

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

INSERÇÃO DE PUBLICIDADE VIRTUAL EM IMAGENS
ESTÁTICAS DE JOGOS DE FUTEBOL

DIOGO ZANELLA

BLUMENAU
2006

2006/01-06

DIOGO ZANELLA

INSERÇÃO DE PUBLICIDADE VIRTUAL EM IMAGENS

ESTÁTICAS DE JOGOS DE FUTEBOL

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Ciências da Computação — Bacharelado.

Prof. Paulo Cesar Rodacki Gomes, Dr - Orientador

BLUMENAU
2006

2006/01-06

INSERÇÃO DE PUBLICIDADE VIRTUAL EM IMAGENS

ESTÁTICAS DE JOGOS DE FUTEBOL

Por

DIOGO ZANELLA

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Prof. Paulo Cesar Rodacki Gomes – Orientador, FURB

Membro: _____
Prof. Mauro Mattos – FURB

Membro: _____
Prof. Antonio Carlos Tavares – FURB

Blumenau, 01 de junho de 2006

Dedico este trabalho a todas as pessoas que, de qualquer forma me ajudaram ou pelo menos não me atrapalharam, mesmo quando precisavam tanto de mim, principalmente à minha família, que tornou o sonho de ser graduado uma realidade.

AGRADECIMENTOS

À Deus, pela oportunidade obtida.

À minha família, pela cobrança, incentivo e financiamento do curso.

À minha namorada Maribel, que sempre me apoiou e não me abandonou, mesmo com a distância imposta pela necessidade de terminar este trabalho.

Aos meus amigos, pela compreensão em alguns momentos de ausência minha.

A Antônio Scuri, do TeCGraf/PUC-Rio pelo auxílio com as dúvidas sobre a biblioteca

IM.

Ao meu orientador, Paulo César Rodacki Gomes, pelo imenso apoio e dedicação.

“Algo só é impossível até que alguém duvide e resolva provar o contrário.”

Albert Einstein

RESUMO

Este trabalho apresenta um estudo sobre inserção de publicidade virtual em imagens estáticas de jogos de futebol. Trata-se da continuação de uma série de trabalhos nesta linha na FURB, onde o objetivo final da pesquisa é promover a inserção virtual em vídeos de futebol. O trabalho atual parte do cálculo de homografia, implementado em um trabalho anterior, e implementa a transformação de projeção de câmera *pin-hole* para calcular as coordenadas de tela da inserção. Depois disso, são utilizados os recursos de mapeamento de texturas do OpenGL para desenhar a inserção dentro da cena. Os resultados são validados em um protótipo de aplicação que implementa o processo descrito.

Palavras-chave: Computação Gráfica. Publicidade virtual.

ABSTRACT

This work deals with virtual insertion of publicity images inside static images of soccer games. Thus it is the continuation of a series of previous work on the subject in FURB, where the main research goal is to achieve virtual insertion into video sequences of soccer matches. The present work initiates with camera homography, implemented in a previous work, and implements the pin-hole camera projective transformation in order to calculate the screen coordinates for the virtual insertion image. The inserted image is drawn with texture mapping features of OpenGL library. The results are validated in a software prototype which implements the described process.

Key-words: Computer graphics. Virtual insertion.

LISTA DE ILUSTRAÇÕES

Figura 1 - Imagem sem publicidade virtual.....	12
Figura 2 - Imagem com publicidade virtual	12
Figura 3 – Public. virtual bidimensional estática.....	15
Figura 4 – Public. virtual tridimensional animada	15
Figura 5 - Sistema de Coordenadas 2D	18
Figura 6 – Sistema de Coordenadas 3D	19
Figura 7 – Projeção paralela.....	20
Figura 8 – Projeção em perspectiva.....	20
Figura 9 – Modelo de projeção da câmera <i>pinhole</i>	20
Figura 10 – Modelo matemático da câmera <i>pinhole</i>	21
Figura 11 – Mapeamento de pontos 3D para pontos 2D	22
Figura 12 – Homografia	22
Figura 13 – Ruído	25
Figura 14 – Sistema de equações que determina os parâmetros da câmera com 4 pontos.....	26
Figura 15 – Equação da coordenada “u”	27
Figura 16 – Equação da coordenada “v”	27
Figura 17 – Diagrama de casos de uso	29
Figura 18 – Diagrama de atividades	31
Quadro 1 – Cálculo da posição na imagem de um ponto no mundo.....	33
Quadro 2 – Mapeamento de Textura	34
Figura 19 – Tela inicial do protótipo	35
Figura 20 – Carga da imagem.....	36
Figura 21 – Pontos de referência selecionados.....	37
Figura 22 – Localização para inserção da publicidade.....	37
Figura 23 – Resultado final	38
Figura 24 – Inserção em outra imagem	39
Figura 25 – Falta de Precisão	40

LISTA DE SIGLAS

2D – Duas Dimensões

3D – Três Dimensões

CD – *Canvas Draw*

FURB – Universidade Regional de Blumenau

IM – *Imaging Toolkit*

IUP – *Portable User Interface*

UML – *Unified Modeling Language*

SUMÁRIO

1 INTRODUÇÃO.....	11
1.1 OBJETIVOS DO TRABALHO	13
1.2 ESTRUTURA DO TRABALHO	13
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 PUBLICIDADE / INSERÇÃO VIRTUAL.....	15
2.2 CALIBRAÇÃO DE CÂMERAS	16
2.2.1 Algoritmo para calibração de câmera	17
2.3 SISTEMAS DE COORDENADAS	17
2.3.1 Sistemas de coordenadas 2D	18
2.3.2 Sistemas de coordenadas 3D	18
2.4 PROJEÇÃO.....	19
2.5 CÂMERAS DE VÍDEO	20
2.5.1 Aberrações de imagem.....	21
2.5.2 Mapeamento de um ponto 3D para um ponto 2D	22
2.6 MAPEAMENTO DE TEXTURA.....	23
2.6.1 Mip-Mapping	24
2.7 TRABALHOS CORRELATOS.....	25
3 DESENVOLVIMENTO DO PROTÓTIPO.....	28
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	28
3.2 ESPECIFICAÇÃO	29
3.2.1 Diagramas de casos de uso.....	29
3.2.2 Diagrama de Atividades	30
3.3 IMPLEMENTAÇÃO	32
3.3.1 Técnicas e ferramentas utilizadas.....	32
3.3.2 Operacionalidade da implementação	34
3.4 RESULTADOS E DISCUSSÃO	39
4 CONCLUSÕES.....	41
4.1 EXTENSÕES	42
REFERÊNCIAS BIBLIOGRÁFICAS	43

1 INTRODUÇÃO

A publicidade virtual vem ganhando cada vez mais espaço entre as formas de propaganda, e não se restringe somente à internet. Hoje é difícil permanecer uma hora em frente à televisão e não presenciar nenhuma forma desse tipo de apelo publicitário, principalmente se a programação for esportiva.

Essa proliferação acontece porque é muito mais simples e barato “desenhar” uma logomarca na imagem do gramado de um campo de futebol, por exemplo, e projetá-la na televisão do que fazer isso no campo propriamente dito.

Gomes (1999, p. 5) diz que “Publicidade Virtual consiste basicamente na inserção eletrônica de anúncios publicitários dentro da imagem de vídeo. A idéia é que esta inserção seja feita de tal maneira que o tele-espectador tenha a impressão de que os anúncios fazem parte da cena realmente.”

Publicidade virtual, portanto, consiste em inserir imagens, logomarcas, *banners* e outras formas de divulgação, chamada ou apelo publicitário, incluindo objetos tridimensionais e até animados em uma imagem de televisão, de forma que, quem assista, tenha a sensação de que aquela imagem está mesmo lá.

Para realizar tal tarefa, os recursos mais utilizados são os de hardware ou de visão computacional. No primeiro caso as câmeras já conhecem previamente os pontos principais da imagem que está sendo filmada e, através desses pontos a imagem publicitária é inserida na imagem capturada, em tempo real. Os métodos baseados em visão computacional utilizam mais recursos de software e consistem em capturar a imagem da câmera, identificar os pontos principais através de algoritmos de reconhecimento de padrões e, com esses pontos, transformar a perspectiva da imagem publicitária e inseri-la de modo que ela se “encaixe” na imagem filmada, para produzir um efeito como mostrado nas figuras 1 e 2.



Fonte: Gomes (1999).

Figura 1 - Imagem sem publicidade virtual



Fonte: Gomes (1999).

Figura 2 - Imagem com publicidade virtual

Para a utilização da visão computacional, existem alguns trabalhos que tratam da identificação e localização dos pontos principais dessas imagens e, através das coordenadas desses pontos, encontram a transformação de perspectiva da cena. Esta técnica é chamada de calibração de câmeras. O problema em aberto é aplicar essa transformação sobre a imagem de publicidade, fazendo com que ela fique com a mesma distorção da cena e, após isso, inseri-la no local desejado da imagem.

Existem outros Trabalhos de Conclusão de Curso de alunos da FURB, que trabalham nessa área, porém em duas linhas diferentes. Hagen (2005), por exemplo, possui toda parte de calibração automática de câmeras e, através dessa calibração, faz o reconhecimento de linhas do campo para vídeos de futebol, porém sem a parte de inserção de publicidade no vídeo, que era também um de seus objetivos. Já Cristofolini (2004) faz o sentido inverso. Através de pontos escolhidos pelo usuário, faz a calibração da câmera e o processo de homografia de imagens estáticas de jogos de futebol.

A idéia é unir processos existentes nestes trabalhos e, com base nesses, criar a distorção na publicidade conforme a perspectiva da imagem, definir sua posição e inseri-la de modo a parecer o mais real possível em imagens estáticas.

1.1 OBJETIVOS DO TRABALHO

O objetivo do trabalho é inserir imagens de publicidade em cenas estáticas de vídeos de jogos de futebol, usando os recursos de visão computacional.

Os objetivos específicos do trabalho são:

- a) localizar os pontos principais da imagem de um campo de futebol utilizando recursos disponíveis em outros trabalhos (calibração de câmeras, localização de pontos), como o de Koser (2003), Starosky (2003), Cristofolini (2004) e Hagen (2005);
- b) encontrar a matriz de transformação para distorcer a publicidade de acordo com a perspectiva do campo;
- c) selecionar na imagem estática do campo, o local em que a publicidade será inserida;
- d) inserir na cena a publicidade transformada e mostrar o resultado final.

1.2 ESTRUTURA DO TRABALHO

O capítulo 2 desse trabalho apresenta toda a fundamentação teórica utilizada para o desenvolvimento desse trabalho e também do protótipo, tendo informações sobre conceitos e teorias importantes para o entendimento do mesmo, bem como trabalhos que serviram de base para o presente trabalho.

No capítulo 3 são dados detalhes sobre os processos do desenvolvimento do protótipo, como definição dos requisitos, especificação (representada pelo diagrama de casos de uso e o de atividades), descrição de técnicas e ferramentas utilizadas, bem como a operacionalidade do protótipo, exemplificando com um estudo de caso os resultados do protótipo.

As conclusões do desenvolvimento do trabalho, bem como do protótipo são apresentados no capítulo 4, juntamente com algumas propostas para continuidade desse trabalho em trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais assuntos e conceitos pertinentes ao entendimento deste trabalho, como mapeamento de textura, sistema de coordenada, bem como trabalhos correlatos.

2.1 PUBLICIDADE / INSERÇÃO VIRTUAL

Gomes (1999) diz que publicidade virtual é um método publicitário que utiliza a tecnologia que as emissoras de televisão possuem para inserir imagens publicitárias no local escolhido pelo anunciante, e consiste basicamente na inserção eletrônica de anúncios de publicidade dentro de imagens de vídeo, feita essa de tal maneira que o tele-espectador tenha a impressão de que aquele anúncio esteja lá realmente.

Além deste método, que utiliza a aparelhagem de televisão (hardware) para a inserção de publicidade nas cenas, esse processo pode ser feito também por software (visão computacional), como já citado anteriormente.

Atualmente existem algumas empresas que desenvolvem este trabalho, como a Orad (2004), porém, a maioria delas, utiliza recursos de hardware para tal tarefa.

Esse tipo de publicidade pode ser tanto bidimensional como tridimensional, estáticas ou animadas, como pode ser visualizado nas figuras 3 e 4.



Fonte: Gomes (1999, p.7).

Figura 3 – Public. virtual bidimensional estática



Fonte: Gomes (1999, p.7).

Figura 4 – Public. virtual tridimensional animada

2.2 CALIBRAÇÃO DE CÂMERAS

Calibração de câmeras é, segundo Koser (2003), uma maneira de transformar um sistema de coordenadas lógico no sistema de coordenadas da câmera de vídeo. Em outras palavras, calibração de câmera é a maneira que temos de, obtendo os parâmetros matemáticos de uma câmera de vídeo, criar uma câmera virtual com os mesmos parâmetros. Fazendo isso, é possível tratar uma imagem de câmera como uma cena real, sabendo onde, no mundo real, se localizaria um ponto qualquer da imagem.

Para poder inserir uma imagem publicitária com realismo em uma imagem de vídeo, é preciso fazer a calibração de câmera.

Como são conhecidas as medidas de um campo de futebol de verdade, é possível associar pontos de uma imagem do campo com pontos do campo real. Cantos da área e do campo, marca do pênalti e centro do campo são alguns exemplos de pontos conhecidos.

Associando esses pontos é possível conhecer os parâmetros matemáticos da câmera que deu origem à cena e calibrar a câmera virtual, para que tenha os mesmos parâmetros e para que se possa fazer a transição de imagens do mundo real para o virtual e vice-versa.

Assim é possível trazer a cena para o mundo virtual, inserir a publicidade e levá-la novamente para o real, fazendo com que a publicidade se torne muito realista e o tele-espectador tenha a sensação de que ela está mesmo lá.

No caso deste trabalho, a calibração é feita para apenas uma imagem, o que simplifica e utiliza menos processamento. Para utilizar esse recurso em um vídeo, por exemplo, seria necessário que o sistema reconhecesse automaticamente os pontos ou linhas da imagem, associasse estes aos pontos ou linhas reais de um campo e só então calibrasse a câmera. E isso tudo, quadro a quadro.

Para transmissões ao vivo este processo quadro a quadro precisa ainda ser feito em

tempo real, o que necessita de um grande poder de processamento. Por isso, neste caso, geralmente esse processo é feito via hardware em transmissões ao vivo.

Caso haja interesse em informações sobre calibração automática e reconhecimento automático de linhas de um campo de futebol pode-se consultar o trabalho de Koser (2003).

2.2.1 Algoritmo para calibração de câmera

O algoritmo aqui mostrado é o mesmo utilizado por Cristofolini (2004) e utiliza como entrada uma imagem estática de jogos de futebol e uma imagem modelo de um campo de futebol. Nessas duas imagens são informados os pontos de referência, que são predefinidos. Para uma correta calibração, é necessário que todos os pontos sejam coplanares, ou seja, todos devem estar no mesmo plano. Nesse caso o plano será o chão do campo de futebol.

A partir desses pontos é calculada a homografia, cujo resultado são os parâmetros da câmera. Para maiores detalhes sobre esse algoritmo pode ser consultado o trabalho de Starosky (2003).

2.3 SISTEMAS DE COORDENADAS

Tendo em vista que uma imagem *raster* utiliza o sistema de coordenadas 2D e a calibração de câmera virtual precisar ser feita em 3D, neste trabalho são usados estes dois sistemas de coordenadas. Por isso, nesta seção serão apresentadas as transformações utilizadas para utilização dos dois sistemas em conjunto.

2.3.1 Sistemas de coordenadas 2D

O sistema de coordenadas 2D é orientado por dois eixos, geralmente chamados de X e Y, que dividem o espaço em 4 quadrantes: X,Y; -X,Y; X,-Y e -X,-Y. Isso pode ser visualizado a seguir.

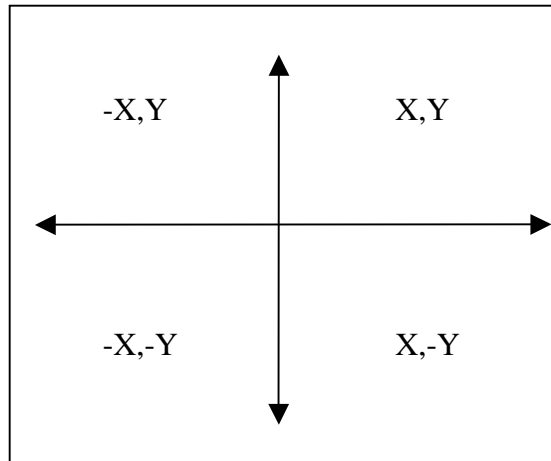


Figura 5 - Sistema de Coordenadas 2D

Como o Canvas, que é um componente existente em algumas bibliotecas como CD e OpenGL, com uma área “desenhável”, ou seja, é um componente que permite inserirmos imagens, seja carregando-as de um arquivo ou mesmo desenhando-as por código de programação através linhas, pontos, polígonos, sólidos, etc., utiliza o sistema de coordenadas 2D, com sua origem no seu canto inferior esquerdo, neste trabalho será utilizado o quadrante X,Y, pois é o sistema utilizado pelo Canvas, isso facilita a manipulação da posição das imagens.

2.3.2 Sistemas de coordenadas 3D

O sistema de coordenadas 3D é orientado por 3 eixos. Além do X e do Y, existe o eixo Z, o que possibilita a localização de pontos em três dimensões. Existindo para esse sistema duas formas de orientação. A “mão direita” e a “mão esquerda”, como mostra a figura

seguinte.

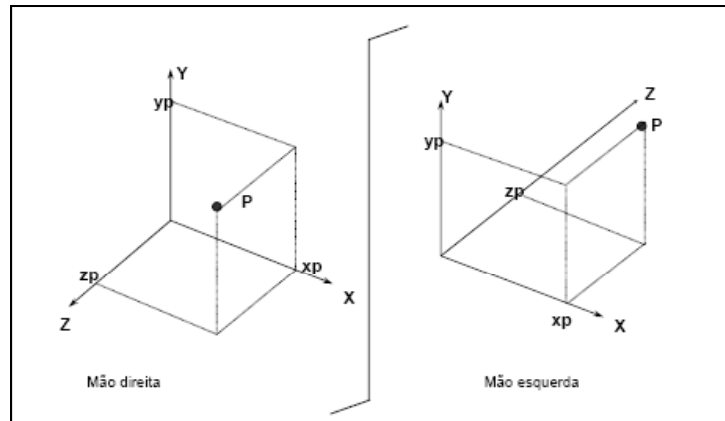


Figura 6 – Sistema de Coordenadas 3D

Neste trabalho será utilizada a regra da mão esquerda.

2.4 PROJEÇÃO

Segundo Foley (1996, p. 229), por possuir um eixo ou coordenada a menos, a visualização em 2D é mais simples que em 3D. Para solucionar o problema de visualização de uma cena 3D em um dispositivo 2D, como uma TV ou tela de computador, é usada a técnica de projeção. Portanto, segundo HILL (2001), projeção é a técnica de transformar e exibir uma cena definida em um mundo tridimensional em uma superfície bidimensional.

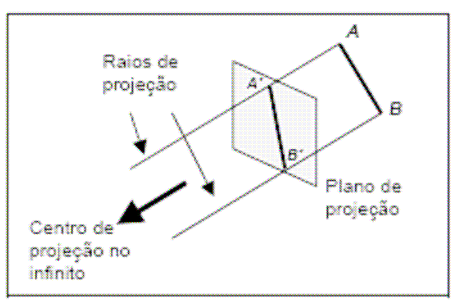
Segundo Cristofolini (2004), os tipos de projeção são classificados em projeção paralela e projeção em perspectiva e a diferença entre os dois tipos está na relação entre o ponto de projeção e o plano de projeção.

Na projeção em perspectiva os projetores formam ângulos oblíquos entre si, portanto a distância entre o ponto de projeção e o plano de projeção é finita e, quanto mais longe o objeto a ser projetado estiver do centro de projeção, menor ele vai ser projetado no plano. Isso não acontece na projeção paralela, já que os projetores formam, entre si, retas paralelas, o que torna a distância entre o ponto de projeção e o plano de projeção infinita.

A projeção paralela é utilizada quando é necessário preservar a escala dos objetos

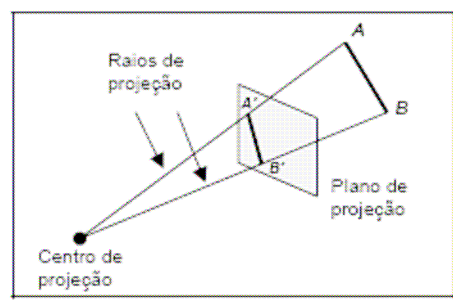
projetados ou quando se precisa fazer cálculos utilizando os ângulos, já que esse tipo de projeção não distorce a cena.

Por possibilitar definir e diferenciar distâncias na cena, neste trabalho optou-se pelo uso da projeção em perspectiva.



Fonte: Wangenheim (2004).

Figura 7 – Projeção paralela



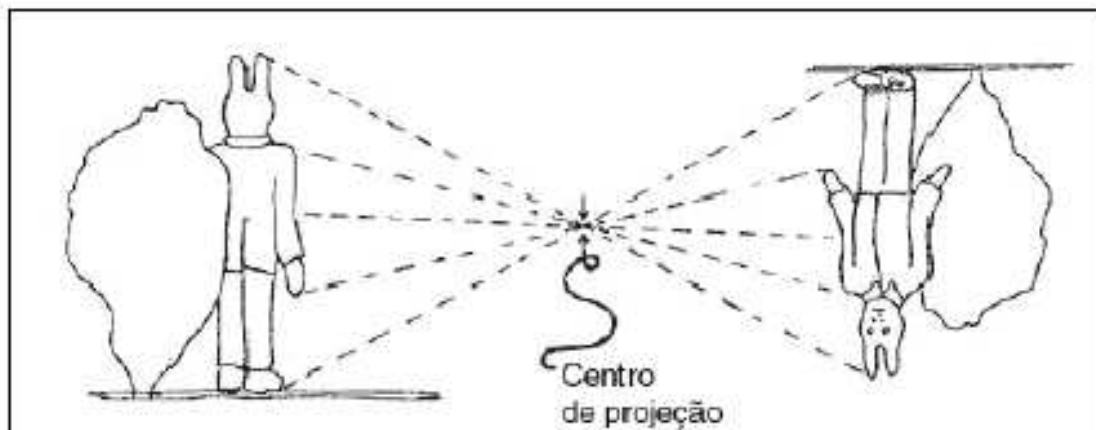
Fonte: Wangenheim (2004).

Figura 8 – Projeção em perspectiva

2.5 CÂMERAS DE VÍDEO

Por utilizar projeção em perspectiva, esse trabalho é baseado em um tipo de câmera denominada *pinhole*, que é apresentada em Szemberg (2001) e pode-se encontrar informações mais detalhadas sobre suas características em Starosky (2003).

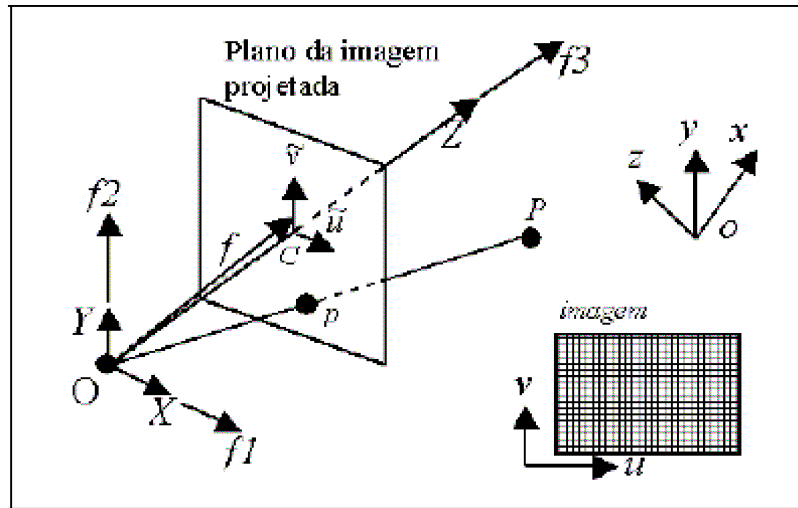
O princípio de funcionamento desse tipo de câmera pode ser visualizado na figura 9 e, segundo Cristofolini (2004), consiste em captar imagens do mundo real através de um orifício e projetá-las invertidas em um plano.



Fonte: Szemberg (2001, p. 98).

Figura 9 – Modelo de projeção da câmera *pinhole*

Sobre essa câmera, o mais importante para o entendimento do presente trabalho, é entender o modelo matemático, que pode ser visualizado na figura 10.



Fonte: Szemberg (2001, p. 98).

Figura 10 – Modelo matemático da câmera *pinhole*

O centro geográfico do plano que as imagens são capturadas é representado na figura 10 pelo ponto C . Este plano representa a área de visualização na tela ou monitor. Já o ponto O representa a posição do olho humano ou o centro de projeção da câmera, ou seja, representa o ponto de vista de quem está olhando para o plano de projeção. O símbolo f representa a distância entre os pontos O e C e interfere no zoom da imagem.

Independentemente da câmera, temos o ponto o , que representa a origem de um sistema de coordenadas lógico, que nesse trabalho representará o modelo do campo de futebol.

2.5.1 Aberrações de imagem

Segundo Koser (2003), aberrações são distorções ou deformações sofridas pelas imagens de vídeo por se afastar do centro óptico da lente, o que dificulta esse trabalho de calibração de câmera, já que linhas que na realidade são retas podem se tornar curvas na imagem capturada pela câmera.

O centro óptico, por sua vez, é definido por Koser (2004) como um ponto imaginário na lente da câmera que serve como base para determinar os demais parâmetros da câmera. Este ponto imaginário, dificilmente coincide com o centro físico da lente da câmera, mas é muito importante conhecê-lo, pois as demais informações sobre a câmera dependem dele.

2.5.2 Mapeamento de um ponto 3D para um ponto 2D

Baseando-se no modelo matemático descrito acima, segundo Cristofolini (2004, p.17), pode-se mapear um ponto tridimensional $P(X,Y,Z)$, para um bidimensional $p(u,v)$ através da seguinte equação:

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

Fonte: Szemberg (2001, p. 99).

Figura 11 – Mapeamento de pontos 3D para pontos 2D

Desenvolvendo esta equação, obtém-se a equação apresentada na figura 12. Segundo Cristofolini (2004, p. 17), a equação obtida representa a homografia para mapear pontos 3D para pontos 2D, e seu desenvolvimento pode ser encontrado em Starosky (2003).

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

Fonte: Szemberg (2001, p. 101).

Figura 12 – Homografia

Na equação, as coordenadas do ponto na tela são os elementos u e v . Os elementos x , y e z representam as coordenadas do ponto no mundo real e, neste caso o parâmetro z está sendo considerado fixo com o valor 1 . Já a matriz H , que possui os elementos de h_{11} a h_{33} , representa os parâmetros da transformação.

Homografia é função obtida através da análise dos parâmetros matemáticos de uma imagem 3D para transformá-la em 2D e vice-versa. Isso é necessário quando queremos representar uma imagem 3D em um plano como o que acontece quando filmamos uma imagem e a mostramos em uma tela. O mesmo acontece quando queremos representar uma imagem 2D em um espaço 3D. Maiores detalhes sobre homografia podem ser encontrados em Cristofolini (2004, p. 18).

2.6 MAPEAMENTO DE TEXTURA

Okamoto (2003, p.6), diz que mapeamento de textura é um método de criar uma complexidade aparente, sem precisar modelar e renderizar cada detalhe de um objeto e é utilizada para dar mais realismo a imagens criadas por computador. Diz também que, geralmente, essa técnica mapeia uma função em uma superfície 3D, mas o domínio pode ser também uni ou bidimensional.

Para mapear uma textura, a imagem deve ser mapeada em espaço 3D (parametrização) que, por sua vez, será mapeada para a tela, geralmente utilizando a técnica de projeção.

Então, é preciso mapear a imagem para 3D e, após isso, marca-se a localização para onde a textura será mapeada. Com essa localização marcada, encontra-se o espaço correspondente na tela e só então a textura é inserida.

Azevedo e Conci (2003), dizem que, por ser muito útil e simples, as técnicas de mapeamento de textura passaram a ser padrão para as interfaces de softwares e hardwares gráficos, pois podem ser usadas tanto em cenas simples como em cenas muito complexas, sem exigir muito poder de processamento.

Se fosse possível olhar o que acontece realmente em uma técnica de mapeamento de textura poder-se-ia perceber que é apenas a modificação da cor do objeto em cada pixel, que

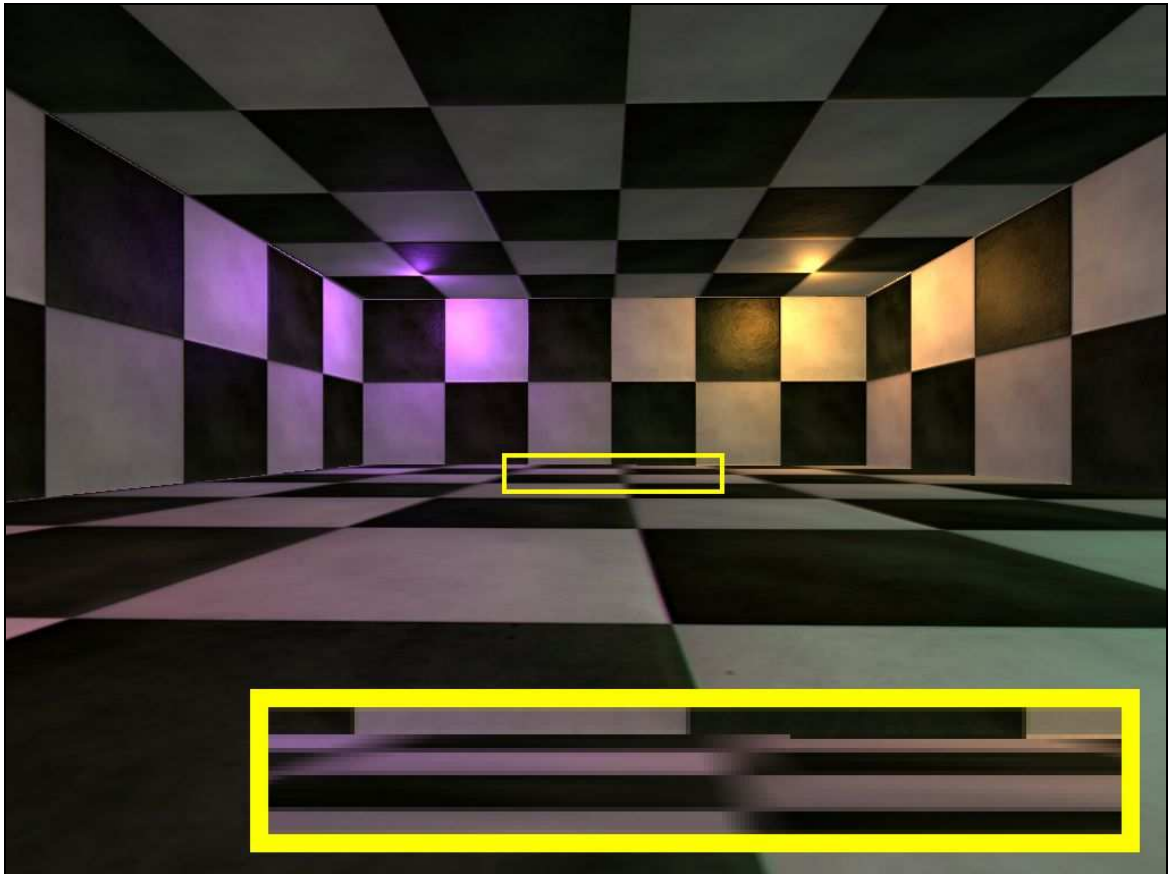
tomará a cor do texel (pixel da textura) correspondente.

Azevedo e Conci (2003) descrevem vários tipos de mapeamento de textura, frisando que não é necessário que as imagem de textura sejam bidimensionais. No caso da textura ser tridimensional, será utilizada somente uma fatia bidimensional dela para o cálculo da iluminação e sombra.

Existe também a técnica de *projective textures*, que não tem coordenadas fixas para a projeção da textura, o que possibilita fazer a textura deslizar sobre a superfície da imagem, podendo criar animações com efeito de água corrente ou refletores, por exemplo.

2.6.1 Mip-Mapping

Segundo Azevedo e Conci (2003, p. 323), a técnica de mip-mapping vem pra resolver dois problemas. Quando a imagem de textura precisa ser muito diminuída devido à perspectiva da cena, a projeção da textura tende a apresentar ruídos de fundo. Isso acontece porque as texturas dos objetos distantes ficam menores que a quantidade de pixels disponíveis. O resultado do ruído pode ser visualizado na figura 13.



Fonte: Fórum Unidev (2005).

Figura 13 – Ruído

O segundo problema apresentado por Azevedo e Conci (2003) é quando o objeto alvo é aproximado e revela detalhes da imagem de textura, como a baixa resolução.

Estes dois problemas podem ser resolvidos com mip-mapping, pois essa técnica utiliza-se de imagens de várias resoluções da mesma textura. As imagens de resolução mais baixa são usadas quando o objeto alvo está distante, pois ele não precisa de muitos pixels. À medida que o objeto é aproximado, são utilizadas imagens com resolução maior, para obter-se mais detalhes e melhor visualização.

2.7 TRABALHOS CORRELATOS

O presente trabalho utiliza procedimentos criados no trabalho de Cristofolini (2004), como o cálculo da homografia e calibração de câmera, bem como a interface, com algumas

modificações. Cristofolini (2004) por sua vez, baseou-se em outros trabalhos como o de Starosky (2003) e Szemberg (2001).

Cristofolini (2004, p. 18) diz que “Existem casos onde os pontos do mundo tridimensional que serão utilizados para a calibração da câmera são todos coplanares, isto é, estão todos no mesmo plano. Este tipo de calibração de câmera é chamado de Homografia.”

Baseando-se no trabalho de Starosky (2003), Cristofolini (2004) encontra a homografia utilizando 4 pares de pontos de referência, construindo um sistema linear de 9 equações e 9 incógnitas, mostrado na figura 14, que é resolvido através do método de Gauss-Jordan. Esses 4 pares são obtidos através da seleção dos pontos de referência feita pelo usuário.

$$\begin{bmatrix}
 x_1 & y_1 & w_1 & 0 & 0 & 0 & 0 & 0 & 0 & -u_1 & 0 & 0 & h_{11} \\
 0 & 0 & 0 & x_1 & y_1 & w_1 & 0 & 0 & 0 & -v_1 & 0 & 0 & h_{12} \\
 0 & 0 & 0 & 0 & 0 & 0 & x_1 & y_1 & w_1 & -t_1 & 0 & 0 & h_{13} \\
 x_2 & y_2 & w_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -u_2 & 0 & h_{21} \\
 0 & 0 & 0 & x_2 & y_2 & w_2 & 0 & 0 & 0 & 0 & -v_2 & 0 & h_{22} \\
 0 & 0 & 0 & 0 & 0 & 0 & x_2 & y_2 & w_2 & 0 & -t_2 & 0 & h_{23} \\
 x_3 & y_3 & w_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -u_3 & h_{31} \\
 0 & 0 & 0 & x_3 & y_3 & w_3 & 0 & 0 & 0 & 0 & 0 & -v_3 & h_{32} \\
 0 & 0 & 0 & 0 & 0 & 0 & x_3 & y_3 & w_3 & 0 & 0 & -t_3 & h_{33} \\
 x_4 & y_4 & w_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_1 \\
 0 & 0 & 0 & x_4 & y_4 & w_4 & 0 & 0 & 0 & 0 & 0 & 0 & s_2 \\
 0 & 0 & 0 & 0 & 0 & 0 & x_4 & y_4 & w_4 & 0 & 0 & 0 & s_3
 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ u_4 \\ v_4 \\ 1 \end{bmatrix}$$

Fonte: Szemberg (2001, p. 102).

Figura 14 – Sistema de equações que determina os parâmetros da câmera com 4 pontos

Através deste sistema de equações é possível encontrar a equação que determina a coordenada u e a coordenada v no cálculo da homografia, como pode ser observado nas figuras 15 e 16.

Essas coordenadas (u e v) darão a posição de cada pixel na tela, levando em conta os

parâmetros da câmera, sendo possível assim, através destas equações conhecer a perspectiva da cena, o que é de grande importância para a inserção de publicidade virtual.

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

Fonte: Szemberg (2001, p. 103).

Figura 15 – Equação da coordenada “u”

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

Fonte: Szemberg (2001, p. 103).

Figura 16 – Equação da coordenada “v”

Tendo essa homografia, Cristofolini (2004) faz a calibração da câmera virtual e, a partir dessa calibração, disponibiliza algumas funcionalidades como cálculo de impedimento, cálculo de distância entre pontos no campo e também recria a cena do jogo em 3D virtualmente.

De todas essas funcionalidades, foram utilizadas neste trabalho a parte de seleção de pontos e calibração de câmera. As funções que permitem o cálculo de distância e impedimento foram deixadas no protótipo apenas por mera ilustração, já que não faziam parte dos objetivos deste. Já as funções que permitem criar uma cena virtual 3D da imagem escolhida foram totalmente excluídas do protótipo.

3 DESENVOLVIMENTO DO PROTÓTIPO

Neste capítulo serão descritas as etapas seguidas no desenvolvimento do protótipo inicialmente proposto nesse trabalho. Serão abordadas as seguintes etapas: requisitos principais do protótipo, especificação do protótipo, implementação e, por fim, os resultados obtidos.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O sistema deve aceitar a entrada de uma imagem estática de um jogo de futebol, geralmente obtida através de transmissões de televisão e, a partir desta imagem, permitir que o usuário determine os pontos (no mínimo cinco) de referência para fazer a calibração da câmera virtual.

Terminado esse processo, o sistema deve disponibilizar ao usuário escolher o local na imagem onde será inserida a publicidade virtual, através da seleção de quatro pontos na imagem do campo de futebol.

Selecionado esses quatro pontos o sistema deve inserir a imagem de publicidade dentro desses pontos utilizando funções OpenGL, visando o maior realismo possível.

O sistema deve permitir também inserir a imagem de publicidade na mesma localização em outra imagem de um jogo de futebol, ou seja, deve guardar os pontos selecionados pelo usuário para a inserção de publicidade.

O protótipo deve ser desenvolvido na linguagem C++, auxiliada pelas bibliotecas IM, CD, IUP e *OpenGL*.

3.2 ESPECIFICAÇÃO

Para o desenvolvimento da especificação do protótipo, utilizou-se da linguagem UML, da qual se pode obter maiores detalhes e explicações em Furlan (1998), e da ferramenta Enterprise Architect 6.1, produzindo-se o diagrama de casos de uso e o diagrama de atividades.

3.2.1 Diagramas de casos de uso

Inicialmente o usuário deve calibrar a câmera, selecionando uma imagem de jogo de futebol e relacionando seus pontos com medidas de um campo real. Feito isso o usuário escolhe o local da publicidade. Como resultado, o protótipo mostrará a cena escolhida inicialmente com a imagem de publicidade no local escolhido pelo usuário.

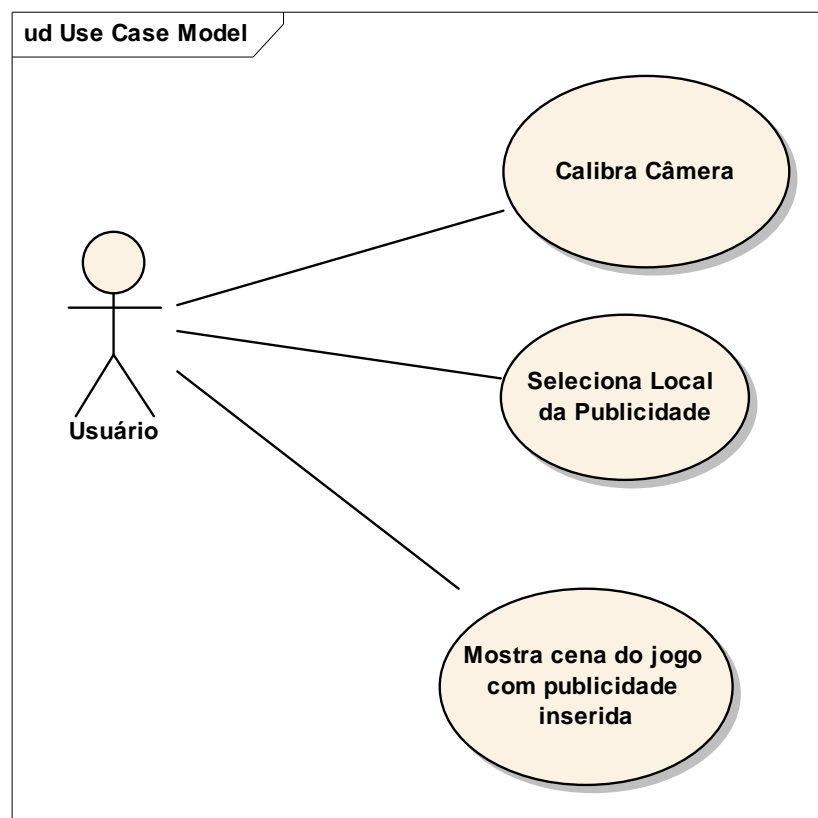


Figura 17 – Diagrama de casos de uso

3.2.2 Diagrama de Atividades

Conforme a figura 18, o primeiro passo é o usuário abrir uma imagem de jogo de futebol. Sendo feito isso sem problemas, é preciso calibrar a câmera. Para isso clica-se no botão “Iniciar” e então começa-se a escolher os pontos de referência para a calibração, sendo que primeiro deve-se escolher o ponto na imagem e depois no modelo de campo pré-definido. Depois de escolhidos os pontos de referência desejados clica-se no botão “Encerrar”, fazendo assim o cálculo da homografia.

Calibrada a câmera, o próximo passo é definir o local em que a publicidade será inserida. Para isso é necessário clicar no botão “Selecionar pontos” e, em seguida, selecionar os quatro pontos que servirão de referência para os limites da imagem de publicidade. Após selecionar os quatro pontos, clica-se em gravar pontos.

Finalmente, para inserir a publicidade no local escolhido, precisa ser dado um clique em “Inserir publicidade”, seguido de outro clique em qualquer local da imagem.

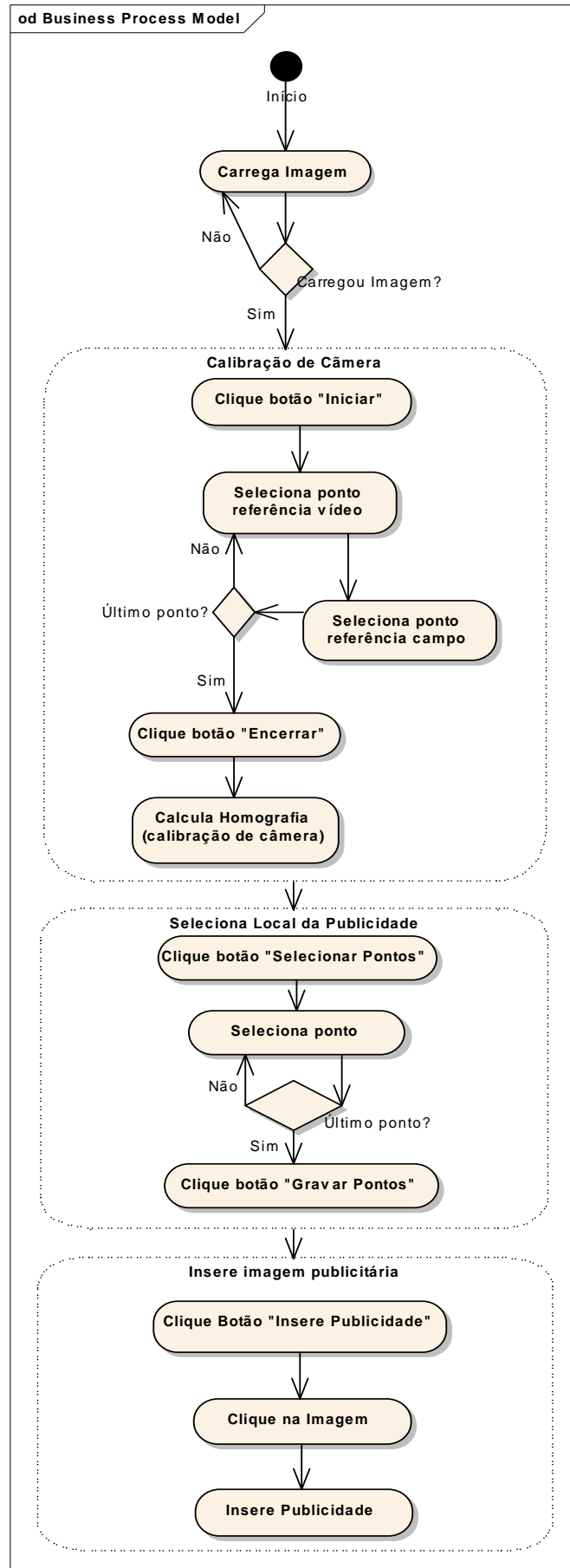


Figura 18 – Diagrama de atividades

3.3 IMPLEMENTAÇÃO

Nesta seção do trabalho são apresentadas técnicas e ferramentas utilizadas, bem como a operacionalidade da implementação do protótipo.

3.3.1 Técnicas e ferramentas utilizadas

A linguagem de programação utilizada no desenvolvimento do protótipo é C++, sendo esta auxiliada pelas bibliotecas IUP (TECGRAF, 2003), para geração de interface, CD (TECGRAF, 2003) e IM (TECGRAF, 2003), para manipulação e apresentação gráfica.

A escolha dessa linguagem deu-se pelo motivo de este ser uma extensão do trabalho de Cristofolini (2004), onde também é utilizada esta linguagem em conjunto com as bibliotecas citadas. Informações mais completas sobre as bibliotecas citadas podem ser encontradas em TecGraf (2003).

Por esta implementação ter sido feita a partir do protótipo desenvolvido por Cristofolini (2004), boa parte da interface desse protótipo foi utilizada, ignorando algumas funcionalidades apresentadas pelo mesmo, como criação de cena virtual 3D da imagem do jogo de futebol, que não são importantes para este trabalho, e adicionando outras que são pertinentes ao protótipo em desenvolvimento.

Inicialmente deu-se importância em estudar o código fonte do protótipo utilizado como base, selecionar rotinas que poderiam ser utilizadas e descartar as que não teriam utilidade para o protótipo a ser desenvolvido.

A partir das rotinas selecionadas, foi implementada inicialmente a seleção de pontos em imagens de jogos de futebol, guardando suas coordenadas do mundo real, com o intuito de inserir esses pontos nos lugares correspondentes em outra imagem de jogo de futebol.

O cálculo da posição no mundo real de um ponto selecionado na imagem já havia sido implementado inicialmente por Cristofolini (2004), porém, para inserir estes pontos pré-selecionados nos lugares correspondentes em outra imagem foi necessário fazer o inverso, ou seja, a partir de um ponto no mundo real, encontrar sua posição na imagem.

Isso pode ser feito multiplicando a matriz da câmera virtual encontrada na homografia pelas coordenadas do ponto a ser transformado, como pode ser visualizado no quadro 1:

```
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]);
```

Quadro 1 – Cálculo da posição na imagem de um ponto no mundo

Implementado isso, foram feitos testes que comprovaram uma eficiência satisfatória dessa funcionalidade.

Como o trabalho de Cristofolini (2004) utiliza um canvas CD pra abrir as imagens a serem trabalhadas e, neste trabalho, foi preciso utilizar funções OpenGL, foi necessário substituir o canvas CD por um canvas GL. Com isso, precisou-se fazer uma correção no tratamento de coordenadas do canvas, que é diferente no CD e no OpenGL.

A biblioteca CD, segundo TecGraf (2003?) é uma biblioteca de gráficos independente de plataforma e que contém funções para suportar aplicações de vetores e imagens e o seu componente canvas trabalha em um espaço bidimensional. Todos os seus dados são dos tipos padrões da linguagem C. Esta biblioteca é livre e pode ser usada em aplicações publicas ou comerciais.

Já a biblioteca OpenGL, além de suportar a maioria das funções que a biblioteca CD suporta, pode trabalhar também no espaço tridimensional e suporta mapeamento de textura. Estas características levaram a optar-se pelo uso do canvas OpenGL para o desenvolvimento deste trabalho.

Concluída essa etapa, deu-se início a fase de inserção de uma imagem de publicidade

virtual dentro desses quatro pontos selecionados. Isso foi feito através de Mapeamento de Textura, utilizando OpenGL, como pode ser visto no código mostrado no quadro 2.

```

void Insere_Publicidade()
{
    int error;
    int u;
    int v;
    glEnable(GL_DEPTH_TEST);
    glDepthFunc(GL_LEQUAL);

    //carga da imagem de publicidade
    imFile* ifile = imFileOpen("logofurb.jpg", &error);
    imImage* img = imFileLoadImage(ifile, 0, &error);
    imFileClose(ifile);

    //conversão dos dados da imagem para o padrão do OpenGL
    unsigned char* imagem = NULL; /* opengl display buffer */
    imagem = (unsigned char*)malloc(3*img->size);
    imConvertPacking(img->data[0], imagem, img->width, img->height, img->depth, img->data_type, 0);

    //inicialização dos dados da textura
    glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
    glTexImage2D(GL_TEXTURE_2D, 0, 3, img->width, img->height, 0, GL_RGB, GL_UNSIGNED_BYTE, imagem);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_DECAL);
    glEnable(GL_TEXTURE_2D);
    glShadeModel(GL_FLAT);

    //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();
}

```

Quadro 2 – Mapeamento de Textura

3.3.2 Operacionalidade da implementação

Levando em conta todas as técnicas e conceitos apresentados até o momento, foi desenvolvido um protótipo para a concretização de toda essa teoria. Para apresentar as funcionalidades do protótipo, a seguir é apresentado um 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.

A tela inicial do protótipo conta com uma área de botões de comandos, um modelo de

um campo de futebol, uma área de mensagens e uma área onde a imagem será trabalhada, como pode ser observado na figura 19.

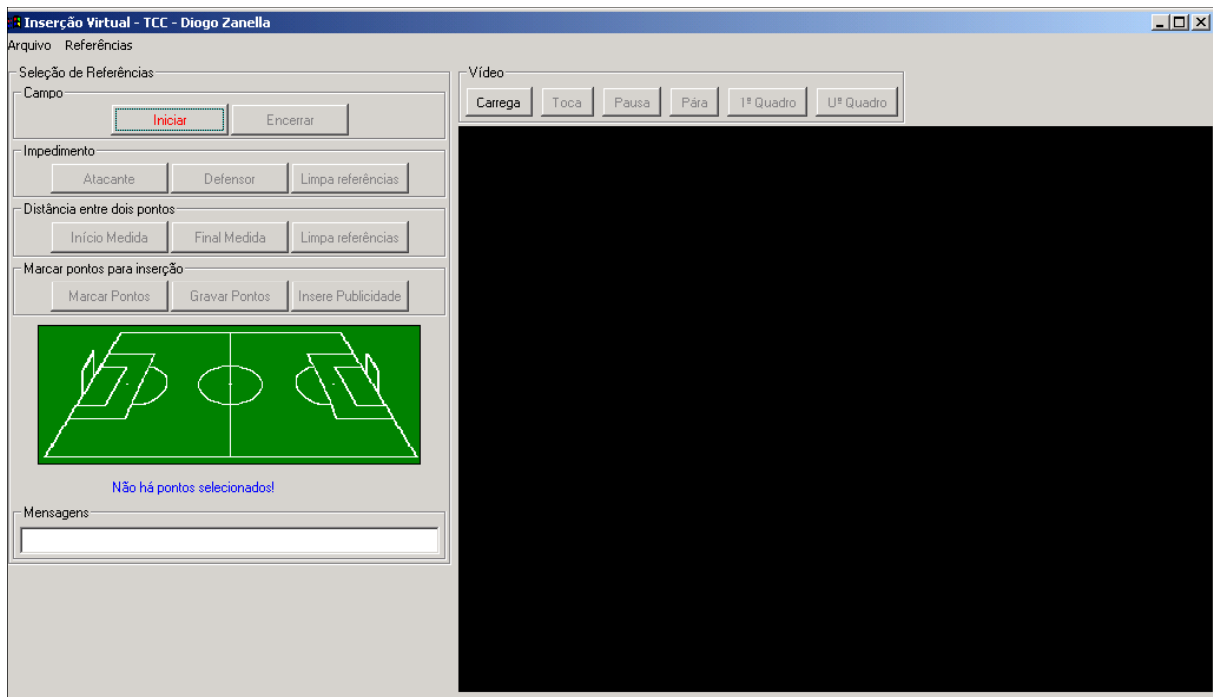


Figura 19 – Tela inicial do protótipo

Até a fase de calibração de câmera, o protótipo segue basicamente os mesmos passos do protótipo descrito por Cristofolini (2004), tendo o usuário que clicar em “Carrega” para carregar uma imagem estática de um jogo de futebol, como é demonstrado na figura 20.

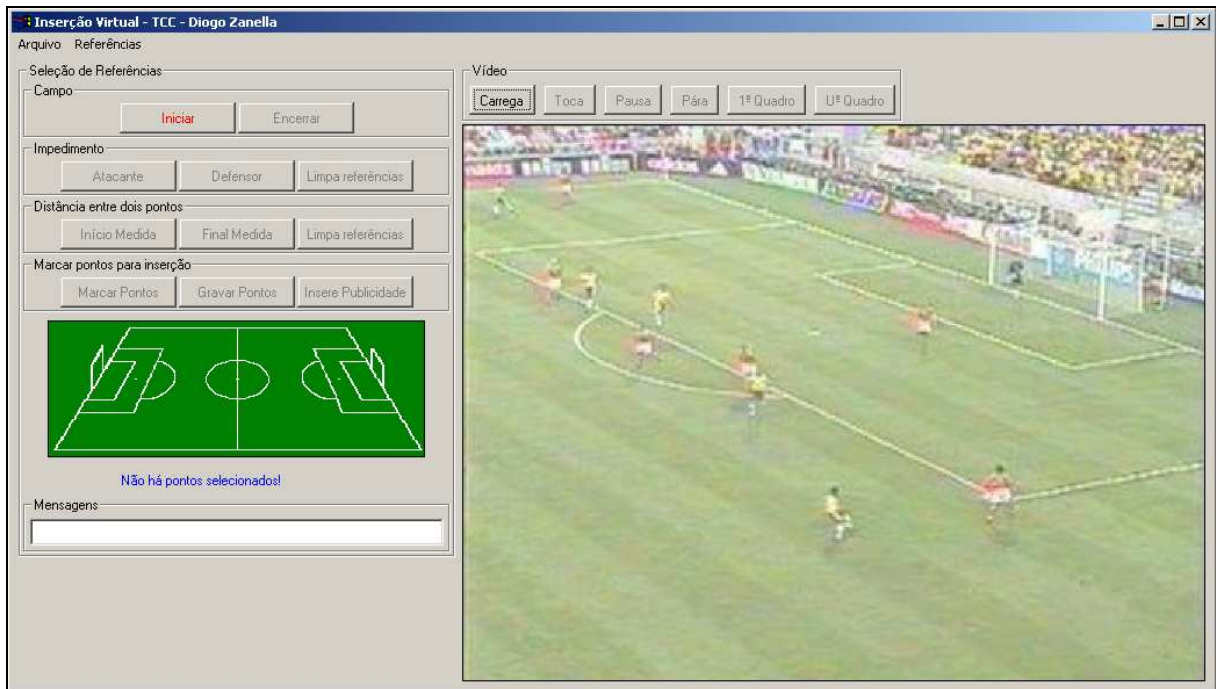


Figura 20 – Carga da imagem

A partir da carga da imagem, dá-se início a fase de calibração de câmera. Para isso é necessário dar um clique em “Iniciar”. Após clicar neste botão, deve-se marcar os pontos de referência para a calibração, escolhendo ponto a ponto, primeiramente na imagem e, logo em seguida no modelo de campo. Existem vários pontos que podem ser selecionados como as intersecções das linhas do campo, o centro do campo e marca do pênalti.

O protótipo pede que sejam selecionados, no mínimo, 5 pontos de referência. Selecionados estes pontos clica-se em “Encerrar” para fazer o cálculo da homografia e liberar os botões para outras funções. A figura 21 mostra a tela do protótipo com 10 pontos selecionados.

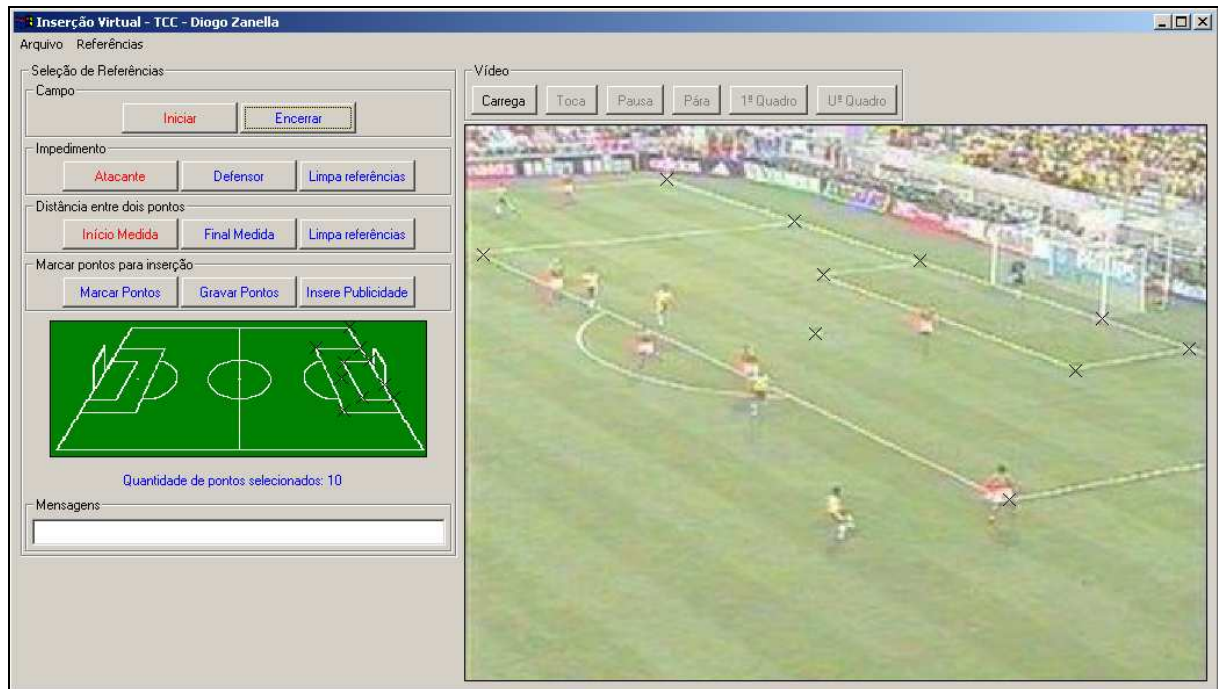


Figura 21 – Pontos de referência selecionados

Terminado o processo de calibração, inicia-se a seleção do local em que a imagem de publicidade será inserida. Esta seleção se dá clicando no botão “Marcar Pontos” e, em seguida, marcando os quatro pontos que servirão para demarcar os limites da publicidade. Depois de marcados, os pontos precisam ser gravados e isso se faz através de um clique em “Gravar Pontos”. Na figura 22 é mostrado um exemplo de seleção dos pontos de localização.



Figura 22 – Localização para inserção da publicidade

Com os pontos de localização devidamente marcados, finalmente pode-se fazer a inserção da publicidade na posição escolhida. Para isso clica-se em “Inserir Publicidade” e, em seguida, em qualquer lugar da imagem, fazendo com que a figura de publicidade apareça entre os quatro pontos selecionados. O resultado desse estudo de caso pode ser observado na figura 23.

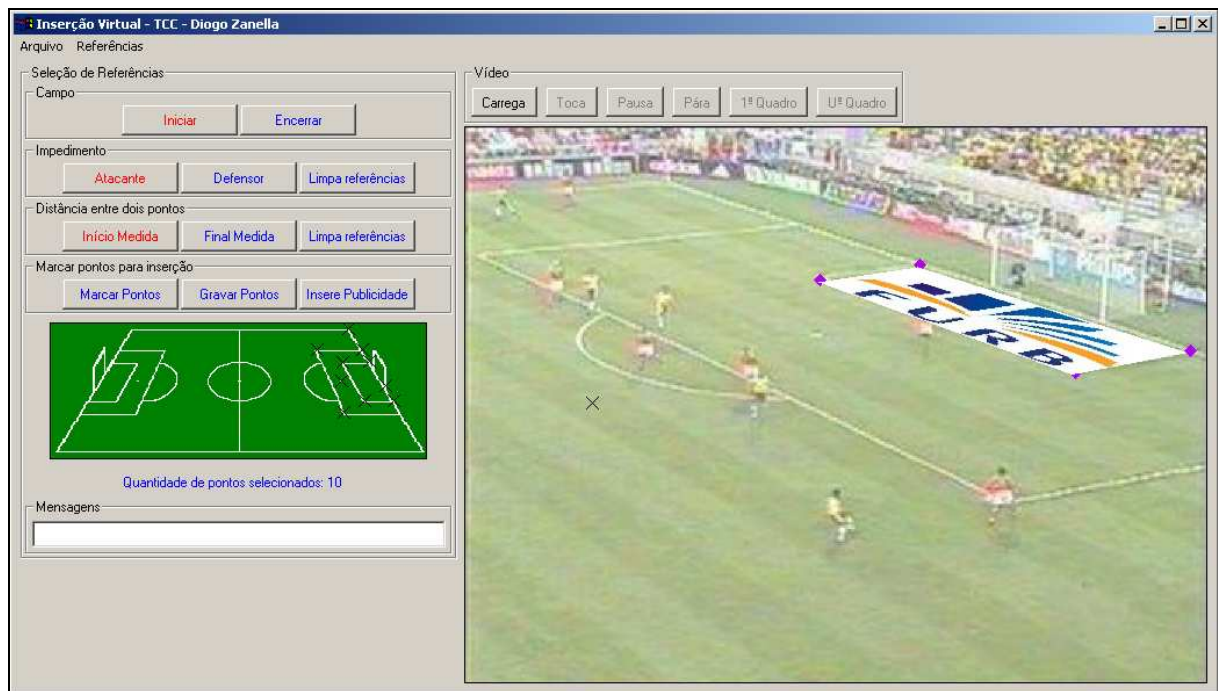


Figura 23 – Resultado final

Como são guardados os pontos no mundo real da localização da publicidade, é possível inseri-la automaticamente no mesmo local em outra imagem, somente calibrando a câmera na outra imagem, como pode ser visto na figura 24.

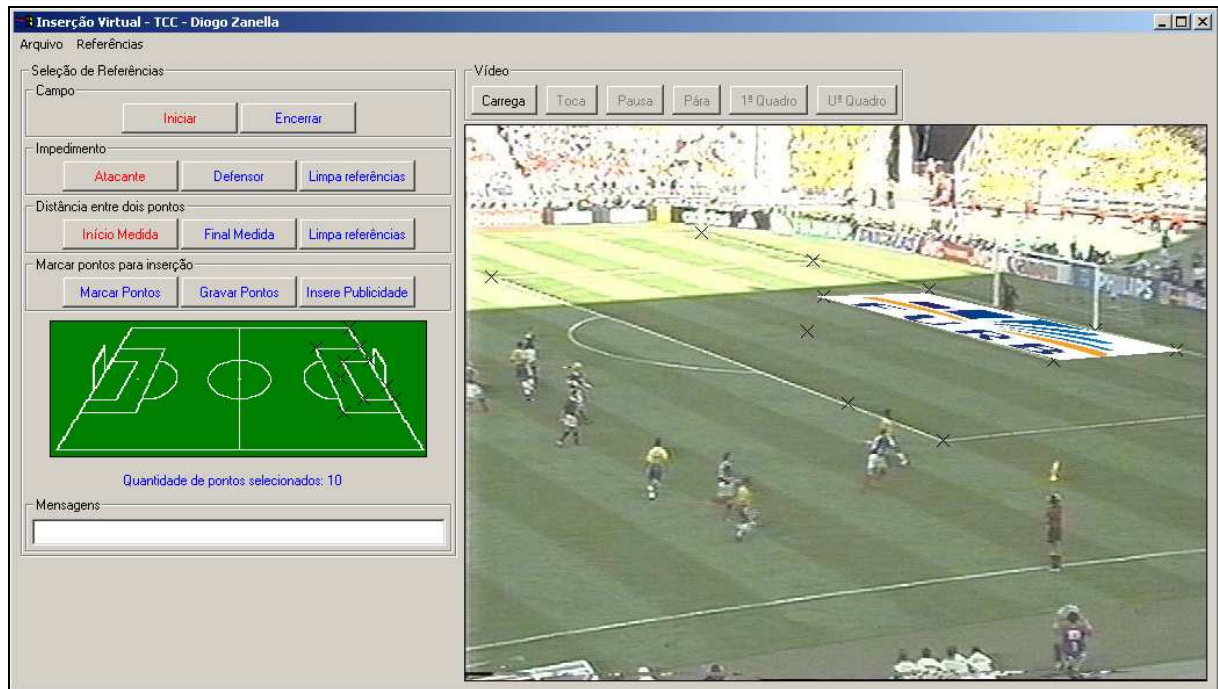


Figura 24 – Inserção em outra imagem

3.4 RESULTADOS E DISCUSSÃO

A maior dificuldade no desenvolvimento do protótipo foi compreender e utilizar as bibliotecas para conseguir fazer o mapeamento de textura no local escolhido. Depois de muito estudo dos parâmetros usados nas bibliotecas e tentativas foi alcançado o resultado desejado.

Foi possível observar também uma enorme diferença na precisão dos cálculos dependendo do número de pontos selecionados, ou seja, percebeu-se que quanto mais pontos de referência o usuário selecionar, mais preciso será o resultado, tanto no cálculo de distância entre pontos com na inserção de publicidade virtual.

A precisão deixou a desejar também no caso de marcarem-se os pontos em uma imagem e depois inserir a publicidade em uma imagem em que a perspectiva seja muito diferente da anterior, como mostra a figura 25.



Figura 25 – Falta de precisão

Neste caso, a localização da publicidade foi marcada na mesma imagem do estudo de caso apresentado anteriormente e depois inserida na cena acima. Pode-se notar que a publicidade não ficou exatamente no local da pequena área como na cena anterior. Isso aconteceu porque a segunda cena está sendo vista de um ângulo um pouco diferente da primeira.

4 CONCLUSÕES

Este trabalho apresentou o estudo e o desenvolvimento de um protótipo para inserção virtual de publicidade em imagens estáticas de jogos de futebol.

As maiores dificuldades encontradas foram na parte final, quando foi preciso inserir a publicidade virtual no local desejado. A princípio, a idéia era desenvolver um algoritmo ou implementar algum já existente para, pixel a pixel, inserir a imagem publicitária no local desejado. Estudando melhor o problema, foi visto que não era necessário “reinventar a roda” e partiu-se para a idéia de usar mapeamento de textura para fazer a inserção virtual da publicidade. Isso fez com que fosse preciso fazer algumas alterações em componentes do protótipo que serviu de base, pois inicialmente usava-se um canvas CD, enquanto o protótipo em desenvolvimento precisaria usar um canvas OpenGL. Isso ocorreu porque a biblioteca CD não suporta mapeamento de texturas.

Terminada a conversão, foi preciso estudar as formas de fazer o mapeamento de texturas utilizando as funções OpenGL, o que tomou a maior parte do tempo de implementação, já que existem alguns parâmetros das funções que precisam ser rigorosamente seguidos, como o tamanho da imagem que seria inserida como publicidade.

Alguns dos problemas foram solucionados com o auxílio do orientador, enquanto outros precisaram de dicas do autor da biblioteca IM para serem solucionados, porém todos os objetivos foram satisfatoriamente alcançados.

A principal limitação do protótipo é a falta de precisão em alguns casos, como quando existem poucos pontos de referência ou quando a publicidade é inserida em uma cena diferente e com uma perspectiva (ângulo de visão) muito diferente da qual onde foram marcados os pontos de localização da inserção. Porém este problema pode servir de motivação para a continuidade deste trabalho.

Outro ponto importante é que a continuidade deste trabalho pode seguir mais de uma linha. Pode-se partir para o uso de vídeos ao invés de imagens estáticas ou apenas reestudar os algoritmos de calibração de câmeras, com o objetivo de melhorar a precisão.

4.1 EXTENSÕES

Uma idéia de extensão deste trabalho seria a implementação do mesmo, possibilitando a inserção de publicidade virtual em vídeos de jogos de futebol, ou até mesmo de outros tipos de jogos, como vôlei, tênis, etc.

Também se pode implementar uma espécie de filtro para que a imagem publicitária não “esconda” os jogadores, ou seja, para que o tele-espectador tenha a impressão de que os jogadores estejam em cima da publicidade, como se a imagem estivesse mesmo desenhada no gramado do campo de futebol.

Pode-se também melhorar a precisão nos casos descritos na seção de resultados e discussões, estudando um melhor algoritmo de homografia e até disponibilizando uma melhor interface para que os pontos sejam marcados mais precisamente.

REFERÊNCIAS BIBLIOGRÁFICAS

AZEVEDO, Eduardo; CONCI, Aura. **Computação gráfica: teoria e prática**. Rio de Janeiro: Elsevier, 2003.

CRISTOFOLINI, Diogo. **Protótipo de um ambiente virtual tridimensional para utilização no cálculo de impedimento de jogadores de futebol**. 2004. 52 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

FOLEY, James D. et al. **Computer graphics: principles and practice**. New York: Addison-Wesley, 1996.

Fórum Unidev. Sorocaba, 2005. Disponível em: <http://unidev.com.br/forum/topic.asp?TOPIC_ID=20421>. Acesso em: 13 jun 2006.

FURLAN, Jose Davi. **Modelagem de objetos através da UML-The Unifield Modeling Language**. Sao Paulo : Makron Books do Brasil, 1998. xiv, 329p, il.

GOMES, Paulo César Rodacki. **Desmistificando a publicidade virtual**. 1999. 24 f. Guia informativo sobre a tecnologia de publicidade virtual. Publicação interna da Rede Globo de Televisão, Divisão de Engenharia de Multimídia. Central Globo de Engenharia, Rio de Janeiro.

HAGEN, Jean Geard. **Protótipo de software para inserção de publicidade virtual em vídeos de jogos de futebol**. 2005. 54 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

HILL, Francis S. **Computers graphics using OpenGL**. Upper Saddle River, New Jersey: Prentice Hall, 2001.

KOSER, Everton Elvio. **Reconhecimento automático de linhas de campos de futebol em arquivos de vídeo para publicidade virtual**. 2003. 58 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Departamento de Ciências Naturais e Exatas, Universidade Regional de Blumenau, Blumenau.

OKAMOTO, Renato Massayuki. **Mapeamento de textura e visualização de terrenos**. Rio de Janeiro, 2003. Disponível em: < <http://orion.lcg.ufrj.br/seminarios/Texturas.ppt>>. Acesso em: 19 maio 2006.

ORAD. **Orad**: hi-tech systems limited. Israel, [2004?]. Disponível em: <<http://www.orad.tv>>. Acesso em: 19 set. 2005.

STAROSKY, Maiko. **Calibração de câmeras para utilização no cálculo de impedimentos de jogadores de futebol a partir de imagens.** 2003. 55 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

SZENBERG, Flávio. **Acompanhamento de cenas com calibração automática de câmeras.** 2001. 170 f. Tese (Doutorado em Computação Gráfica) - Departamento de Informática, Pontifícia Universidade Católica, Rio de Janeiro.

TECGraf. **Ferramentas de desenvolvimento de programas.** Rio de Janeiro, [2003?]. Disponível em: <http://www.tecgraf.puc-rio.br/f_prodp/ferr_c.htm>. Acesso em: 19 set. 2005.

WANGENHEIM, Aldo von. **Visão computacional.** Florianópolis, [2003?]. Disponível em: <<http://www.inf.ufsc.br/~visao>>. Acesso em: 15 set. 2005.