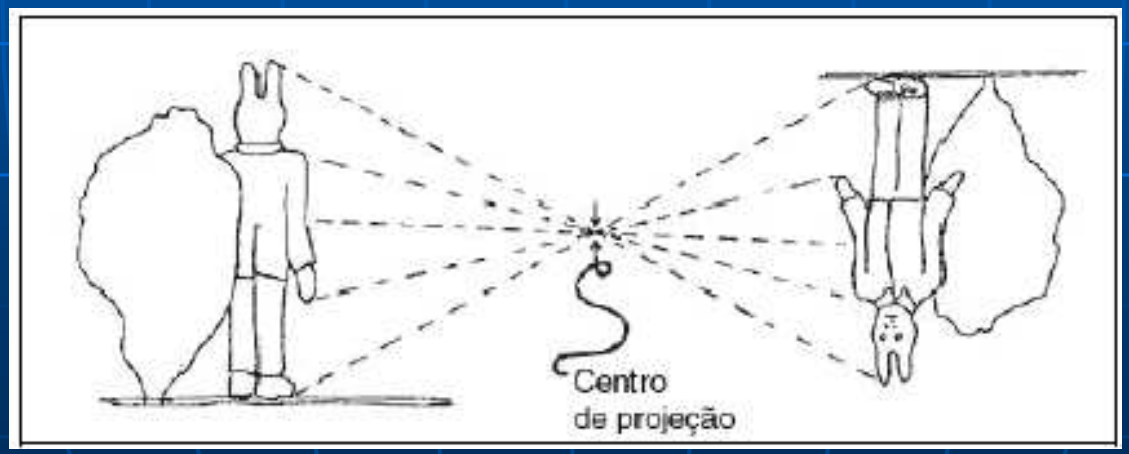
- Projeção
  - Paralela
  - Perspectiva



Fonte: Wangenheim (2004)

FUNDAMENTAÇÃO

- Câmeras de Vídeo
  - *Pinhole*



Fonte: Szemberg (2001)

FUNDAMENTAÇÃO

- Homografia

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

$$\begin{bmatrix} us \\ vs \\ s \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Fonte: Szemberg (2001)

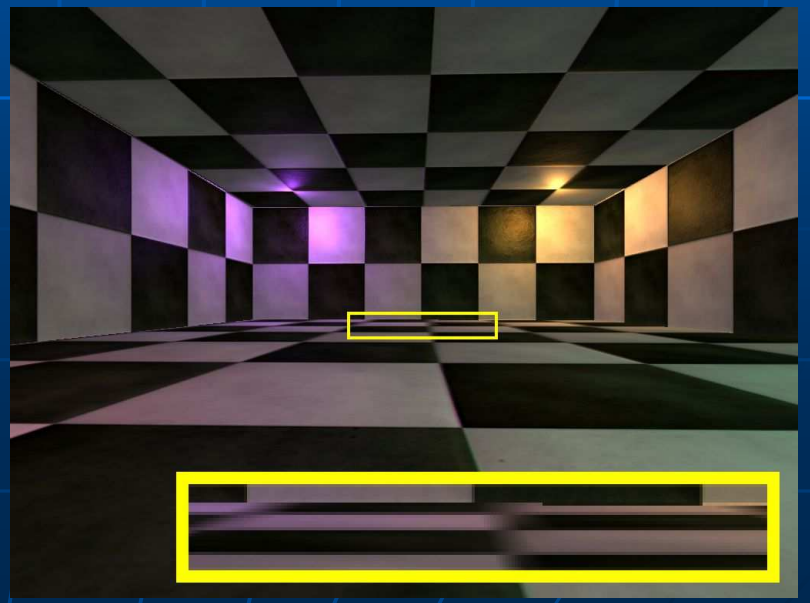
- u,v = coordenadas do ponto na tela
- x, y, z = coordenadas do pondo no mundo real
- matriz H = parâmetros de transformação

FUNDAMENTAÇÃO

- Mapeamento de Textura
  - O que é?
  - *Projective textures*
  - Mip-mapping

Ruído

Fonte: Fórum Unidev



FUNDAMENTAÇÃO

- Trabalhos Correlatos
  - Cristofolini (2004)
    - Starosky (2003)
    - Szemberg (2001)

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

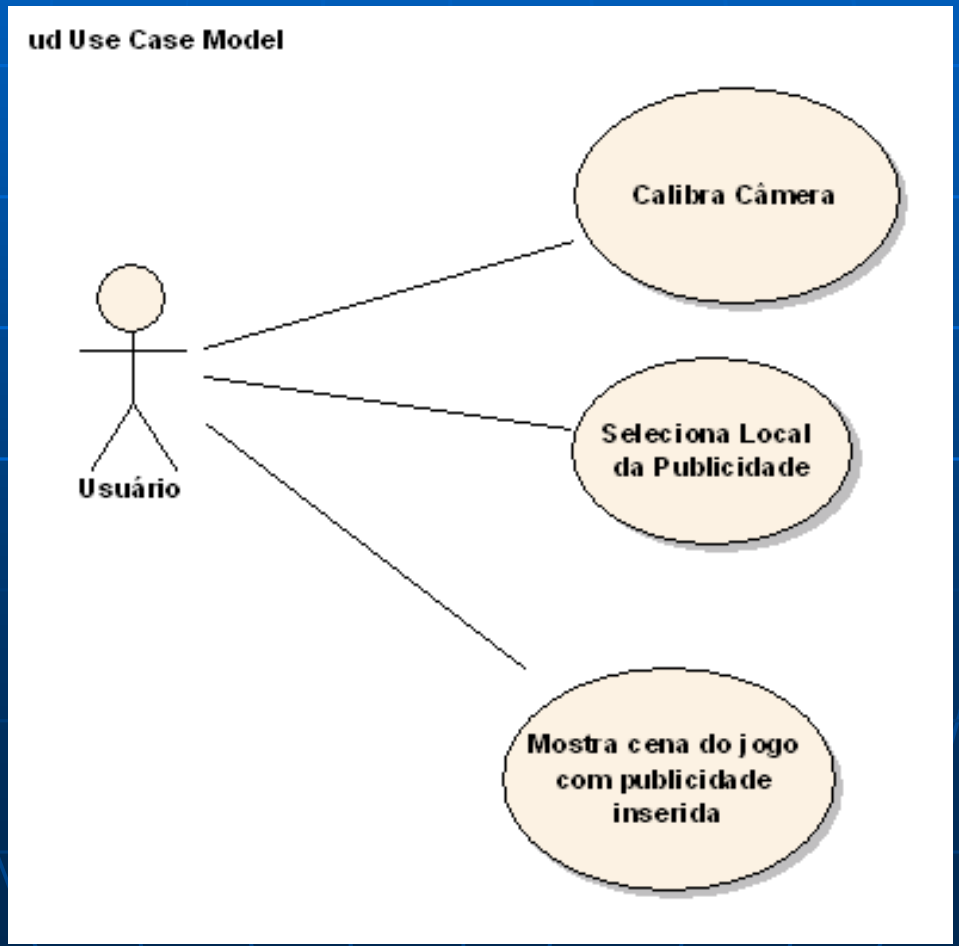
$$v_k = \frac{h_{21}x_k + h_{22}y_k + h_{23}w_k}{h_{31}x_k + h_{32}y_k + h_{33}w_k}$$

DESENVOLVIMENTO

- Requisitos do Problema
  - Funcionais
    - Entrada de imagem
      - Pontos de referência
    - Localização da publicidade
    - Inserção da publicidade
      - Outra imagem
  - Não Funcionais
    - C++
      - Tecgraf
        - IM
        - IUP
        - CD
      - OpenGL

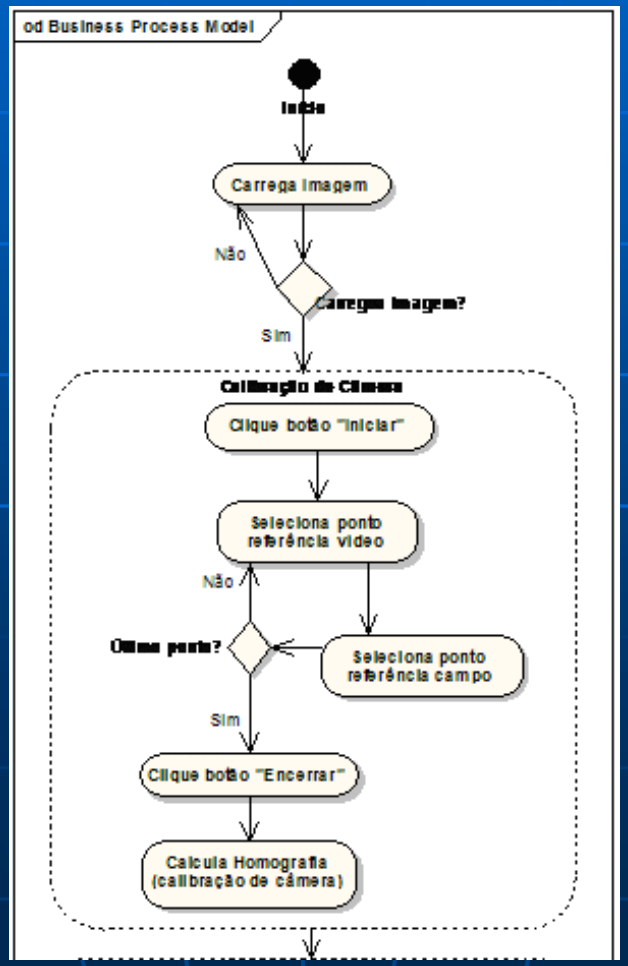
ESPECIFICAÇÃO

- Diagrama de Casos de Uso (Use Case)



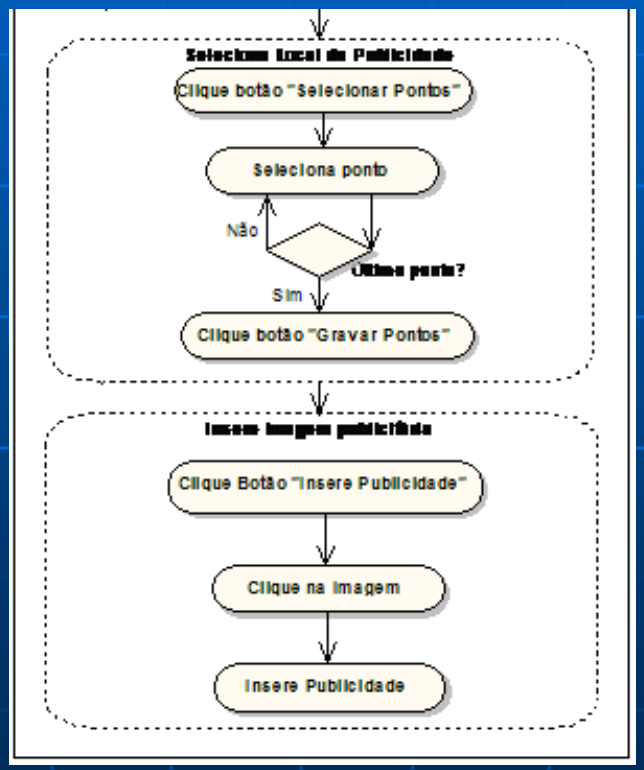
ESPECIFICAÇÃO

- Diagrama de Atividades



ESPECIFICAÇÃO

- Diagrama de Atividades



ESPECIFICAÇÃO

- Técnicas e Ferramentas
  - Implementação
    - Interface
      - Cristofolini (2004)
        - IUP
    - Manipulação/Apresentação gráfica
      - IM
      - CD x OpenGL

ESPECIFICAÇÃO

- Técnicas e Ferramentas
  - Implementação
    - Rotinas
      - Cristofolini (2004)
        - Calibração de câmera (Homografia)
        - Ponto na tela → Coordenada mundo real
      - Ponto mundo real → Coordenada na tela

```
Ti[1] = (Cam[1][1]*Pos.x) + (Cam[1][2]*Pos.y) + (Cam[1][3]*1);
Ti[2] = (Cam[2][1]*Pos.x) + (Cam[2][2]*Pos.y) + (Cam[2][3]*1);
Ti[3] = (Cam[3][1]*Pos.x) + (Cam[3][2]*Pos.y) + (Cam[3][3]*1);

u = (int)(Ti[1]/Ti[3]);
v = (int)(Ti[2]/Ti[3]);
```

ESPECIFICAÇÃO

- Técnicas e Ferramentas
  - Implementação
    - Mapeamento de Textura
      - Canvas CD → Canvas OpenGL
      - Inserção publicidade

```
//inserção da textura
glClear(GL_DEPTH_BUFFER_BIT);
glBegin(GL_POLYGON);
Calcula_Posicao_Canvas(u, v, Ponto1);
glTexCoord2f(0.0, 0.0); glVertex2f(u, 480-v);
Calcula_Posicao_Canvas(u, v, Ponto2);
glTexCoord2f(0.0, 1.0); glVertex2f(u, 480-v);
Calcula_Posicao_Canvas(u, v, Ponto3);
glTexCoord2f(1.0, 1.0); glVertex2f(u, 480-v);
Calcula_Posicao_Canvas(u, v, Ponto4);
glTexCoord2f(1.0, 0.0); glVertex2f(u, 480-v);
glEnd();
glFlush();
```

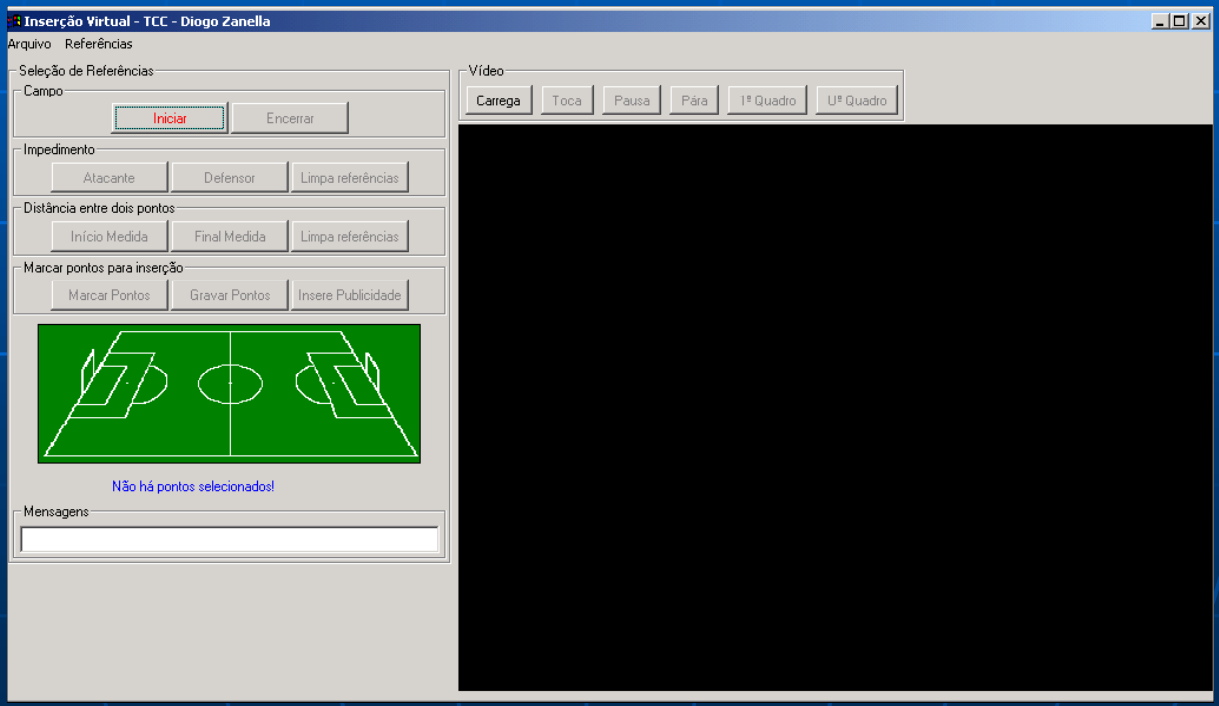
OPARACIONALIDADE

- Estudo de Caso

"Tendo uma imagem estática de jogo de futebol, calcular a homografia da câmera para, com isso, poder inserir publicidade virtual em qualquer local da imagem, levando em conta os parâmetros matemáticos da câmera e perspectiva da cena. "

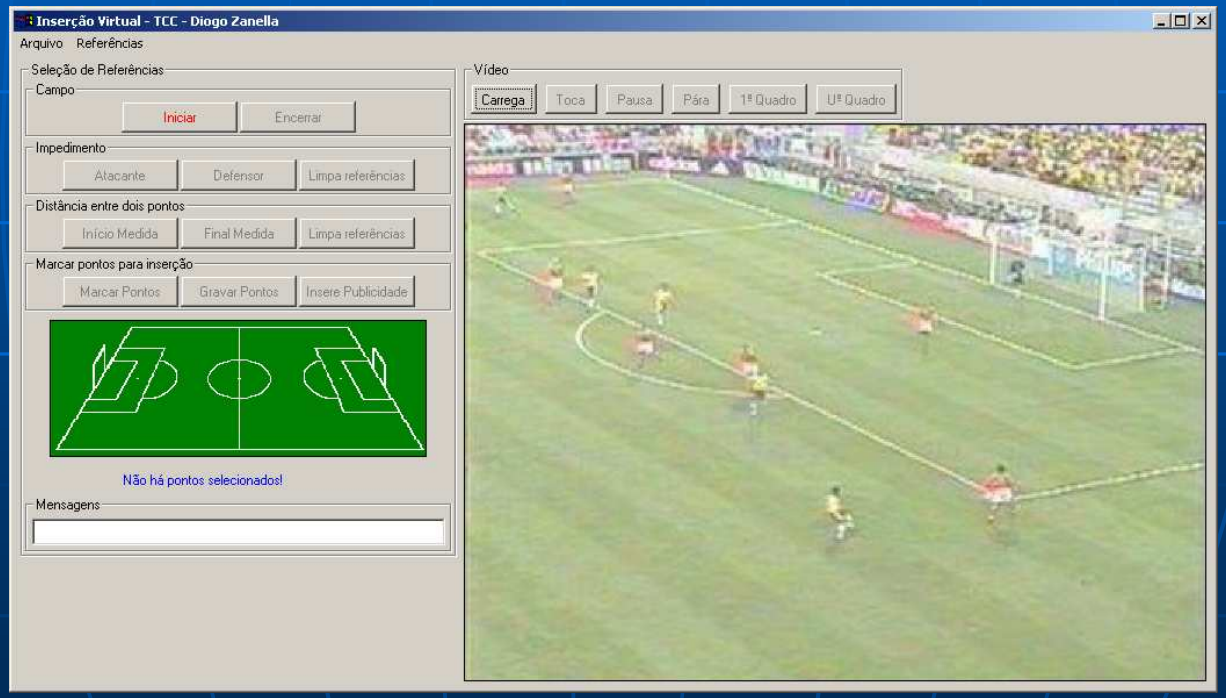
OPERACIONALIDADE

Tela Inicial



OPERACIONALIDADE

Carga da Imagem

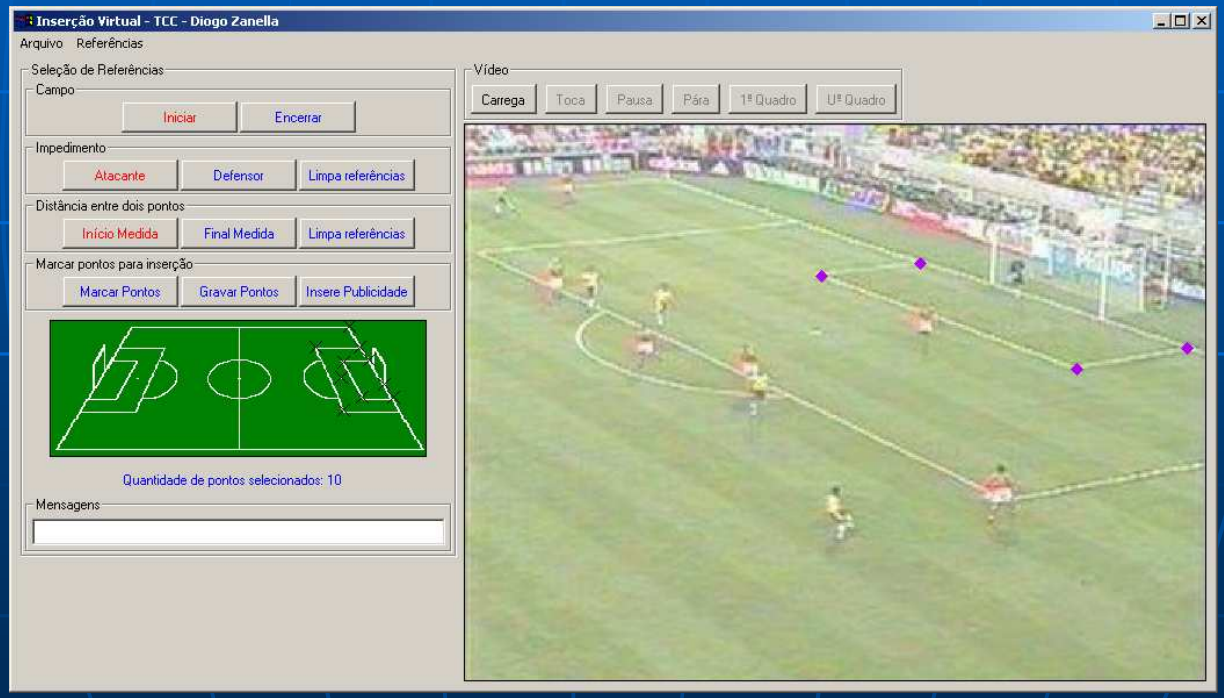


OPERACIONALIDADE

Seleção dos Pontos de Referência

OPERACIONALIDADE

Localização para Inserção da Publicidade

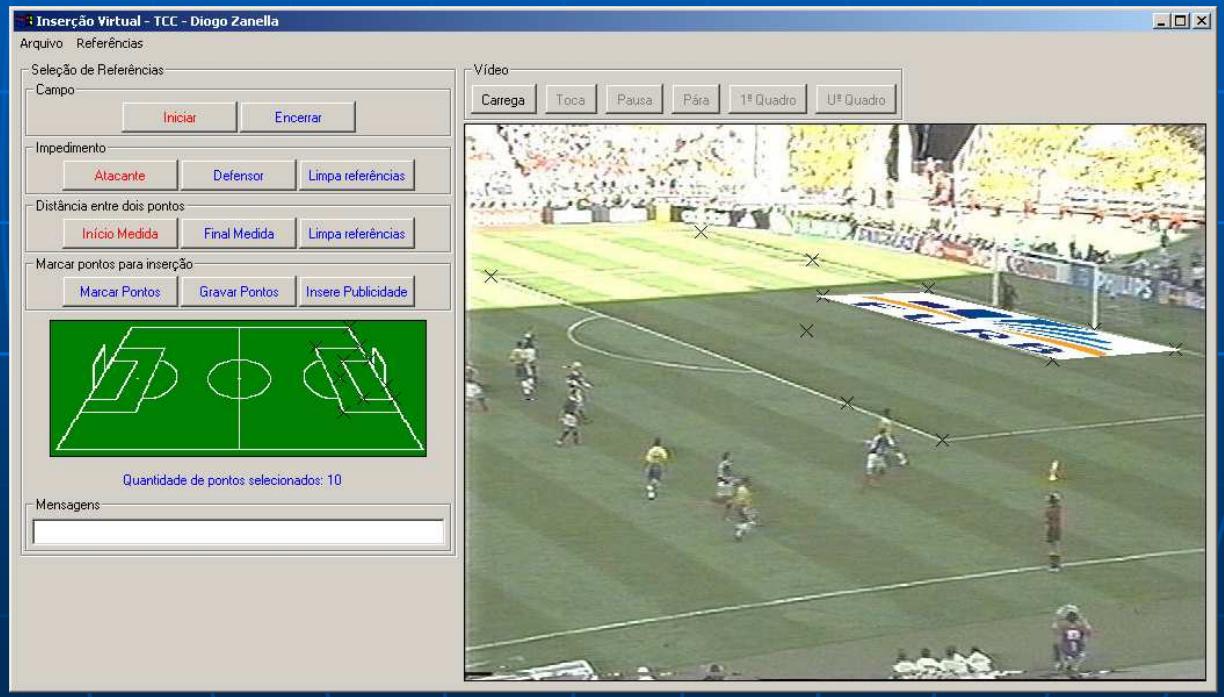


OPERACIONALIDADE

Resultado Final

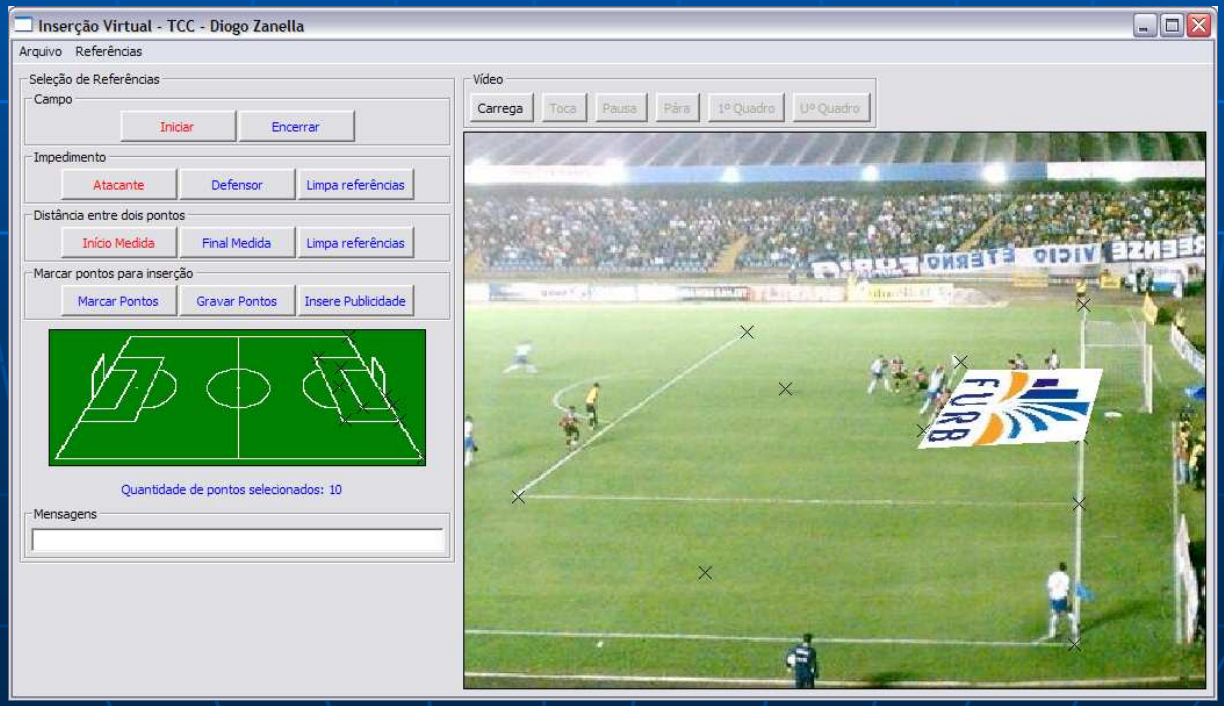
OPERACIONALIDADE

Inserção em Outra Imagem



OPARACIONALIDADE

- Resultados e Discussão
  - Dificuldades
  - Falta de precisão



CONCLUSÃO

- Idéia inicial
  - O que mudou? / Por que mudou?
- Conversão
- Dificuldade
- Ajuda
- Limitação
- Continuidade

EXTENSÕES

- Inserção de publicidade virtual em vídeos (outros tipos de campos)
- Filtro
- Melhoria na precisão