

Esteban Walter Gonzalez Clua

Impostores com Relevô

Tese de doutorado

Departamento de Informática

Rio de Janeiro, 2 de abril de 2004

**Esteban Walter Gonzalez Clua**

**Impostores com Relevô**

**Tese de Doutorado**

Tese apresentada como requisito parcial para obtenção do título de Doutor pelo Programa de Pós-Graduação em Informática da PUC-Rio.

Orientador: Bruno Feijó  
Co-orientador: Marcelo Dreux

Rio de Janeiro, 2 de abril de 2004

**Esteban Walter Gonzalez Clua**

## **Impostores com Relevo**

Tese apresentada como requisito parcial para obtenção do título de Doutor pelo Programa de Pós-Graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

**Bruno Feijó**

Orientador  
PUC-Rio

**Marcelo Dreux**

Co-orientador  
PUC-Rio

**Waldemar Celes**

PUC-Rio

**Manuel Menezes Oliveira**

UFRGS

**Hélio Lopes**

PUC-Rio

**Luiz Eduardo Sauerbronn**

UFRJ

**Edilberto Strauss**

UFRJ

**Sidnei Paciornik**

Puc-Rio

**José Eugênio Leal**

Coordenador(a) Setorial do Centro Técnico Científico - PUC-Rio

Rio de Janeiro, 2 de abril de 2004

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

### **Esteban Walter Gonzalez Clua**

Graduou-se em Ciência da Computação pelo IME-USP em 1996. É mestre em informática na área de Computação Gráfica pela PUC-Rio. Atualmente é pesquisador do ICAD-Igames, Puc-Rio e trabalha no desenvolvimento de games e ferramentas para a área.

#### Ficha Catalográfica

Clua, Esteban Walter Gonzalez

Impostores com Relevo / Esteban Walter Gonzalez Clua; orientador: Bruno Feijó, co-orientador: Marcelo Dreux. – Rio de Janeiro: PUC, Departamento de Informática, 2004. v., 127 f.: il. ; 29,7 cm

1. Tese (doutorado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática. Inclui referências bibliográficas.

Modelagem baseada em imagens, Rendering baseado em imagens, jogos para computador, texturas com relevo, impostores, impostores com relevo, sistemas distribuídos para visualização, pipeline gráfico programável.

Para meus pais, meus irmãos e meus orientadores.

## Agradecimentos

Se cheguei a este ponto, é porque consegui acabar a tese! E ter conseguido terminá-la se deve a muitas pessoas, para quem palavras são pouco para retribuir.

Agradeço aos meus pais, que sempre foram modelo para mim, até no aspecto acadêmico. Posso dizer que cresci dentro de um centro de pesquisas... De igual maneira, agradeço aos meus irmãos, que sempre foram grandes amigos para mim e tiveram um papel importante por ter me apaixonado pela área de games.

Agradeço aos Bruno Feijó e Marcelo Dreux, a quem considero como verdadeiros amigos, antes de orientadores. OBRIGADO MESMO, do fundo do coração!

Agradeço aos que me ajudaram na maior das boas vontades a poder implementar muitas coisas: Francisco Fonseca, Fábio Policarpo, César Pozzer, Lauro Kozovitz, Gilliard Lopes, Lucas Machado... O que seria de mim sem vocês?... Também devo muito a alguns professores e pesquisadores que em algum momento me ajudaram. Se não fossem algumas conversas e dicas, não teria tido as idéias deste trabalho: Luiz Velho, Waldemar Celes, Manuel Oliveira, Noemi Rodrigues, Maria das Graças, Luiza Novaes, Marcelo Gattass...

Finalmente, agradeço a todos os que me ajudaram e orientaram, de alguma forma, a encontrar minha paixão pela computação gráfica, pelo entretenimento digital (games...) e obviamente ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pela ajuda financeira; Agradeço também à FINEP pela concretização do VisionLab, ao qual este trabalho está vinculado.

Dedico este trabalho a todos vocês, a todos os membros da tripulação do ICAD-IGames e a muitos outros cuja amizade considero o que de melhor tenho.

## Resumo

Clua, Esteban Walter Gonzalez. **Impostores com relevo**. Rio de Janeiro, 2004. 127 p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

O presente trabalho introduz o conceito de impostores com relevo: uma maneira eficiente para representar objetos por imagens em sistemas que requerem visualização em tempo real, especialmente jogos 3D e ambientes de realidade virtual. Para tanto, mesclam-se métodos tipicamente pertencentes à área de renderização baseada em imagens com métodos tradicionais de visualização baseada em geometria. A técnica requer do usuário apenas a modelagem geométrica da entidade a ser representada. Posteriormente o sistema sintetiza texturas com relevo, dinamicamente atualizadas quando necessário, e as visualiza utilizando o método de mapeamento de texturas com relevo. Esta abordagem permite inserir modelos complexos, tanto pela sua natureza geométrica, como pelo seu processo de visualização, no *pipeline* gráfico em tempo real. Além disso, os impostores com relevo procuram aproveitar o tempo ocioso ou recursos paralelos disponíveis no processador, de forma a balancear a carga de processamento de visualização entre CPU/GPU. Estes impostores também tornam possível a representação de qualquer tipo de objeto geométrico através de mapeamento de texturas com relevo.

## Palavras-chave

Modelagem baseada em imagens, Rendering baseado em imagens, jogos para computador, texturas com relevo, impostores, impostores com relevo, sistemas distribuídos para visualização, pipeline gráfico programável.

## Abstract

Clua, Esteban Walter Gonzalez. **Relief Impostors**. Rio de Janeiro, 2004. 127 p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The present work introduces the concept of relief impostors: an efficient manner of representing objects by images in systems that require real time rendering, such as 3D games and virtual reality environments. For this purpose, typical methods of image-based rendering are mixed with traditional geometry based rendering methods. This technique only requires from the user the geometric modeling of the entity to be represented. After this, the system synthesizes relief textures, dynamically refreshed when necessary, and renders them using the method of relief texture mapping. This approach allows complex models to be inserted into the real time pipeline system. This complexity arise either from the geometric nature of the model or its process of visualization. Also, the relief impostors try to use the idle time or parallel resources available on the processor, in order to balance the work to be done between the CPU and GPU. Furthermore, they make possible the representation of any kind of geometric object by the relief texture mapping technique.

## Keywords

Image-based modeling, image-based rendering, 3D computer games, relief textures, impostors, relief impostors, distributed visualization systems, programmable graphic pipeline.



# Sumário

1	Introdução	18
1.1	Objetivos do trabalho	18
1.2	Conceitos Envolvidos	20
1.3	Estrutura da dissertação	21
1.4	Contribuições Alcançadas	22
2	Renderização Baseada em Imagens	24
2.1	Introdução	24
2.2	A Função Plenóptica	24
2.2.1	Modelagem de cenários completos	26
2.2.2	Modelagem de panoramas	28
2.2.3	Aplicações de <i>ibr</i> para <i>cache</i> e <i>culling</i>	29
2.2.4	Modelagem de objetos por imagens	33
2.3	Componentes de classificação para os métodos de <i>ibr</i>	34
2.4	3D <i>Image Warping</i>	38
2.4.1	Definição de <i>Warping</i> em Imagens	38
2.4.2	View-Morphing	38
2.4.3	3D <i>Image Warping</i>	42
2.5	Discussão	46
3	Modelagem de Objetos Baseada em imagens	47
3.1	Introdução	47
3.2	Sprites e Billboards	48
3.3	Impostores	50
3.4	Texturas com Relevo	52
3.4.1	Ordem Compatível com Oclusão	58
3.4.2	Texturas com relevo em panoramas cilíndricos	61

3.4.3 Representação de objetos 3D utilizando um conjunto de texturas com relevo	61
4 Impostores com Relevo	65
4.1 Introdução	65
4.1.1 Multi-resolução para Impostores com relevo	68
4.2 Discussão	70
5 Medida de Erro para Impostores com relevo	71
5.1 Introdução	71
5.2 Criação do Impostor com Relevo	72
5.3 Atualização do Impostor com Relevo	72
5.4 Métrica de Erro Acumulado para Impostores com relevo	75
5.5 Métrica de Erro baseado no ponto crítico do Impostor com Relevo	78
5.6 Métrica de Erro baseado em amostragem de pontos críticos	80
5.7 Discussão	81
6 A GPU	82
6.1 Introdução	82
6.2 GPU's e Renderização Baseada em Imagens	84
6.3 A Linguagem Cg	85
6.4 Cálculo de Iluminação <i>Per-Pixel</i> utilizando <i>pipeline</i> programável e mapa de normais	85
6.5 Implementação de Texturas com Relevo em Hardware	87
6.6 Simulação de Shading para sprites sem normal-maps	91
6.7 Discussão	92
7 Processamento Paralelo	93
7.1 Introdução – Classificação de Sistemas Paralelos	93
7.2 <i>Multi-threading</i> e <i>Hyper-threading</i>	94
7.3 Paralelismo em pipelines de visualização tempo real	95

7.4	Paralelismo e os Impostores com relevo	96
8	Implementação e Resultados Práticos	99
8.1	Framework utilizado	99
8.2	Implementação básica do estágio de <i>pre-warping</i>	100
8.2.1	Amostragem Unidimensional Realizada em dois passos	101
8.2.2	Amostragem Assimétrica Realizada em dois passos	103
8.2.3	Amostragem Realizada em Dois Passos com Compensação de Deslocamento	103
8.2.4	Amostragem Intercalada Realizada em um Passo	103
8.3	<i>Pre-warping</i> serial no <i>pipeline</i> gráfico	104
8.4	<i>Pre-warping</i> com <i>Time-Slice</i> fixo	106
8.5	<i>Pre-warping</i> com <i>time-slice</i> variável	106
8.6	<i>Pre-warping</i> com <i>multi-threading</i>	107
8.7	<i>Pre-warping</i> multi-processado	109
8.8	<i>Pre-warping</i> com atualização dinâmica dos Impostores com relevo	112
9	Conclusão	117
9.1	Contribuições	117
9.2	Trabalhos Futuros	118
	Referências Bibliográficas	121

## Lista de figuras

figura 1.1 – Carga de processamento da CPU num jogo 3D	19
figura 1.2 – Exemplo de um impostor com relevo	20
figura 2.1 – Representação gráfica da função plenóptica	25
figura 2.2 – Exemplos de portais	30
figura 2.3 – Grafo para representação de portais	31
figura 2.4 – <i>View dependent textures</i> para portais	32
figura 2.5 – Exemplo de imagens fontes para o <i>view morphing</i>	38
figura 2.6 – <i>Morphing</i> que não é <i>shape preserving</i>	39
figura 2.7 – <i>View morphing</i> : movimento paralelo da câmera	41
figura 2.8 – <i>View morphing</i> : transformação completa nas imagens	42
figura 2.9 – <i>3D image warping</i> : descrição da projeção de um ponto	43
figura 3.1 – Exemplo de um <i>sprite</i>	47
figura 3.2 – Exemplo de <i>bump-mapping</i> e <i>displacement-mapping</i>	52
figura 3.3 – Modelos de câmera perspectiva e ortogonal	54
figura 3.4 – Etapas da fatorização da equação de McMillan	55
figura 3.5 – Etapa de <i>pre-warping</i> da equação de McMillan	56
figura 3.6 – Tratamento de conflito de <i>pixels</i> para o <i>warping</i>	58
figura 3.7 – Ordem compatível de oclusão	59
figura 3.8 – Texturas com relevo para espaços cilíndricos	60
figura 3.9 – Objeto sendo representado por 6 texturas com relevo	61

figura 4.1 – Topologia incorreta para objeto de 6 texturas com relevo	63
figura 4.2 – Polígonos não visíveis pelas seis vistas ortogonais	64
figura 4.3 – Regiões para validade de um impostor com relevo	65
figura 4.4 – Multi-resolução para impostores	67
figura 4.5 – Cálculo para validar resolução do impostor	68
figura 5.1 – Estágios de um impostor com relevo	69
figura 5.2 – Medida de Schaufler para aproximação	71
figura 5.3 – Medida de Schaufler para movimento paralelo	72
figura 5.4 – Preenchimento de buracos por <i>texels</i> interpolados	73
figura 5.5 – Deslocamento vertical e horizontal de um <i>texel</i>	74
figura 5.6 – <i>texels</i> de maior descontinuidade na textura	76
figura 5.7 – Métrica de erro baseada em vários pontos críticos	78
figura 6.1 – Rasterização de polígonos por hardware	81
figura 6.2 – Conflito de <i>texels</i> no processo de <i>pre-warping</i>	86
figura 6.3 – Objeto com 1 textura com relevo implementado em GPU	87
figura 6.4 – Paralelepípedo de texturas com relevo em GPU	88
figura 7.1 – Estágios na visualização dos impostores com relevo	95
figura 7.2 – Dependência entre os processos paralelos	95
figura 7.3 – Sistema de previsão para diminuir tempo de espera	96
figura 8.1 – Diagrama da estrutura do <i>framework</i> desenvolvido	97
figura 8.2 – Interpolação para a amostragem unidimensional	100
figura 8.3 – <i>Framework</i> com <i>pre-warping</i> serial	102
figura 8.4 – Distribuição entre CPU e GPU da implementação serial	103
figura 8.5 – <i>Framework</i> para <i>time slice</i> do <i>pre-warping</i>	105

figura 8.6 – Máquina de estados para sincronizar o <i>pre-warping</i>	106
figura 8.7 – <i>Framework</i> para texturas com relevo com <i>multi-thread</i>	106
figura 8.8 – Detalhamento do <i>framework</i> com <i>multi-thread</i>	107
figura 8.9 – Arquitetura do sistema para multi-processamento	108
figura 8.10 – Como dividir a imagem para n pedaços distintos	108
figura 8.11 – Exemplos de Impostores com relevo	111
figura 8.12 – Sistema com otimização e métricas de erro	112
figura 8.13 – Renderização com <i>software shader</i> e com GPU	113
figura 8.14 – <i>Software shader</i> e tempo de consumo de CPU	113

## Lista de tabelas

tabela 6.1 – Desempenho para objeto com 1 textura com relevo	87
tabela 6.2 – Desempenho para objeto com 6 texturas com relevo	88
tabela 8.1 – Performance obtida para os algoritmos de amostragem	102
tabela 8.2 – Performance obtida para abordagens paralelas e serial	110
tabela 8.3 – performance com e sem teste de Schaufler	111

## **Conteúdo do CD**

Código fonte completo do Framework utilizado

Recursos necessários para executar aplicações (modelos, texturas, etc.)

Implementação Framework com pre-warping serial

Implementação do Framework para time slice do pre-warping

Implementação do Framework para texturas com relevo com multi-thread

Implementação do Pre-warping com atualização dinâmica dos Impostores com relevo



I am the Architect. I created the Matrix. I've been waiting for you.

*The Matrix*