Antonia Lucinelma Pessoa Albuquerque

Cenários Virtuais com um Estudo de Sincronismo de Câmera

Dissertação apresentada ao Departamento de Informática da PUC-Rio como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Orientador: Jonas Gomes
Co-Orientador: Marcelo Gattass

Departamento de Informática

Pontifícia Universidade Católica do Rio de Janeiro

Rio de Janeiro, 06 de Abril de 1999.

Aos meus Pais

e
ao João

"Tous les événements sont enchaînés dans le meilleur des mondes possibles ... -
... mais il faut cultiver notre jardin."

Voltaire

Agradecimentos

Este trabalho é fruto de grande esforço ao longo destes anos, mas não seria possível realizá-lo sem o apoio recebido.

Aos professores que contribuiram com o conhecimento e experiência, meu muito obrigada.

Ao meu orientador, Jonas Gomes, agradeço sua orientação valiosa e pela honra de tê-lo como orientador.

Agradeço ao laboratório VISGRAF pelas especiais condições de trabalho que proporciona e aos colegas do VISGRAF/IMPA e da PUC-Rio pela companhia e convivência que tornam o ambiente de trabalho tão agradável.

Um agradecimento especial aos amigos Sérgio Pinheiro e Adelailson Peixoto.

À Internet, sem a qual não teria sido possível realizar este trabalho, o meu reconhecimento.

"Quando sonhamos sozinhos é só um sonho.
Quando sonhamos juntos é o começo de uma nova realidade."

Dom Helder

Nota

A maioria das referências usadas neste trabalho, tanto livros quanto artigos, estão em inglês. Naturalmente, no decorrer dos anos formou-se uma terminologia técnica da área, em inglês. A tradução destes termos para o português, já largamente usados em inglês, ao invés de contribuir para esclarecer poderiam dificultar ou causar uma perda de significado, não acrescentando vantagens a este trabalho. Por isso optei por manter estes termos em inglês ao longo do texto, e tecer explicações específicas sempre que necessário.

Resumo

Técnicas de filmagem com uso de efeitos especiais existem desde a década de 20, muito antes do advento do computador. Duas delas são conhecidas como *Back Projection*, quando um ator representa diante de uma tela que reproduz uma outra filmagem (muito comum em cenas de viagem de trem) e *Blue Screen*, quando um ator representa diante de um fundo azul para posterior composição com outra cena. No entanto foi o avanço da computação gráfica e dos computadores que possibilitaram uma grande evolução nesta área. É com este enfoque que o presente trabalho aborda Cenários Virtuais, descrevendo sua conceituação, e mostrando sua correlação com outras áreas da computação gráfica. As tecnologias relevantes em cenografia virtual são identificadas dentro da computação gráfica e discutidas através de soluções existentes e problemas em aberto. Dentre os problemas apresentados, o trabalho estuda uma técnica baseada em otimização diferencial visando o sincronismo de câmera que permita a interação, em tempo real, das imagens real e sintética.

Abstract

Techniques of filming using special effects have existed since the 1920's, well before the advent of computers. Two of them are known as Back Projection - when an actor acts in front of a screen that reproduces other footage (very common in train scenes), and Blue Screen - when an actor acts in front of a blue wall for later composition with another scene. However, it was computer graphics and the computer's advance that made possible great evolution in this area. This work approaches Virtual Sets, describing its conceptualization, and showing its correlation with other areas in computer graphics. The virtual sets' pertinent technologies are identified in computer graphics and have their given solutions and unsolved problems argued. Amongst the presented problems, the work studies one technique based on differential otimization aiming at the synchronism of camera that would allow interaction, in real time, of the real and synthetic images.

Índice

Lista de Figuras

Crédito das Figuras

As figuras citadas abaixo foram obtidas pela Internet e são propriedades dos respectivos sites, que poderão vir a ser alterados ou mesmo desativados por responsabilidade de seus proprietários:

2.1 - http://www.evl.uic.edu/pape/CAVE/
2.2 - http://www.evl.uic.edu/EVL/VR/ImmersaDesk.shtml
2.3 - http://www.evl.uic.edu/EVL/VR/systems.shtml#idesk
2.4 - http://www.cc.gatech.edu/gvu/virtual/Phobia/
2.5 - http://www.cc.gatech.edu/gvu/virtual/CDS/
2.6 - http://viswiz.gmd.de/VMSD/PAGES.en/projects.workbench.html
2.7 e 2.8 - http://www.hitl.washington.edu/publications/p-95-17/
2.9 - http://www.csl.sony.co.jp/project/VS/index.html
2.10 e 2.11 - http://www.ai.mit.edu/projects/medical-vision/surgery/surgical_navigation.html
2.12 - http://www.cs.unc.edu/∼us/
2.13 - http://www.cs.columbia.edu/graphics/projects/archAnatomy/architecturalAnatomy.html
2.14 - http://www.orad.co.il/virsets/index.htm
2.15 - http://www.rtset.co.il/larus.htm
2.16 - http://www.visgraf.impa.br/Projects/virtualset/page04.html
2.17 - http://www-cui.darmstadt.gmd.de/visit/Activities/Vist/Introduction/
2.18 - http://www.dcs.qmw.ac.uk/research/parallel/monalisa/arrows.html
2.19 - http://www.cs.cmu.edu/afs/cs/project/stereo-machine/www/z-key.html
2.20 - http://www-inria-graphlib.inria.fr/equipes/syntim/analyse/video-eng.html

3.1 - http://www.orad.co.il/what/index.htm, em FRANCE '98 WORLD CUP NEWS
3.3 - http://website.lineone.net/∼radamec_broadcast/virtual1.html
3.4 e 3.5 - http://www.sgi.com/onyx2/sys_hardware.html

4.1 - http://www.ashera.com/3D.html
4.3 - http://reality.sgi.com/opengl/tips/rts/
4.4 - http://www.sgi.com/apps/geospatial_imaging/irquality_wp.html
4.5 - http://www.cs.ubc.ca/spider/fearing/home.html
4.7 - http://www.cs.berkeley.edu/∼healey/ibr/ibr.shtml

5.8 - http://www.cs.cmu.edu/afs/cs.cmu.edu/project/stereo-machine/www/StereoMachine.html
5.9 - http://www.dcs.qmw.ac.uk/research/parallel/monalisa/maps/../presenter.html
5.10 - http://www.cs.cmu.edu/afs/cs/project/stereo-machine/www/z-key.html

6.1 - http://www.visgraf.impa.br/juizvirtual/metodo.htm, extraídas do arquivo *ppt*
6.2 - http://www.visgraf.impa.br/Projects/virtualset/page02.html

6.3 - http://website.lineone.net/∼radamec_broadcast/freed.html
6.4 - http://www.dcs.qmw.ac.uk/research/parallel/monalisa/maps/../presenter.html
6.5 - http://www.orad.co.il/virsets/index.htm

As figuras abaixo são cortesia de seus autores:

5.6 e 5.7 - cedidas por (Gomes *et al.* , 1998)

Capítulo 1

Introdução

A utilização de efeitos especiais nas produções cinematográficas vem evoluindo de maneira gradativa com o advento do computador. A cada passo na evolução da computação gráfica conta-se com maiores recursos para realizar-se efeitos que antes pareciam impossíveis ou muito dispendiosos para serem concretizados.

Esta dissertação trata desta produção de efeitos especiais, sob o nome de *Cenários Virtuais*, vista pela ótica da computação gráfica, o que antes era tratado apenas como Cinematografia. *Estúdios Virtuais* ou *Cenários Virtuais* são denominações dadas às técnicas de filmagem nas quais o ator não interage com o cenário, que é gerado por computador.

O reconhecimento de Cenários Virtuais como uma área de aplicação da computação gráfica é muito recente. Por ser pouco reconhecida sob este aspecto e principalmente por ser uma área que faz fronteira com muitas outras linhas de pesquisa que estudam problemas semelhantes, encontra-se uma sobreposição de denominações a diferentes linhas de pesquisa.

1.1 Áreas correlatas

No Capítulo 2 será apresentada uma análise com o intuito de clarificar e explicar as diferentes áreas de pesquisa relacionadas. Todas estas áreas têm em comum o objetivo de **integrar mundos real e virtual** (no sentido de gerado por computador), com diferentes finalidades de aplicações e, portanto, exigindo muitas vezes estudos específicos para técnicas semelhantes.

A integração do mundo do homem (real) com o mundo do computador (virtual) está cada vez mais presente e pretende ser a comunicação de um futuro mais próximo do que se pode imaginar. Muitos benefícios já podem hoje ser obtidos destes resultados.

Baseado nesta abordagem, quatro áreas serão citadas como pertencentes a uma classificação que não se encontra bem definida na literatura correspondente. São a saber:

- Realidade Virtual

- Ambientes Virtuais Colaborativos
- Realidade Aumentada
- Cenários Virtuais

Diversas aplicações com objetivos diferentes, nestas quatro áreas, serão citadas no Capítulo 2.

1.2 Sistemas de Cenários Virtuais

Estes sistemas utilizam diversas técnicas de composição de imagem que combinam imagens reais, filmadas por uma câmera em um estúdio real, com outras imagens reais ou com imagens sintéticas (geradas por computação gráfica), em tempo real ou não. São técnicas muito importantes na produção de cinema e programas de televisão. Atualmente, alguns sistemas comerciais desenvolvidos para aplicações de cenários virtuais encontram-se disponíveis no mercado e serão citados no Capítulo 2.

Nos sistemas de cenários virtuais pode-se identificar três tipos de "usuários finais":

- atores
- operadores do sistema
- telespectadores

Os atores e operadores participam do processo de autoria. O telespectador é um consumidor dos resultados do sistema. Estes três tipos de usuários impõem diferentes demandas de qualidade ao sistema, tais como:

- para o ator é importante que o sistema proporcione facilidades no ensaio e permita uma visualizacão prévia do cenário virtual no qual ele irá "atuar".
- do ponto de vista do operador, o sistema deve apresentar características mais próximas possíveis do processo convencional de filmagem, não exigindo um total aprendizado de novas técnicas.
- para o telespectador a técnica de cenografia virtual deve ser imperceptível. O objetivo é que o ator pareça estar sendo filmado naquele ambiente.

Atender a todas estas demandas gera um grande número de problemas a serem tratados na elaboração de um sistema de cenografia virtual.

Os estudos desenvolvidos na área de cenários virtuais objetivam, além de produzir efeitos especiais para cinema e televisão, a substituição de cenários convencionais produzidos em estúdios, por cenários sintéticos (gerados por computador).

Pode-se enumerar algumas das várias vantagens no uso de cenários virtuais em relação a cenários convencionais:

- Grande flexibilidade em mudar o cenário de um programa para outro em poucos minutos, usando o mesmo espaço físico do estúdio, gerando portanto grande economia na produção de cenários e reduzindo a necessidade de grandes espaços físicos.

- Diretores também podem usufruir destas técnicas para previamente analisar cenas a partir de diferentes tomadas.

- Cenas nas quais o ator corre riscos podem ser feitas de maneira segura com o uso de cenografia virtual.

- Permite a produção de efeitos especiais mais sofisticados melhorando muito a qualidade dos filmes.

A partir de diferentes técnicas pode-se obter muitos resultados usando recursos que são transparentes ao público. Alguns exemplos são:

- Geração de imagens reais, geração de imagens virtuais e a composição destas imagens.

- As imagens a serem posteriormente combinadas podem ter a mesma escala ou escalas diferentes. Isto permite também a utilização de maquetes, efeito muito usado em grandes produções cinematográficas.

- A composição do ator pode ser feita com outras imagens reais, por exemplo para parecer que o ator está em Paris quando ele está dentro do estúdio.

- Efeitos gráficos utilizando diversas técnicas de computação gráfica e processamento de imagens para obter efeitos de cenografia virtual que não são possíveis por métodos convencionais.

Observe que todos os exemplos citados permitem combinação entre si, dando portanto uma gama muito rica de resultados.

Uma abordagem introdutória de uma aplicação real pode ser vista em (Siegmann, 1994).

1.3 Objetivo da Dissertação

Este trabalho apresenta uma análise das áreas correlatas citadas especificando características técnicas e ilustrando com trabalhos desenvolvidos estas diferentes áreas. Identifica as tecnologias relevantes para cenografia virtual dentro da computação gráfica. Os capítulos serão organizados como um *survey* de cenários virtuais, dissertando sobre principais aspectos, com o objetivo de gerar uma documentação para suprir a escassez de literatura técnica nesta área.

Uma abordagem importante do trabalho trata de um estudo de caso de uma técnica de otimização diferencial visando uma aplicação para *sincronismo de câmera* direcionado para cenário virtual. Sincronismo de câmera consiste em correlacionar corretamente os parâmetros da câmera virtual e da câmera real, o que será bem definido ao longo deste trabalho. Este estudo será especificado no Capítulo 7.

Pode-se perceber que o trabalho tem uma abordagem do geral para o particular, isto é, disserta de maneira geral sobre cenário virtual e áreas correlatas e depois apresenta um estudo de caso específico que pode ser aplicado a esta área. Isto se justifica no que foi dito anteriormente: é uma abordagem recente e tem poucos trabalhos desenvolvidos com esta ótica.

Porque sincronismo de câmera foi escolhido para estudo de caso? Porque este problema constitui o ponto central de um sistema de cenário virtual. Quando o sincronismo depende de equipamentos eletro-mecânicos gera uma grande limitação financeira, pois são equipamentos de alto custo. Aqui está o coração destes sistemas, pois as técnicas de sincronismo utilizadas limitam ou ampliam as habilidades do sistema.

1.3.1 Estrutura da Dissertação

O trabalho é composto dos seguintes capítulos:

Capítulo 2: Áreas Correlatas

Este capítulo apresenta uma análise entre áreas de pesquisa, descreve trabalhos com aplicações em Cenários Virtuais e áreas correlatas, seus diferentes enfoques, resultados e limitações. Relaciona estes trabalhos com a análise apresentada.

Capítulo 3: Cenários Virtuais

Aqui apresenta-se um estudo conceitual de Cenários Virtuais. Mostra diversas questões a serem tratadas quando objetiva resolver problemas específicos correlatos. Essa conceituação identifica as três principais tecnologias relacionadas ao uso de computação gráfica em cenografia virtual, que são: Rendering, Composição de Imagens e Sincronismo. Cada um destes problemas será abordado nos capítulos subsequentes.

Capítulo 4: Rendering

O cenário sintético deve ser renderizado segundo a posição da câmera virtual para obter uma perspectiva adequada do cenário em relação à imagem real. Surgem diferentes abordagens para tratamento do rendering nestes sistemas, principalmente quando a intenção é gerar um cenário em tempo real, trazendo à tona toda complexidade envolvida neste tópico relativa a desempenho, memória, qualidade de imagem e coerência temporal.

Capítulo 5: Composição

Este capítulo trata da composição de imagens. A imagem sintética gerada precisa ser combinada com a imagem real, para parecer um único ambiente. Existem diferentes técnicas e aplicações no domínio da composição de imagens que permitem muitos efeitos diferentes. Esta parte requer também muita atenção e estudo.

Capítulo 6: Sincronismo de Câmera

Todo sistema de cenografia virtual possui duas câmeras a serem controladas, uma denominada câmera real, que é a câmera que filma o ator e a outra, denominada câmera virtual que é a câmera gerada por computador, que controla o cenário sintético. Para haver coerência na imagem final estas câmeras precisam estar na mesma posição em seus respectivos universos. Isto denomina-se sincronismo de câmera. Grande parte do sucesso do resultado final é devido a este controle. Isto será abordado neste capítulo.

Capítulo 7: Estudo de Caso

Em *Through-the-Lens Camera Control* (Gleicher & Witkin, 1992) os autores introduzem um conjunto de técnicas que permitem ao usuário manipular uma câmera virtual pelo controle e restrições de propriedades na imagem vistas através das lentes da câmera. O presente estudo quer mostrar o potencial destas técnicas aplicadas a cenários virtuais, visando sincronismo, em tempo real, das câmeras real e virtual.

Capítulo 8: Contribuições e Conclusões

Neste capítulo serão apresentadas as conclusões obtidas com o trabalho, qual a contribuição efetuada e indica diretrizes para problemas em aberto.

Capítulo 2

Áreas Correlatas

As áreas aqui apresentadas têm como objetivo comum integrar mundos real e virtual dispondo de diferentes técnicas e visando as mais diversas aplicações. Este capítulo apresenta uma análise destas áreas, descreve trabalhos desenvolvidos para cenários virtuais e áreas correlatas, apresentando seus resultados, aplicações e limitações.

Análise

Entende-se por mundo virtual objetos e lugares gerados por computador baseados em um conjunto de dados $3D$, que os descrevem geometricamente para simular o mundo real. A este mundo virtual denomina-se *ambiente virtual*. O ambiente virtual pode ter a característica de ser imersível ou não, dependendo do tratamento dado no sistema. *Imersão* significa permitir ao usuário experimentar o sentimento de presença neste ambiente virtual e interagir com o mesmo, navegando no ambiente, tocando em objetos, escutando sons.

Esta sensação de fazer parte do ambiente virtual assim como a visão tridimensional com profundidade vivenciada pelo usuário, só são possíveis com o uso de equipamentos de visão estéreo (ativo ou passivo) pelo usuário, que propiciam a visão de uma imagem com profundidade a partir de duas imagens geradas com ligeira diferença na posição do observador, denominada informação *paralaxe*.

Devido a estas características, quatro áreas foram identificadas como correlatas: Realidade Virtual, Ambientes Virtuais Colaborativos, Realidade Aumentada e Cenários Virtuais. Existem distinções na abordagem destas áreas em função do grau de relação entre os ambientes virtuais e o mundo real. Alguns sistemas concedem imersão ao usuário, outros têm diferente grau de interação do usuário com o ambiente virtual e alguns caracterizam-se pela ausência total de interatividade e imersão.

Realidade Virtual

Esta área, que é a mais conhecida popularmente, onde as pessoas tendem a classificar o que é gerado por computador como realidade virtual, tem suas características técnicas inerentes para poder receber este nome.

Realidade virtual pode ser definida como um grande campo de apresentação de informação multi-sensorial, gerada de acordo com o comportamento do usuário, em tempo real.

Em realidade virtual existe um ambiente tridimensional gerado por computador e com o qual o homem no mundo real pode interagir com a sensação de estar imerso neste ambiente. Para isto é necessário que o usuário deste sistema, isto é, a pessoa que experimenta esta imersão, utilize algum dispositivo ótico que induza a esta integração com o ambiente virtual e perca o vínculo com o mundo real, onde o usuário realmente está. O dispositivo mais comum é o HMD - *Head-Mounted Display*, que é um tipo de óculos-capacete que permite a visão com perspectiva e profundidades no mundo virtual. O uso de luvas especiais em alguns sistemas permite também a interação tátil com objetos do mundo virtual. O maior alvo destes sistemas é conseguir total imersão do usuário.

Nestes sistemas é o movimento da cabeça do usuário que determina o ângulo de visão, por isso é necessário fazer o acompanhamento deste movimento chamado *head-tracking*. O dispositivo de head-tracking fornece informação de localização e orientação da cabeça do usuário para uma estação de computação gráfica que calcula as imagens na tela, para que estas sejam consistentes com a direção em que o usuário está olhando para o mundo virtual. A quase totalidade das aplicações nesta área são em tempo real e baseada na posição do observador (usuário).

Existem outros tipos de sistema de realidade virtual que proporcionam imersão parcial, que é o caso de anfiteatros de diversões onde as pessoas põem um óculos (chamado óculos passivo) por que elas estão apenas recebendo a imagem, e que pode ter a sua quebra de imersão ao olhar para baixo, por exemplo. Estes sistemas são de imersão parcial porque ainda não está resolvido o problema da imersão total para várias pessoas ao mesmo tempo.

Esta área surgiu como um novo paradigma de interação homem-computador no qual usuários não são simples observadores externos de imagens numa tela de computador mas são participantes ativos num mundo tridimensional, gerado por computador. O usuário fica completamente imerso num mundo artificial e perde o vínculo com o mundo real.

Ambientes Virtuais Colaborativos

Ambientes virtuais colaborativos é a denominação dada às pesquisas voltadas ao desenvolvimento de ambientes que propiciem a realidade virtual de forma compartilhada, por pessoas que estejam em lugares distantes no planeta, necessitando, portanto, de estarem conectadas em rede. É uma abordagem que difere do caso do anfiteatro citado acima, pois lá há coletividade no mesmo ambiente, aqui a pessoa estará compartilhando o mesmo ambiente virtual com outra pessoa à distância.

A concepção e construção de ambientes virtuais colaborativos é inerentemente multidisciplinar e baseada em princípios de percepção, psicologia e física. Em termos concretos, a definição e especificação de um ambiente virtual começa com a análise da aplicação, compreensão do papel das pessoas, do papel das máquinas e da integração de todos estes elementos, para poder atingir seus objetivos.

As pessoas que desenvolvem pesquisas nesta área acreditam ser esta a comunicação do futuro. Pessoas poderão ir ao shopping virtual juntas, embora estejam em países diferentes, ou cientistas de diferentes países poderão simultaneamente trabalhar na mesma pesquisa integrados ao mesmo laboratório virtual. Esta área envolve os problemas da área de computação gráfica e necessita de um grande desenvolvimento na área de redes de alta velocidade e banco de dados multimídia. São aplicações em tempo real, também referenciadas como *telepresença*.

Realidade Aumentada

"Nós acreditamos que um dos mais fortes usos de mundos virtuais não é para substituir o mundo real e sim completar a visão do usuário, no mundo real". Esta idéia, introduzida pelo trabalho pioneiro de Ivan Sutherland (Sutherland, 1968) sobre *head-mounted displays*, é então referenciada como *realidade aumentada*.

Um sistema de realidade aumentada gera uma imagem resultante de uma combinação de imagens. É uma combinação de uma cena real com uma cena virtual, gerada por computador, para enriquecer a cena final com informação adicional. Em todas as diferentes aplicações possíveis desta área, a realidade aumentada apresentada ao usuário melhora a sua performance e percepção do mundo em que ele observa.

Assim como realidade virtual, realidade aumentada requer o uso de dispositivos óticos pelo usuário, mas com o objetivo principal de sobrepor as imagens sintéticas com imagens do mundo real. Aqui existe a composição de imagens numa conotação muito forte, pois é uma sobreposição onde a precisão é de extrema importância e a imagem virtual visa acrescentar informações à imagem real. Com transparência parcial dos "displays" o usuário pode simultaneamente ver o mundo real e as imagens geradas por computador.

Existem muitas semelhanças entre um sistema de realidade virtual e outro de realidade aumentada, principalmente pelo aspecto da imersão e processamento em tempo real, exigindo o controle da posição do usuário. Mas a grande visível diferença entre os dois sistemas está na forma de imersão. Realidade virtual esforça-se para gerar um ambiente totalmente imersível. Os sentidos visual, perceptual e, em alguns sistemas, auditivo ficam sob controle do sistema. Ao contrário, um sistema de realidade aumentada quer enriquecer a cena do mundo real necessitando que o usuário mantenha o sentimento de presença no mundo real, portanto ele não pode ficar totalmente imerso. As imagens reais e virtuais são misturadas para gerar uma imagem mais completa. Portanto, um sistema para combinar estas imagens tem características que não existem num sistema de realidade virtual.

Desenvolver tecnologia para esta combinação constitui uma área de pesquisa.

Cenários Virtuais

Assim como na realidade aumentada, cenários virtuais resultam da combinação de duas imagens, uma do mundo real e outra gerada por computador.

Porque então cenografia virtual pode ser vista como uma linha de pesquisa diferenciada das demais? Porque aqui não há imersão do usuário no ambiente virtual. O usuário telespectador, que é um usuário final deste sistema, não precisa de nenhum dispositivo ótico, ele é passivo. A composição de imagens real e virtual é ilimitada quanto ao seu estilo, as aplicações podem ser feitas em tempo real ou não. E o principal alvo é conseguir que o telespectador não perceba a existência de um cenário virtual, que o ator pareça estar imerso naquele ambiente. O ator não interage, na realidade, com o cenário. A forma como a composição de imagem é tratada aqui difere de como é tratada em realidade aumentada.

Também poderia-se dizer se não é apenas uma área de composição de imagens. Como será descrito ao longo deste trabalho, a composição é parte integrante e fundamental destes sistemas, mas existem outras questões relevantes que constituem um conjunto bem mais complexo de problemas a serem tratados. As características mais detalhadas destes sistemas serão descritas no capítulo 3.

As próximas seções apresentam projetos desenvolvidos nas diferentes linhas de pesquisa. O agrupamento dos trabalhos nas respectivas áreas foi feito com base na análise apresentada.

2.1 Realidade Virtual

É importante frisar que realidade virtual quer atingir imersão total, quebra do vínculo com a realidade, o usuário tem que usar um dispositivo ótico e não há imagens reais combinadas com imagens virtuais.

Os mais importantes projetos desenvolvidos para realidade virtual são descritos a seguir.

- **CAVE**, Electronic Visualization Laboratory(EVL) - University of Illinois, Chicago.

  Em adição aos mais conhecidos dispositivos de realidade virtual - "head-mounted displays" e "binocular omni-oriented monitor(BOOM) displays" - o Electronic Visualization Laboratory introduziu um terceito dispositivo em 1992: uma sala construída com grandes telas nas quais imagens gráficas são projetadas sobre três paredes e no piso. CAVE, o nome escolhido para o teatro de realidade virtual, é simultaneamente um acrônimo recursivo (CAVE Automatic Virtual Environment).

  Desde o desenvolvimento do CAVE, o Electronic Visualization Laboratory tem-se especializado como principal área, na pesquisa e desenvolvimento de software, hardware e ferramentas de comunicação em rede para Realidade Virtual .

O CAVE é um compartimento para várias pessoas, com alta-resolução, onde a ilusão de imersão é criada pela projeção de imagens $3D$ que envolvem completamente o usuário. O sistema faz o tracking da cabeça e da mão do usuário e complementa com o áudio permitindo que a pessoa explore o mundo virtual movendo-se pelo compartimento e tendo a sensação de arrastar objetos.

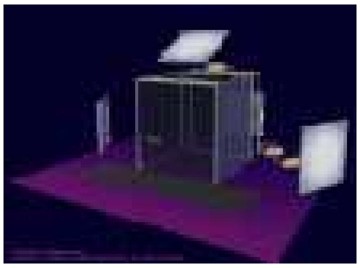


Figura 2.1: CAVE

Está sendo comercializado pela Pyramid Systems Inc. A pesquisa e desenvolvimento do CAVE continuam em andamento. Neste sistema todas as projeções são calculadas a partir do ponto de vista do usuário. Um head-tracker fornece a informação sobre a posição do usuário e as imagens são calculadas para cada olho. Para usufruir do efeito estéreo, o usuário coloca um óculos estéreo ativo que alternadamente bloqueia o olho esquerdo e o direito.

Os principais objetivos que levaram ao desenvolvimento do CAVE são:

1. apresentar imagens coloridas de alta resolução;
2. diminuir a quantidade de equipamentos pesados que o usuário tem que usar nesse tipo de sistema;
3. reduzir distorções geométricas;
4. diminuir a sensibilidade a erros induzidos por rotações da cabeça;
5. desenvolver ferramentas de visualização que possam ser acopladas a supercomputadores em rede e fontes de dados (Cruz-Neira *et al.* , 1992).

- **ImmersaDESK**, Electronic Visualization Laboratory(EVL).

O ImmersaDesk é um móvel onde sua maior vantagem está na portabilidade e baixo custo. É um dispositivo com o formato de uma mesa plana com um computador operado por um sistema de áudio. Este sistema baseado em projeção oferece um tipo de realidade virtual que é semi-imersivo. Ao invés de envolver o usuário com imagens gráficas e bloquear o mundo real, o ImmersaDesk oferece uma tela de projeção posterior, a um ângulo de 45 graus. É necessário o uso de óculos estéreo, um dispositivo

Figura 2.2: ImmersaDesk

de áudio e *hand tracking* para fazer o tracking das mãos. O tamanho e posição da tela fornece um amplo ângulo de visão suficiente para permitir ao usuário olhar para baixo. A resolução é 1024 x 768 e com 96Hz (Cruz-Neira *et al.* , 1994).

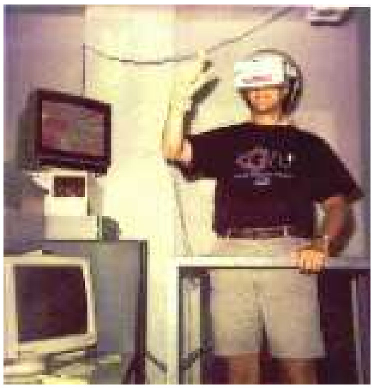
- **IWall**, Electronic Visualization Laboratory(EVL).

Figura 2.3: IWall

É uma grande tela de alta resolução, com estéreo ativo ou passivo, que pode ser acompanhada por grandes públicos. Suporta áudio e é operado por duas estações Silicon Graphics, SGI Onyxes com Reality Engines ou InfiniteReality Engines. Óculos passivos polarizados de baixo custo podem ser usados no lugar de óculos ativos usados no CAVE e no ImmersaDesk.

O IWall atinge sua imersão pela projeção na tela, mas não permite ao usuário olhar para baixo, o que é um problema com a disposição das cadeiras em um auditório normal. Está em estudo um tipo de tracking para grandes áreas, pois no momento só é possível o tracking de uma pessoa por vez (Cruz-Neira *et al.* , 1995).

- **Virtual Reality Exposure Therapy**, GVU Center - Georgia Institute of Technology .



(a) Terapia para medo de altura

(b) Elevador virtual

Figura 2.4: Terapia de Exposição com realidade virtual

"Terapia de exposição" usando realidade virtual envolve expor o paciente a um ambiente virtual que estimule sua fobia ao invés de expor o paciente a este estímulo em uma situação real, ou ter que fazê-lo imaginar este estímulo.

Uma das maiores vantagens desse tratamento é a facilidade para criar situações que correspondam à fobia do paciente, como por exemplo, tratar o medo de avião usando um avião virtual, o que representa ganho em termos de dinheiro e tempo. Outras vantagens são: evitar a exposição do paciente a um ambiente público, violando assim a privacidade do paciente; dar ao terapeuta maior controle sobre diversos parâmetros da análise; ter grande aceitação por parte do paciente. Estes estudos têm mostrado ser bastante efetivo na redução de fobias, ansiedades, medo de altura promovendo melhoria considerável nas atitudes dos pacientes (Hodges *et al.* , 1995).

Alguns projetos com objetivos semelhantes, para tratamento terapêutico com uso de realidade virtual, são desenvolvidos na University of Washington, HIT Lab.

São eles:
Mind over Matter–Virtual Reality and Pain Control (Hoffman, 1996)
Parkinson's Project (Weghorst & Prothero, 1994)
Phobia Desensitization in Virtual Environments (Hoffman, 1997).

- **The Conceptual Design Space**, Graphics, Visualization and Usability Center.

  É uma aplicação em tempo real, para criação de projetos de ambientes $3D$, de forma interativa. A primeira utilização destas técnicas é para arquitetura. Estudantes do Georgia Institute of Technology têm usado para criar projetos de edifícios. Os estudantes podem não somente inspecionar os prédios como também modificá-los, adicionar detalhes enquanto estão imersos no mundo virtual.

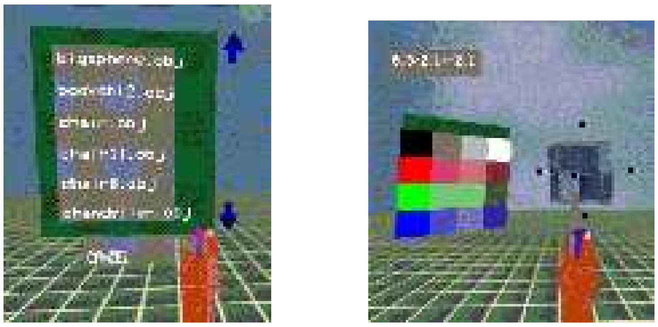


Figura 2.5: Ambiente de projeto virtual

  Usuários do Conceptual Design Space podem criar prédios simples de uma maneira intuitiva, simplesmente escolhendo vértices e adicionando a eles uma terceira dimensão, através da especificação de uma altura para cada vértice. As paredes e teto são criados automaticamente pelo Conceptual Design Space. Após criada a estrutura básica pode-se experimentar diferentes cores e texturas, adicionar móveis e mudar o telhado.

  No futuro, o Conceptual Design Space será capaz de permitir mudanças de luz interativamente, figuras humanas por escala, bibliotecas de objetos de arquitetura e texturas, assim como a capacidade de exportar seus modelos conceituais para pacotes de CAD, para posteriores refinamentos (Bowman, 1996).

- **The Responsive Workbench**, German National Research Center for Computer Science.

  O projeto quer transformar o conceito usual de diálogo na comunicação homem-máquina. Adaptar estações de trabalho multimídia para uma forma mais voltada para aplicações de uso da ciência, medicina e arquitetura. O dispositivo de exibição (display) é projetado para fazer parte do ambiente de trabalho humano. No momento, objetos $3D$ ainda são exibidos em mesas. O usuário interage com o cenário virtual, manipula-o como se fosse real, e quando solicita obtém informações do computador ao fundo.

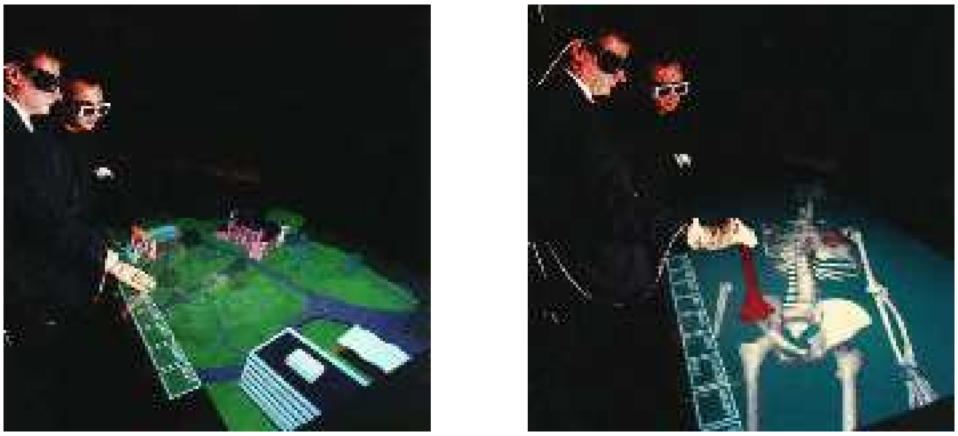


Figura 2.6: O computador é invisível integrado ao mundo do usuário

Neste novo conceito o usuário experimenta simulações do mundo no computador, mas o computador é invisível integrado ao mundo do usuário. No dia a dia, objetos e atividades se tornam entradas e saídas deste ambiente. Computadores são considerados como parte da vida diária e não ficam mais isolados em “desktops”. O sistema do computador pode-se adaptar para melhorar o ambiente de vida humana.

As seguintes aplicações têm sido envolvidas neste novo tipo de ambiente: treinamento médico não sequencial, planejamento cirúrgico, visualização de fluidos dinâmicos e modelagem molecular (Kruger *et al.* , 1995).

2.2 Ambientes Virtuais Colaborativos

O que caracteriza a área de Ambientes Virtuais Colaborativos é a interação do homem com um mundo virtual $3D$, o que coincide até aqui com a denominada área de realidade virtual, diferindo no contexto de coletividade. Aqui pretende-se integrar várias pessoas distantes fisicamente ao mesmo mundo artificial, querendo torná-lo o mais natural possível e vencendo a barreira da distância através da comunicação em rede.

Alguns dos projetos desenvolvidos nesta área são os seguintes:

- **The GreenSpace Project**, University of Washington.

  O projeto GreenSpace teve início em Abril de 1993 e continua até o momento, sendo desenvolvido no Human Interface Technology Laboratory em colaboração com o Fujitsu Research Institute (FRI) de Tokyo. Os objetivos do projeto são desenvolver e demonstrar uma mídia de comunicação imersiva onde participantes distantes usufruam de um sentimento de presença num ambiente virtual compartilhado.

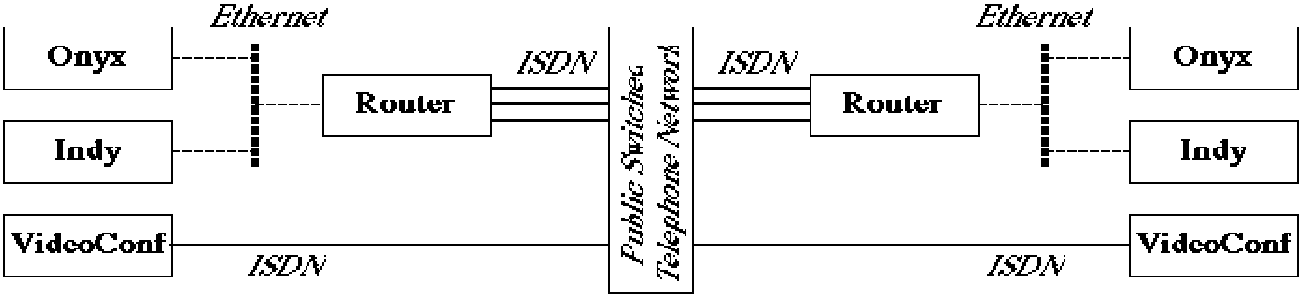


Figura 2.7: Configuração da rede

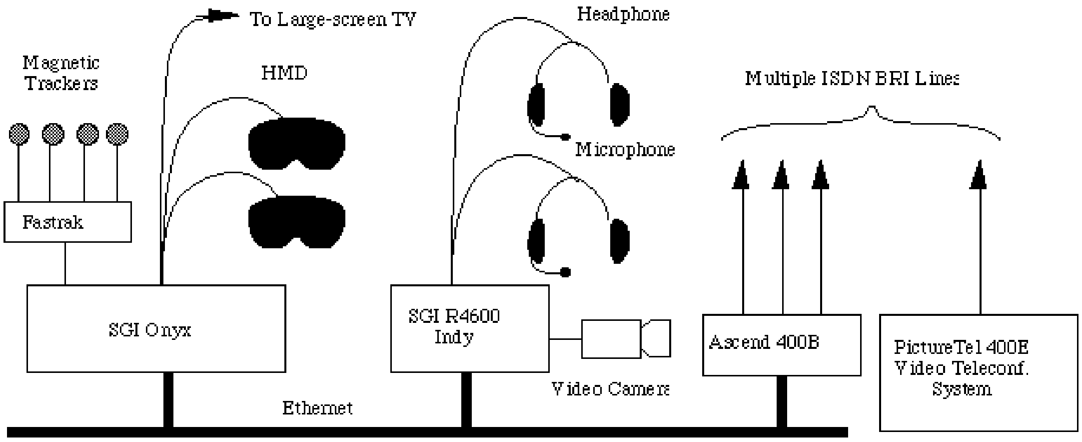


Figura 2.8: Configuração do ambiente

O projeto pretende promover a colaboração à distância entre 100 ou mais participantes em redes de banda larga tais como SONET/ATM, imersos num ambiente rico em acuidades visual, auditiva e tátil (Mandeville *et al.* , 1995).

- **I-WAY Project ou Information Wide Area Year**, Electronic Visualization Laboratory(EVL).

  Desde 1993 EVL vem trabalhando em resultados envolvendo o desenvolvimento de redes de banda larga e baixa latência, capazes de suportar dados suficientes para integrar supercomputadores e dispositivos de realidade virtual espalhados pelo país. Estes *links* entre supercomputadores, CAVEs e I-DESKs criam laboratórios virtuais, que maximizam o uso eficiente de recursos e permitem cientistas, designers, engenheiros, artistas e educadores trabalharem de maneira integrada em projetos a longa distância.

  Um dos principais objetivos do projeto I-WAY é promover o primeiro teste em aplicações de larga escala de IP/ATM (redes de alta velocidade) para supercomputadores e usar realidade virtual e supercomputadores juntos (DeFanti *et al.* , 1996).

- **The Virtual Society Project**, Sony Computer Science Laboratory Inc.

  O objetivo do projeto é investigar como a comunidade futura *online* irá evoluir. A equipe do projeto acredita que sistemas futuros online serão caracterizados por um alto grau de interação, hábeis para suportar multimídia e principalmente hábeis para suportar espaços $3D$ compartilhados. Usuários não terão forum de conversas baseados apenas em texto, mas irão adentrar ambientes $3D$, onde poderão interagir com esse mundo e com outros usuários dentro deste mundo.

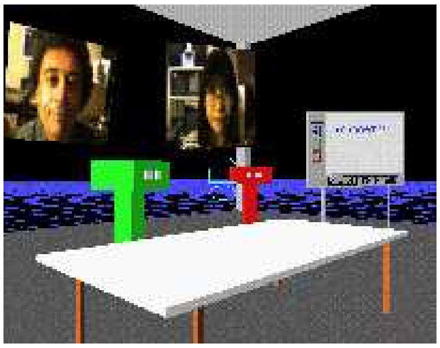


Figura 2.9: Ambiente de conferência compartilhado com suporte de vídeo ao vivo

O conceito do Virtual Society é simples e intuitivo: prover um ambiente $3D$ compartilhado no qual "modeladores de mundo" possam construir mundos $3D$ e popular esses mundos com serviços. Mais do que construir modelos $3D$ sem vida, eles querem construir mundos nos quais pessoas possam participar. Por exemplo, fazer compras ou ver um filme juntas mesmo estando em países distantes (Honda *et al.* , 1995).

2.3 Realidade Aumentada

Realidade Aumentada combina imagens geradas por computador, dispositivos de realidade virtual e imagens do mundo real, para enriquecer a cena final com informações adicionais. O usuário precisa ver simultaneamente o mundo real e as imagens geradas por computador. Vê-se portanto que as técnicas e equipamentos necessários não podem ser as mesmas para as demais áreas.

As pesquisas são realizadas em diferentes domínios. Por exemplo:

Área médica - é uma das áreas mais importantes de pesquisa. Estudos de imagens pré-operatórias do paciente dão ao cirurgião a visão necessária da anatomia interna do paciente. O técnico de ultrason pode ver uma imagem volumétrica renderizada do feto sobreposta ao abdomem da paciente, usando um dispositivo ótico; a imagem aparece como se estivesse no interior do abdomem e é corretamente renderizada a cada movimento da cabeça do médico.

Treinamento militar - as forças armadas têm usado displays nas cabines dos pilotos, para apresentar informações ao piloto. Estas informações são exibidas no pára-brisa da cabine ou na viseira do capacete de vôo. Esta técnica tem sido muito usada para treinamento de vôo em operações militares. Ao olhar o horizonte, por exemplo, o soldado equipado com este display poderia ver outro helicóptero voando acima da linha do horizonte e praticar estratégias de vôo (Metzger, 1993).

Projetos de Engenharia - alguns trabalhos vêm sendo desenvolvidos com o auxílio de realidade aumentada, com o objetivo de melhor estudar projetos ou acrescentar informações a estruturas já existentes. **Architectural Anatomy**, da Columbia University, é um exemplo de trabalho nesta área, e será apresentado adiante.

Robótica e Telerobótica - realiza experimentos para controle e análise de movimentos de robôs através da realidade aumentada. **Augmented Telerobotic Control** é um trabalho que descreve um sistema para controle de *telerobótica aumentada*, desenvolvido na University of Toronto. A imagem de vídeo do robô é sobreposta a uma imagem do modelo geométrico deste robô e com isso é possível fazer análises de precisão de movimentos, entre outros objetivos (Milgram *et al.* , 1995).

Manufatura e manutenção - a área de manutenção de equipamentos também tem sido apoiada com o uso de realidade aumentada, dando condições ao técnico de visualizar peças internas de um equipamento sem precisar desmontá-lo. Auxilia também no conhecimento de equipamentos dispensando o uso de extensos manuais.

Entretenimento - na literatura disponível os trabalhos referenciados como realidade aumentada aplicados a entretenimento são, segundo a análise apresentada, aplicações de cenários virtuais. Estes trabalhos misturam o conceito de composição de imagem com sobreposição de imagens, citando por exemplo a técnica de *bluescreen* como realidade aumentada. Também não destacam o fato de não haver imersão por parte do usuário destes sistemas. De acordo com o exposto no início deste capítulo estas aplicações não possuem as mesmas características técnicas, e portanto são referenciadas neste trabalho dentro de cenários virtuais.

Alguns projetos desenvolvidos nos diferentes domínios de aplicação da realidade aumentada são citados a seguir:

- **Project on Image Guided Surgery**, pelo MIT Artificial Intelligence Lab.

  Estão sendo desenvolvidas ferramentas para análise de imagens para permitir criar uma estrutura tridimensional detalhada e seus relacionamentos, a partir de imagens médicas. Tais ferramentas irão permitir aos cirurgiões visualizar estruturas internas através de uma sobreposição automatizada de reconstruções $3D$ de anatomia interna sobre imagens de vídeo ao vivo de um paciente.

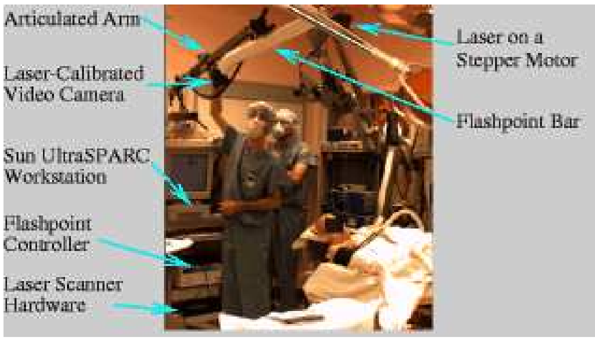


Figura 2.10: Sistema de navegação cirúrgica

Sem a visualização permitida pela realidade aumentada é difícil localizar uma anormalidade numa região, apenas olhando uma imagem de vídeo colorida. A superposição destas imagens claramente ilustra esta região de tratamento e permite que o cirurgião veja onde estão localizadas as estruturas internas relativa ao ponto de visão da câmera. (Ettinger *et al.* , 1997).

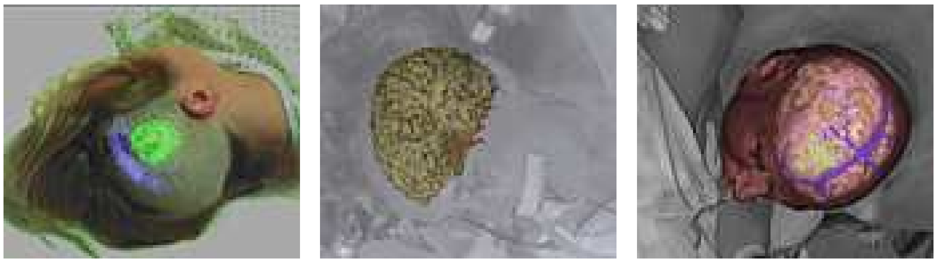


Figura 2.11: Visualização com realidade aumentada

- **Ultrasound Visualization Research**, University of North Carolina - Chapel Hill. O objetivo deste projeto é desenvolver e operar um sistema que permita a um médico ver diretamente dentro do paciente. São necessárias imagens ecográficas de ultrason, um head-mounted display (HMD), para ver atráves do vídeo, e um computador gráfico de alta performance para criar imagens que resultam da combinação das imagens de ultrason geradas por computador com imagens de vídeo, ao vivo, do paciente.

  Estes sistemas podem ser usados em obstetrícia, procedimento de diagnósticos (como biópsias), cardiologias, etc. Os estudos servem para aprofundar e melhorar as tecnologias de realidade aumentada tais como tracking, sistemas HMD e representação visual (Fuchs *et al.* , 1996).

Figura 2.12: Ultrasom com o uso de realidade aumentada

- **Architectural Anatomy**, Columbia University.

  É um projeto de cooperação entre o *Computer Graphics and User Interfaces Lab* e o *Building Technologies Group* da *Columbia's Graduate School of Architecture*. O protótipo do projeto gera a sobreposição de uma representação gráfica de partes do sistema estrutural do prédio sob a visão do usuário, na sala onde este se encontra. O mundo virtual sobreposto tipicamente mostra o contorno das estruturas de concreto, vigas e colunas da sala, permitindo ao usuário ver as estruturas embutidas nas paredes. O modelo é baseado nos desenhos estruturais fornecidos pelo pessoal de projeto.

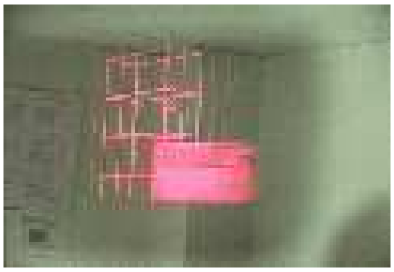


Figura 2.13: Armação de ferro interna ao pilar selecionado

O sistema é interativo. Quando o usuário escolhe uma coluna, esta é selecionada com o mouse, indicando que o usuário quer ver sua estrutura interna juntamente com a análise estrutural. Após a seleção, a coluna muda o estilo de linha e sua armação de ferro interna aparece, juntamente com uma janela X11 que contém a avaliação da análise estrutural desta coluna. O usuário deve usar um head-mounted display desenvolvido pelo próprio laboratório (Feiner *et al.* , 1995).

- **KARMA (Knowledge-based Augmented Reality for Maintenance Assistance)**, Columbia University Computer Graphics and User Interfaces Lab.

  É um sistema para explicar a manutenção de uma impressora laser e exige o uso de um head-mounted display pelo usuário. Vários trackers Logitech $3D$ são presos a componentes chaves da impressora, permitindo ao sistema monitorar suas posições e orientações. Por exemplo, para mostrar ao usuário determinada peça, o sistema determina se a peça está obstruída por outras. Se está, ele exibe uma imagem da peça bloqueada de maneira que aparente estar sendo vista através das outras peças. Se a peça já é visível, o sistema não precisa desenhá-la (Feiner *et al.* , 1993).

2.4 Cenários Virtuais

O alvo aqui é parecer que o mundo real e virtual são apenas um, embora quem está no mundo real não esteja interagindo com o mundo virtual.

Sistemas de cenários virtuais são desenvolvidos para dar aos projetistas de cenários e aos engenheiros de softwares a habilidade para criar ambientes $3D$ gerados por computador em tempo real, nos quais atores e câmeras possam mover-se livremente.

2.4.1 Sistemas que estão sendo comercializados

- **Orad's Virtual Set**, em Israel.

  Este sistema, baseado em reconhecimento de padrões, é capaz de fazer composição em tempo real entre uma imagem de vídeo e um background (imagem de fundo) sintético, implementado em um computador para processamento de imagens com capacidade de 30 GOPS (giga operations per second), desenvolvido pela empresa. Os parâmetros de câmera e lentes são extraídos em tempo real sem nenhum tipo de sensor, usando um background tipo *grid* (uma malha) com dois tons de azul (Orad, 1995).

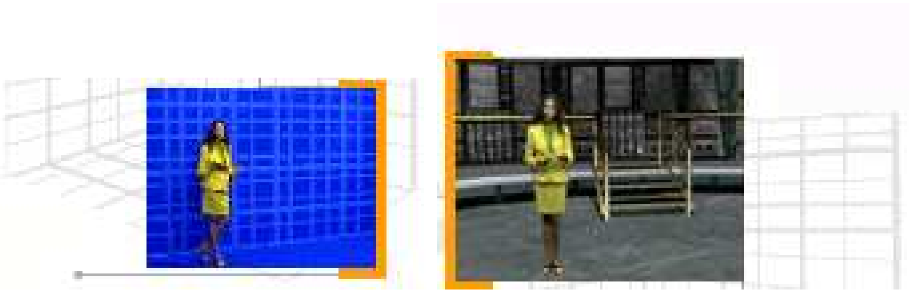


Figura 2.14: Orad's virtual set systems

- **Discreet Logic**, na Austrália.

  É um sistema composto de três módulos que desempenham funções específicas. FROST, VAPOUR, e GLASS oferecem uma solução para criação de cenários virtuais. FROST é um sistema de modelagem $3D$ e animação capaz de criar e animar elementos gráficos em tempo real. VAPOUR melhora as habilidades de FROST, proporcionando rendering rápido e realístico em tempo real para cenários virtuais $3D$, além da flexibilidade de combinar cenas ao vivo com ambientes de computação gráfica. GLASS proporciona controle e precisão das câmeras real e virtual, combinando dados de posição da câmera com a estação gráfica $3D$. Os dados de câmera gerados pelo GLASS podem ser usados em tempo real ou armazenados para pós-produção. O sistema opera em ambos os modos *online* e *offline* (Future-Reality, 1995).

- **RT-SET**.

  Oferece a opção de configurações para dois Sistemas de Estúdio Virtual(VSS) (RT-Set, 1994):

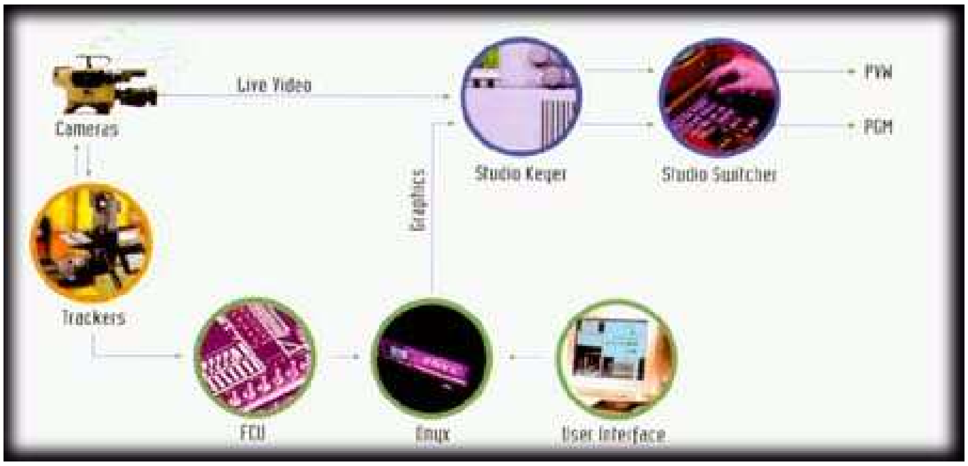


Figura 2.15: Larus - integra câmeras, lentes, switchers, tracking systems e outros equipamentos

  Larus - permite integração em tempo real de atores ao vivo com cenários virtuais $3D$ durante transmissão ao vivo. A integração instantânea com o ambiente do estúdio inclui câmeras, lentes, switchers, sistemas de tracking e outros equipamentos. O sistema suporta uma estação SGI Onyx InfiniteReality e hardware próprio incluindo *RT-SET's Fast Communications Unit* (FCU) que acelera a comunicação câmera-computador.

  Otus - permite integração de atores ao vivo com cenários virtuais $3D$ para transmissão de programas em vídeo-tape.

2.4.2 Projetos Desenvolvidos

- **Cenário Virtual e Composição de Imagens**, Projeto Visgraf - IMPA.

Figura 2.16: Composição de imagens com a mesma escala

Foi desenvolvido em conjunto com a Rede Globo de Televisão um sistema chamado *Sitema Memory Head* baseado em técnicas de robótica e processamento de imagens para a produção de efeitos especiais sofisticados.

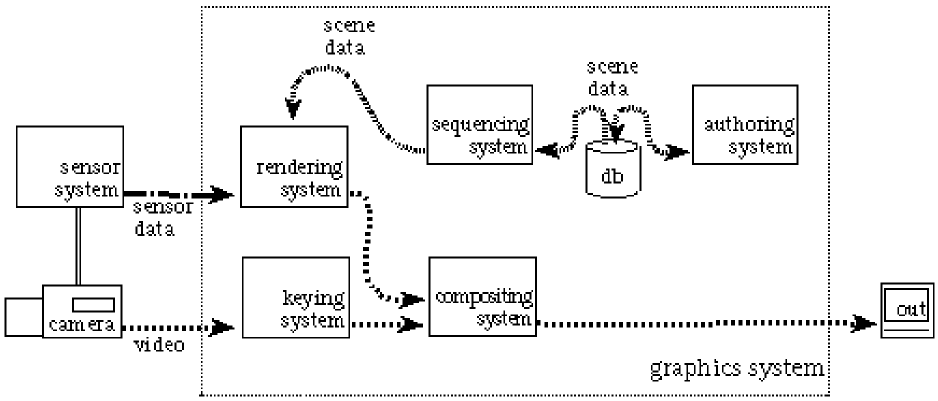
O objetivo principal é gerar uma única imagem final que reproduz o efeito desejado, obtida da composição de diversas imagens provenientes de câmeras controladas por computador. Esta composição é pós-processada, não sendo portanto uma aplicação em tempo real. Memory Head é um equipamento de *motion control* com o qual pode-se registrar os movimentos de pan, tilt, zoom e foco de uma câmera acoplada a um tripé. Dessa forma uma seqüência de movimentos da câmera é armazenada, podendo ser repetida posteriormente de forma precisa (Costa *et al.* , 1996).

- **Long-Range Camera Movement**, NHK S. and T. Research Laboratories - Japan.

Foi projetado um sistema capaz de operar uma câmera virtual sobre um campo de visão que é fisicamente impossível por sistemas convencionais, simulando um estúdio de tamanho infinito. Este efeito é produzido em tempo real, fazendo o efeito equivalente ao de um *Digital Video Processor* (DVP), em pós-processamento. Este laboratório desenvolveu alguns hardwares para desempenhar funções específicas neste sistema (Hayashi *et al.* , 1996).

- **Projeto Vist**, na Alemanha.

  Foi desenvolvido um sistema composto basicamente de três partes: uma câmera padrão para produções de televisão, um sistema de sensores para fazer o tracking dos parâmetros de câmera e um sistema gráfico que usa os parâmetros obtidos dos sensores para criar a correlação com o background. Este terceiro componente, o sistema gráfico, é composto de: um sistema de rendering, um sistema de composição, um sistema de sequenciamento, um sistema de autoria e um sistema de *keying*. É uma aplicação para operar em tempo real (Vist, 1994).



(a) Componentes do sistema

(b) Resultado final

Figura 2.17: The Virtual Studio (VIST)

- **Projeto Monalisa**, na Inglaterra.

  Monalisa - Modelling Natural Images for Synthesis and Animation, é um projeto de pesquisa cooperativo europeu que investiga o uso de imagem em computação gráfica e ambientes virtuais para transmissões de televisão em tempo real.

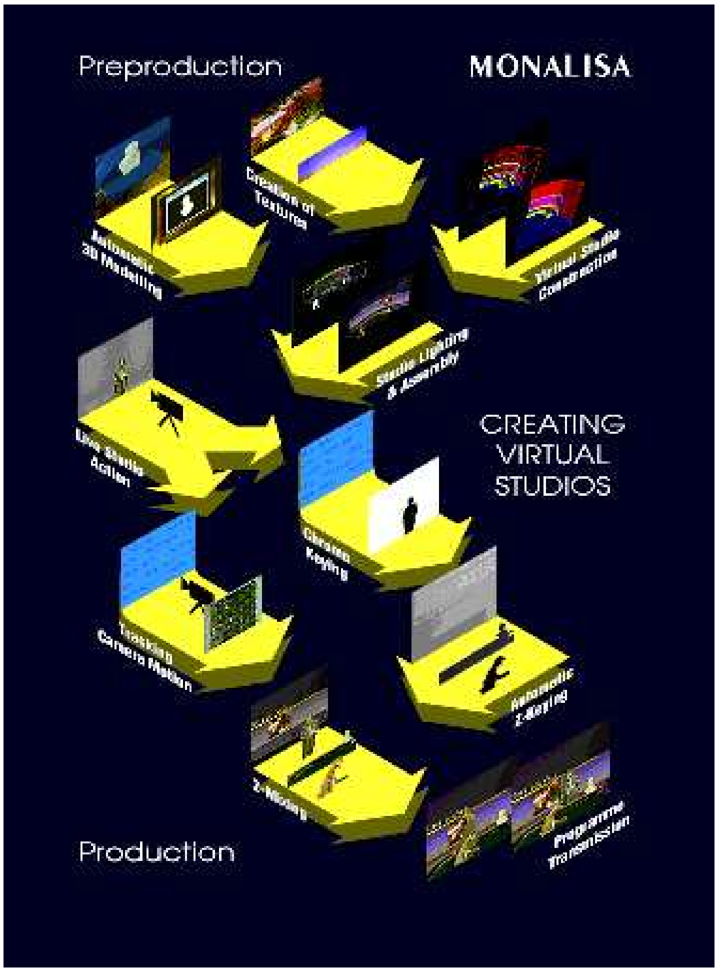


Figura 2.18: Partes integrantes do sistema Monalisa

É um sistema composto de vários módulos como rendering, modelagem, composição e tracking. Para este último foi desenvolvido um algoritmo de estimativa de movimento para medir movimento global da imagem de forma que a câmera possa mover sem a necessidade de sensores. Neste caso o background não pode ser de um azul uniforme como é convencional e deve conter informação necessária para a identificação do movimento (Monalisa, 1994).

- **Projeto Z-Key**, na Carnegie Mellon University.

  Z-Key é um método que combina imagem real com sintética, em tempo real. Usa informação de profundidade, pixel a pixel, a partir de um mapa de profundidade e compara alternadamente a profundidade das duas imagens em cada pixel, conectando a saída para a imagem mais próxima da câmera. Como resultado, tanto objetos reais como virtuais podem ocultar um ao outro (Kanade *et al.* , 1996).

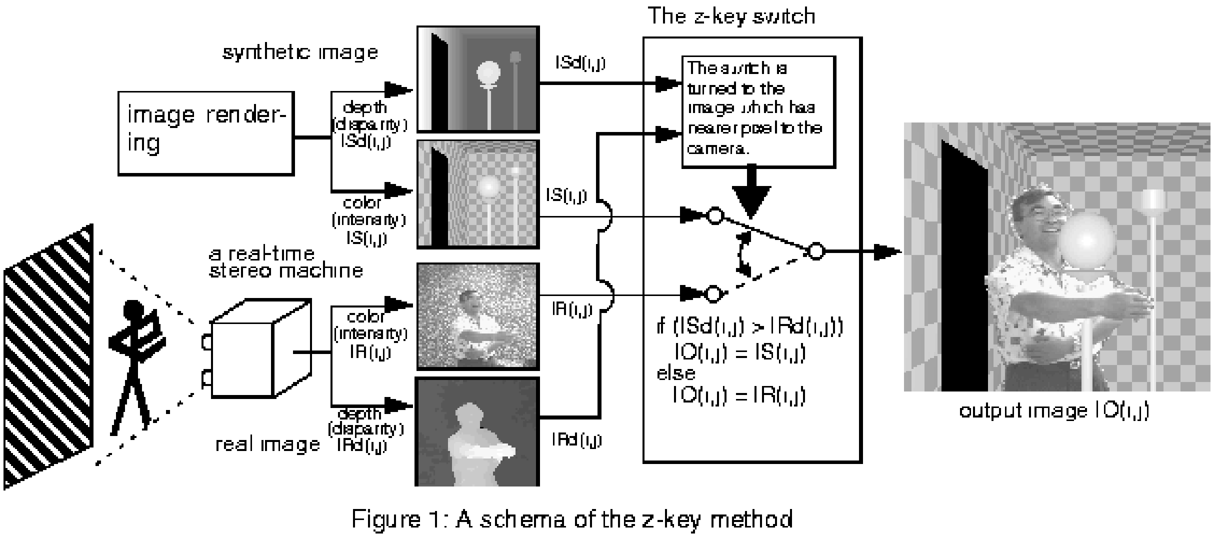


Figura 2.19: Z-key

- **Réalité Enrichie par Synthèse**, INRIA.

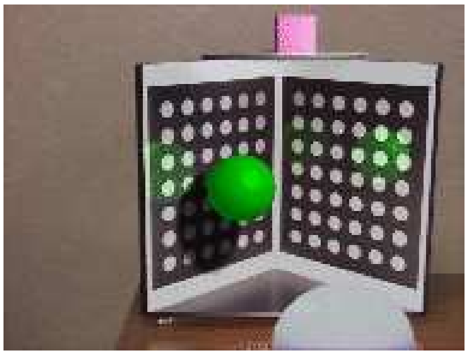


Figura 2.20: Inserção de objetos virtuais em uma sequência de vídeo real

Este projeto tem como alvo automatizar a combinação de objetos reais e sintéticos em uma mesma seqüência de vídeo animada, assegurando interações visuais e físicas entre os mundos real e virtual tais como: oclusão, sombra, colisões, etc.

A idéia principal é reconhecer e posicionar objetos $3D$ em imagens $2D$ para construir um tipo de máscara $3D$, que pode ser usada na composição com ambientes virtuais gerando uma cena mais complexa. Este trabalho não opera em tempo real (Jancène *et al.* , 1996).

Capítulo 3

Cenários Virtuais

A produção de efeitos especiais por computador tem contribuído muito com a evolução da indústria cinematográfica e de televisão, mas o reconhecimento de Cenários Virtuais como uma área da computação gráfica é um conceito novo.

Existem alguns softwares comerciais para este fim e poucos trabalhos na área acadêmica, consequentemente, pouca informação técnica disponível.

O consumidor direto destes sistemas são as empresas de televisão. A Rede Globo de Televisão tem usado constantemente esta tecnologia em suas transmissões, com um sistema adquirido da Orad's Virtual Sets. Um exemplo recente é a transmissão da Copa 98 direto da França usando um estúdio improvisado em uma sala e apresentando um cenário sofisticado e virtual (Figura 3.1).

Figura 3.1: Copa 98 - Rede Globo

No processo de cenários virtuais tem-se, de forma genérica, uma imagem filmada por uma câmera em um cenário real composta com uma imagem gerada por computação gráfica, para criar a imagem final.

Tradicionalmente os termos *foreground* e *background* eram usados para referenciar a imagem de frente e a imagem de fundo, respectivamente. Em cenários virtuais este conceito muda um pouco porque pretende-se "inserir" o ator em um ambiente virtual $3D$, portanto o ator pode ficar atrás de elementos virtuais do cenário. Os termos passam a ter então o seguinte significado: foreground - para a imagem real, filmada no estúdio e background - para o cenário sintético, admitindo-se sobreposições das duas imagens em qualquer ordem. A evolução das técnicas de combinar imagens será apresentada no Capítulo 5.

Existem, de uma maneira genérica, duas principais abordagens:

- produzir esta interação como uma pós-produção, podendo-se para isso aplicar diversas técnicas de composição para obter os efeitos desejados;
- produzir esta interação em tempo real, utilizando-se estes recursos para transmissões ao vivo.

O objetivo principal é tornar a interação foreground-background cada vez mais realista para o telespectador. Em ambas as abordagens citadas a captura dos parâmetros da câmera real constitui um sério problema e na maioria dos sistemas desenvolvidos utilizou-se sensores ou câmeras robotizadas. Estes equipamentos são de alto custo limitando, sob este aspecto, o uso destas técnicas.

Para produções em tempo real tem-se como principal desafio permitir que o ator possa mover-se livremente no cenário dando a certeza de que ele está imerso no ambiente virtual. Toda a ênfase neste trabalho será dada para as aplicações em tempo real.

Para resultados mais realistas é preciso controlar três tipos de movimentos de maneira sincronizada:

- movimento da câmera
- movimento dos atores
- movimento de elementos do cenário virtual

Conciliar estes três movimentos acarreta uma grande complexidade de estudos de caso para a geração de um sistema. Como as imagens são geradas de modo separado e depois combinadas, esses movimentos relativos causam sérios problemas de *registro* (alinhamento) para composição.

Conciliar ampla liberdade para cada elemento acima já não é trivial e envolve vários problemas ainda em aberto. Para os três elementos simultaneamente é muito difícil, e no entanto as soluções são indispensáveis de modo a não limitar a criatividade na direção de uma cena. Relacionados a esta nova performance surgem, portanto, problemas específicos a serem tratados.

3.1 Descrição dos sistemas

De uma maneira geral os métodos usados em sistemas de cenários virtuais podem ser esquematizados como mostra a Figura 3.2.

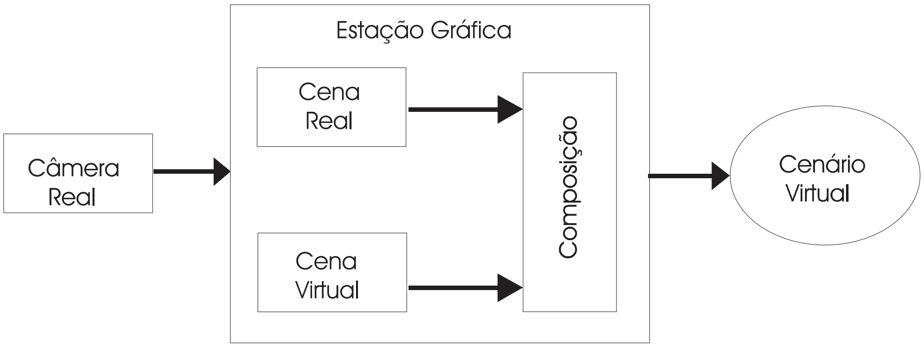


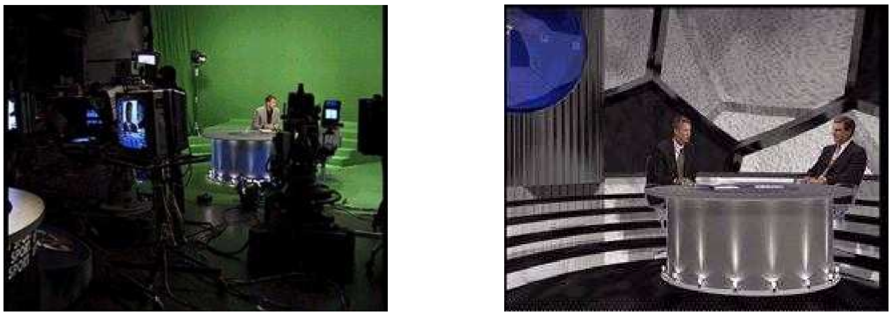
Figura 3.2: Esquema funcional de um sistema de cenografia virtual

Analisando a Figura 3.2 pode-se descrever o fluxo de dados e etapas do processo de construção destes sistemas.

Aqui deve-se considerar dois referenciais, um chamado *real*, onde se encontram os atores e a câmera (denominada câmera real), e outro chamado *virtual*, que é o mundo interno ao computador e que possui a sua câmera, denominada câmera virtual.

Tem-se, como problema central, conhecer a posição da câmera real (no estúdio) em cada instante, para que a câmera virtual possa ser posicionada no cenário sintético na posição correspondente. Isto exige muita precisão. Esta etapa é denominada *sincronismo de câmera* e será mais detalhadamente estudada no Capítulo 6. A cada momento da filmagem a posição da câmera precisa ser informada ao sistema, para que este possa calcular a posição da câmera virtual, e a partir daí gerar o cenário sintético. Isto chama-se *tracking* da câmera. A geração deste cenário no computador pertence a uma nova etapa denominada *rendering*, que permite gerar a cena virtual com aspecto realístico, iluminação adequada e em harmonia com os efeitos que se quer obter. Depois disso, as imagens, do ator e a sintética, precisam ser combinadas para gerar uma única imagem final, para parecer que tudo acontece no mesmo ambiente. Esta etapa denomina-se *composição de imagens*. Em muitos sistemas usa-se a técnica de *Chroma-key* para composição. Esta técnica é muito conhecida e praticada ao longo de anos em produções de televisão e será melhor definida no Capítulo 5.

A Figura 3.3 ilustra este resultado. A Radamec Broadcast Systems Ltd em U.K (Radamec, 1995) desenvolveu um sistema de estúdios virtuais para transmissão de televisão chamado *Virtual Scenario*, usando técnicas de chroma-key e sincronismo de câmera através de um *Sensor Head*, equipamento que faz o tracking mecânico de pan e tilt, para que a posição exata da câmera seja conhecida pelo sistema que processa o cenário sintético, a cada instante.



(a) Chroma Key Studio (BBC Television Centre)

(b) Football Focus

Figura 3.3: Sistema *Virtual Scenario*

3.1.1 Integração dos componentes

Com o objetivo comum, alguns sistemas têm sido desenvolvidos utilizando técnicas distintas para tratar os problemas, formando uma gama de soluções com diferentes habilidades e limitações para os sistemas. Na Seção 3.1 foram identificadas três principais etapas que compõem o processo de geração de um cenário virtual e requerem tratamento específico para suas características próprias. Mas ao integrar estas etapas surgem novas questões que precisam ser resolvidas:

Rendering

- **Modelagem**

  Como os cenários precisam ser renderizados em tempo real existe limitação para o número de polígonos na construção dos elementos de cena. Fatores como número de texturas permitidas, quais objetos serão animados e quais objetos ficarão estáticos, e para os animados quais as trajetórias, são decisões da fase de modelagem que influenciam em todo o processamento do cenário. As aplicações nas quais pretende-se incluir a projeção de um vídeo sobre um objeto virtual também requer previsões por parte da modelagem, pois este objeto precisa ter uma malha separada que não esteja conectada hierarquicamente a nenhum outro objeto; este tipo de aplicação está ilustrado na Figura 3.1.

- **Iluminação**

  Olhando a combinação de uma cena real com uma cena gerada por computador pode-se facilmente perceber uma discrepância na iluminação das duas cenas. Devido a este fato as luzes do cenário virtual precisam coincidir em tipo e posicionamento com as luzes do cenário real. As posições das fontes de luz sintética e real devem estar em perfeito sincronismo, isto é, as luzes virtuais devem estar na mesma posição da iluminação no mundo real.

  As sombras são consequentemente um outro problema, principalmente para conseguir colocar as sombras dos atores (real) combinadas com a cena virtual.

- **Colorização**

  A cor de um objeto em uma cena depende de vários fatores ao longo da apresentação de um programa, porém dois aspectos fundamentais devem ser levados em conta para se ter sucesso com as cores: objetos virtuais e objetos reais devem ter suas cores constantes, e fundamentalmente é preciso usar o mesmo sistema de representação de cor, desde a etapa de projeto até a transmissão do programa, para evitar distorções cromáticas quando vários *displays* diferentes são utilizados. Para isto deve-se estabelecer um modelo de cor independente do dispositivo, como o CIE-LAB, por exemplo (Oschatz, 1994).

  Quanto à constância das cores, este controle torna-se problemático sempre que os objetos precisarem ter suas cores recalculadas devido a mudanças nas fontes de luz. Compreender como empregar corretamente cores e texturas são pontos chaves para obter um alto nível de realismo em um cenário virtual.

- **Formatos de vídeo**

  Os sistemas de codificação de vídeo PAL e NTSC podem representar apenas um subconjunto de todas as cores. Então novamente, ainda na fase de projeto a seleção do sistema de cor deve ser feita em concordância com o sinal de codificação PAL ou NTSC para permitir uma filtragem apropriada, e ser capaz de evitar modificações de cor fora de controle durante uma transmissão (Oschatz, 1994).

Composição

- **Oclusão**

  Oclusão é quando um objeto está encobrindo parcialmente o outro. Em cenários virtuais, a oclusão tanto de atores reais como de objetos virtuais é uma questão importante. Um sistema que quer posicionar objetos virtuais à frente e atrás de atores reais, ao mesmo tempo, não pode ter um tratamento simples de composição. Um exemplo deste tipo de oclusão ocorre quando um ator está atrás de uma mesa virtual e na frente de um cenário sintético, ao mesmo tempo.

Encontram-se na literatura diferentes formas de tratar este problema. A Universidade de Carolina do Norte, Chapell Hill (UNC) desenvolveu um sistema chamado *Pixel Flow* (Molnar *et al.* , 1992) que oferece uma arquitetura de hardware para gerar uma cena inteira em tempo real usando técnicas de composição de imagem. O sistema consegue agilizar os resultados de rendering para tempo real e ainda tratar o problema de oclusão. Outra forma de tratar oclusão é através da informação de profundidade da cena, que será discutida no Capítulo 5.

- **Distorção**

  A imagem obtida da câmera real sempre tem distorção introduzida pelas lentes da câmera. A imagem sintética não tem distorção porque o rendering é calculado de forma a minimizar o erro e a distorção para que a imagem tenha alta qualidade, isso faz com que fique bastante perceptível a diferença entre as duas imagens. Uma solução consiste em introduzir distorção na imagem gerada por computador para que esta pareça mais realista e perca um pouco da aparência "excessivamente perfeita" que ajuda a detectar que trata-se de uma imagem sintética.

Sincronismo

- **Delay entre câmera e imagem sintética**

  Um dos grandes problemas na visualização final destes sistemas é a defasagem temporal entre a imagem real e a imagem gerada por computador. A tomada de cena muda com o movimento de câmera, entretanto a cena sintética muda mais ou menos simultaneamente, pois um certo intervalo de tempo é necessário para que a cena sintética seja calculada. Esta defasagem quebra a impressão de fusão entre os cenários real e virtual. Vários fatores são responsáveis por isto e alguns deles dependem do sincronismo de câmera.

  Os sistemas que utilizam sensores nas câmeras consomem algum tempo entre capturar uma certa posição do sensor até o inicio da transmissão destes dados. Este tempo é da ordem de milisegundos e pode ser previamente conhecido de acordo com a marca do sensor.

  A transferência dos dados do *tracker* para o computador de controle também demanda algum tempo. Usualmente os sistemas de tracking são conectados ao computador de controle por interfaces seriais. Além disso existe um intervalo de tempo entre a chegada completa dos dados de tracking ao computador e a saída da imagem sintética.

  Então toda defasagem está relacionada à filtragem dos dados, esforço de gerenciamento destes dados, chamadas ao sistema operacional, atualização da cena virtual e consequentemente o rendering desta cena. O tempo de rendering depende diretamente da complexidade da cena, da qualidade desejada e do sistema de computador utilizados.

Um último fator de defasagem entre as imagens real e sintética deve-se ao tempo entre o término da geração da imagem sintética até a exibição desta imagem na tela de vídeo. Por exemplo, o sistema PAL exibe 25 imagens por segundo. Cada imagem consiste de 576 linhas, que são mostradas em dois campos de 288 linhas cada. Até a exibição do último pixel na tela são necessários aproximadamente 40 milisegundos (Oschatz, 1994).

- **Corte**

  Com muita frequência, várias câmeras são utilizadas na produção de um programa, havendo alternância entre as diversas câmeras. Nestes casos torna-se mais difícil gerar o cenário virtual, pois este deve alternar suas cenas em sincronismo com a cena real. E ainda levar em conta os diferentes tipos de transição como, por exemplo, *corte seco* ou *cross-fade*.

3.1.2 Benefícios da cenografia virtual

Flexibilidade

Com cenários virtuais pode-se em poucos minutos mudar o cenário de um programa de entrevistas para um noticiário. Considerando que cenários reais são trabalhosos para construir, transportar e armazenar, cenários virtuais podem ser armazenados em unidades de disco, e permitir mudanças rápidas. Possibilita criar ambientes que poderiam ser extremamente caros se construídos. À medida que os hardwares evoluem e também os softwares, estes ambientes vão oferecer maiores vantagens. Limitações devidas a número de polígonos e texturas são constantemente melhoradas pelo avanço da tecnologia. A composição em tempo real permite que o ator se locomova para diferentes lugares em cena e a ilusão torna-se forte e convincente. Uma vez construído o modelo $3D$ do cenário, este fica disponível para usá-lo de diferentes formas, mudar a cor da parede e o tapete, por exemplo, ao vivo. Toda esta flexibilidade permite um grande número de efeitos especiais.

Custo x Espaço

O custo de investimento inicial é alto podendo ficar em torno U$ 1,5 milhões de dólares (Siegmann, 1994). Mas o espaço físico requerido para a produção com cenário virtual é um espaço mínimo para a filmagem do ator; vários shows diferentes podem ser filmados usando um mesmo espaço físico pequeno. O uso destas técnicas permite ampliar o ambiente dando a ilusão de ser muito maior, além disso a mudança em um estilo de cenário requer alguns minutos de computador. Portanto, ao longo do tempo, isto representa uma economia significativa de custos.

3.1.3 Hardware

As três principais etapas identificadas em um sistema de cenários virtuais formam três subsistemas: Sincronismo, Rendering e Composição; torna-se indispensável que o hardware tenha habilidades específicas para atender esta performance em tempo real.

Requisistos necessários (Gibbs & Baudisch, 1996):

- alta precisão para o tracking;
- mapeamento de textura em tempo real;
- taxas de rendering compatíveis com a taxa de operação de vídeos;
- texturas de alta resolução;
- suporte para múltiplas câmeras;
- sistema de tracking com interface para várias câmeras;

Os computadores que vêm sendo utilizados pelos produtores destes sistemas são da linha *Silicon Graphics Onyx2 InfiniteReality2*, que possuem várias configurações diferentes (Graphics, 1998).

O computador **Onyx2^{TM} InfiniteReality2^{TM} and InfiniteReality Multirack System** é o mais sofisticado e possui as seguintes características:

Figura 3.4: Onyx2^{TM} InfiniteReality2^{TM} and InfiniteReality Multirack System

- 256MB a 256GB de memória RAM
- acima de 2.3TB de capacidade de armazenamento de disco interno

- RealityMonster^{TM}, um multisubsistema de rendering com capacidade de processamento superior a 210 milhões de polígonos por segundo e 7.2 gigapixels por segundo ou 1GB de memória para textura.
- integra áudio e CD-ROM padrão
- acima de 144 I/O slots (opcional PCI)
- monitor de 24" com resolução de 1920x1200.

Outro modelo potente, mas um pouco menos sofisticado, é o **Onyx2 InfiniteReality2 and InfiniteReality Deskside System**, com:

Figura 3.5: Onyx2 InfiniteReality2 and InfiniteReality Deskside System

- 80MB ou 160MB de frame buffer
- 256MB a 8GB de memória RAM
- acima de 90GB de capacidade de armazenamento de disco interno
- capacidade de processamento superior a 13 milhões de polígonos por segundo
- integra áudio e CD-ROM padrão
- 4 I/O slots (opcional PCI)
- monitor de 24" com resolução de 1920x1200.

Tem-se então todo um conjunto de problemas complexos integrados, tornando a realização de um sistema de cenários virtuais uma tarefa multidiciplinar.
Os próximos capítulos abordarão técnicas e alternativas para os três principais subsistemas que compõem os sistemas de cenários virtuais.

Capítulo 4

Rendering

O processo de criar imagens por computador pode ser separado em duas etapas: *Modelagem* e *Rendering*. Modelagem é a descrição tridimensional da cena para o sistema; depois de criado o modelo $3D$, a cena pode ser visualizada de diversos pontos diferentes pelo processo de rendering, variando-se a posição do observador, que é a câmera virtual. O resultado do rendering é uma imagem bidimensional obtida a partir de uma vista de um modelo tridimensional, e equivale a tirar uma foto da cena tridimensional.

4.1 Projetando Cenários Virtuais

Os cenários não podem parecer desenhos ou imagens geradas por computador, é necessário convencer a audiência. Toda a tecnologia envolvida no conhecimento de rendering se faz necessária para a obtenção de um resultado satisfatório. As técnicas de cor, iluminação e texturas são fundamentais para a utilização destes recursos de maneira adequada sem sobrecarregar a cena desnecessariamente, além disso existe a limitação imposta pela modelagem, pois o fator tempo de processamento é fundamental. Para transmissões ao vivo o cenário precisa ser renderizado em tempo real, exigindo uma maior performance do sistema e ao mesmo tempo estabelecendo um duelo entre qualidade e eficiência, que sempre foi um ponto de fronteira nas decisões dos processos de rendering.

Portanto, dois critérios são imprescindíveis para um sistema de rendering:

- **Fotorealismo**

  A habilidade de sintetizar imagens geradas por computador com qualidade comparável às imagens capturadas por uma câmera é chamada de fotorealismo. Em algumas áreas da computação gráfica o fotorealismo não é prioridade, pois estas estão interessadas no modelo físico, como CAD para peças mecânicas por exemplo, mas no caso de cenários virtuais o sucesso depende de um efeito visual realístico e convincente. A necessidade de pisos com mármores ou madeiras, paredes com pinturas especiais, luzes sofisticadas, móveis cromados, efeitos de refletores fazem com que o fotorealismo seja prioritário para os cenários. Mas este resultado requer sofisticação em ambos, modelagem e rendering.

(a) (b)

Figura 4.1: Cenário virtual com rendering fotorealístico

A Figura 4.1 mostra uma imagem fotorealística usada para cenário virtual, gerada por (Wright, 1997).

- **Tempo Real**

  O Rendering para cenário virtual, além de envolver todos os problemas inerentes ao processo convencional, acrescenta o fato de que, na maioria das vezes, as imagens devem ser geradas em tempo real. Chama-se tempo real ao rendering de sucessivas imagens, dentro de uma frequência tal que o olho humano perceba como uma animação (Oschatz, 1994). No caso de cenografia virtual, o cenário deve ser renderizado a cada movimento da câmera real, dos atores, ou de objetos da cena. Uma sequência de imagens renderizadas, onde cada imagem tem suaves diferenças em relação a anterior, produz uma animação. A animação requer em média 30 quadros (imagens renderizadas) por segundo, daí a importância do tempo no processo de rendering.

4.2 Model-based rendering

O cenário virtual passa inicialmente pelo processo de modelagem para gerar o modelo $3D$, a iluminação deve ser projetada como parte deste modelo e de forma compatível com a iluminação do estúdio de filmagem. Após a geração do modelo $3D$, o cenário é renderizado utilizando-se um algoritmo de rendering (modelo de iluminação). Esta etapa pode ser usada para visualizar o cenário e fazer planejamentos da filmagem; mas em um processo de transmissão ao vivo, o cenário precisa receber o rendering em tempo real e em sincronismo com a câmera real. Para cenários gerados a partir de modelos $3D$ não existem limitações no movimento da câmera real devido ao modelo e sim pelas restrições de controle do sistema (ver Capítulo 6).

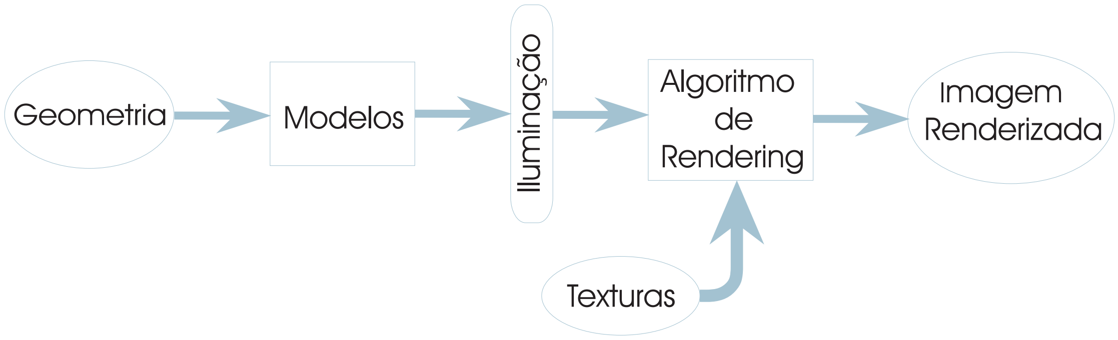


Figura 4.2: Model-based Rendering

A Figura 4.2 ilustra este processo de rendering que é o mais clássico da computação gráfica e também denominado *Model-based Rendering*.

4.2.1 Iluminação e Colorização

O termo colorização ou cor é empregado no sentido da palavra *Shading*, em inglês. Shading é uma terminologia que envolve o processo de calcular a cor de um ponto em uma superfície e requer o conhecimento das fontes de luz, propriedades dos materiais especificados para os objetos da cena e da posição dos objetos em relação às fontes de luz.

A função do cálculo da visibilidade no sistema é dar a ilusão de que os objetos mais próximos à câmera podem encobrir os objetos mais distantes e também passar a noção de profundidade para o olho humano. Além disso, o cálculo da visibilidade pode também ser usado para calcular sombras, pois as partes da cena que estão na sombra são as que não são visíveis a partir de uma fonte de luz. O cálculo das sombras, que é de responsabilidade do algoritmo de rendering, é de extrema importância por dois aspectos: tanto para que sejam geradas simultaneamente com o cenário em tempo real como também para que possam ser combinadas de forma coerente com as sombras dos atores na cena real. Daí ser indispensável a compatibilidade na iluminação dos dois ambientes (real e virtual), pois uma transmissão sem sombras ou onde as sombras são conflitantes por estarem em direções opostas geraria um resultado desastroso invalidando toda técnica usada e qualquer performance dos algoritmos.

A forma de calcular a luz difere de algoritmo para algoritmo e simular o modelo físico de iluminação em cenas geradas por computador é um processo elaborado e caro computacionalmente. Isto acarreta complexidade na modelagem da cena e exige mais técnica do algoritmo de rendering para reproduzir o mesmo comportamento da iluminação real. Os algoritmos mais completos e os mais usados para rendering de cenas complexas são *Raytracing* e *Radiosidade*. Raytracing intercala o cálculo da superfície visível e o cálculo da cor para mostrar sombras, reflexão e refração; é um algoritmo dependente da posição do observador: discretiza o plano de visão para determinar pontos visíveis em função de raios que partem do observador. Radiosidade é independente da posição do observador e discretiza o ambiente para avaliar a iluminação em qualquer ponto, a partir de qualquer direção; este algoritmo separa completamente a determinação da superfície visível do cálculo da cor (Foley *et al.* , 1992).

Técnicas de textura

Textura é uma técnica muito importante na área de rendering para dar aspectos reais a objetos, permitir a simplificação da modelagem dos mesmos quando a geometria do modelo não é indispensável à aplicação, e acrescentar realismo à cena.

Um valor de textura pode ser usado para modificar a cor, capacidade de reflexão, transparência ou outras propriedades de uma superfície que estejam relacionadas com seu processo de *shading* (Crow, 1989).

Diferentes técnicas de síntese de texturas são apropriadas para diferentes níveis de detalhes. Dentre elas, pode-se destacar as realizadas no domínio tridimensional, produzidas por funções do espaço tridimensional $f(x, y, z) : R^3 \to C$, sendo C um espaço de cor, e usadas para calcular a intensidade de cada pixel que irá representar a textura do objeto, dadas as coordenadas (x, y, z) do objeto no ponto que está sendo analisado. Esta técnica é denominada genericamente de textura sólida. Valores de texturas podem também ser determinados por funções do espaço bidimensional $f(x, y) : R^2 \to C$, processo que consiste em levar uma imagem I do domínio do R^2 em uma superfície S do R^3 através de uma transformação $T : I \to S \subset R^3$. Esta técnica chamada mapeamento de textura equivale a colar uma imagem sobre uma superfície.

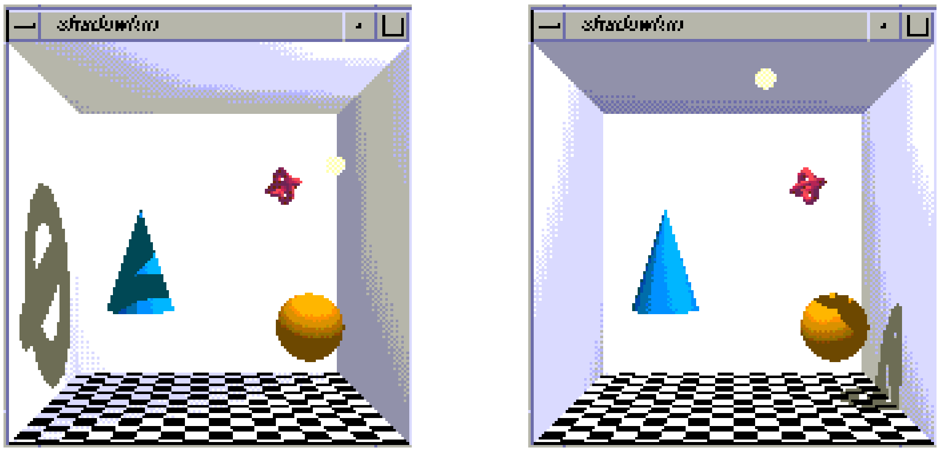
Imagens fotográficas são uma excelente fonte de funções $2D$ para gerar texturas, e isto tem motivado muitos estudos de métodos para mapear texturas $2D$ em superfícies. Uma abordagem introdutória em texturas pode ser vista em (Albuquerque, 1998). Em cenografia virtual este recurso é largamente empregado e os equipamentos mais potentes têm implementado esta técnica em hardware para permitir a geração de textura em tempo real.

Em (Bourke, 1992) pode ser visto um conjunto de imagens que ilustram as várias técnicas de rendering usando diferentes modelos de iluminação e texturas.

4.2.2 Sistemas comerciais para Visualização

OpenGL

OpenGL, desenvolvido pela Silicon Graphics, Inc. em 1992, foi o primeiro ambiente portável para o desenvolvimento de aplicações gráficas $2D$ e $3D$. Foi implementado como um *toolkit* para ser usado em cima de aplicações em linguagem C. Consiste de um conjunto de várias funções e procedimentos que permitem ao programador especificar objetos e operações para a produção de imagens gráficas com alta qualidade (Silicon Graphics, 1997a).



(a) (b)

Figura 4.3: Sombras geradas em tempo real pelo OpenGL

OpenGL permite renderings muito rápidos, facilitando aplicações em tempo real. Um exemplo de sombras geradas em tempo real, Figura 4.3, pode ser visto em (Silicon Graphics, 1997b). O modelo de iluminação utilizado neste processo é o de *Gouraud*, que não atinge o realismo fotográfico mas permite utilizar textura em tempo real, resultado muito importante para o uso de novas técnicas (ver Seção 4.4).

OpenInventor

OpenInventor, também desenvolvido pela Silicon Graphics, Inc., é um toolkit $3D$ orientado a objeto que contém bancos de dados de modelos para cenas tridimensionais facilitando o trabalho de programação em computação gráfica. Dispõe de uma biblioteca de objetos tais como: cubos, polígonos, textos, materiais, câmeras, luzes, trackballs, boxes, visualizadores $3D$, e editores para acelerar o tempo de programacão e ampliar as habilidades do programa.

Este toolkit funciona em cima do OpenGL, define um formato de arquivo padrão para dados $3D$, funciona com sistemas de janela e é independente da plataforma (Silicon Graphics, 1994/95).

RenderMan

RenderMan é uma ferramenta para rendering, desenvolvida pela Pixar, muito usada por estúdios de televisão para criar efeitos visuais realísticos.

O *RenderMan Interface* é uma interface padrão entre programas de modelagem e programas de rendering capaz de produzir imagens de alta qualidade, permitindo que as informações necessárias para especificar uma imagem fotorealística possam ser passadas para diferentes programas de rendering de maneira compacta e eficiente. Os diversos sistemas de rendering que fazem parte desta interface, dentre outros, são: *z-buffer-based*, *scanline-based*, *raytracing*, rendering de terreno. A interface é projetada para sistemas de rendering interativos, tanto em *batch* como em tempo real. O rendering em tempo real exige que toda informação necessária para desenhar uma primitiva geométrica, em particular, esteja disponível quando a primitiva for definida.

Outro módulo importante é o *RenderMan Shading Language*, que é uma linguagem de programação com a qual novos materiais e fontes de luz podem ser criados. É também usada para especificar deformações, projeções especiais de câmera e funções simples para processamento de imagem. A Shading Language é projetada para trabalhar com qualquer algoritmo de rendering, incluindo scanline, z-buffer, ray-tracing e radiosidade (Studios, 1997).

4.3 Tempo Real

Os requisitos para rendering em tempo real não dependem apenas da capacidade de processamento da máquina, mas também de algoritmos eficientes. Na realidade existem pesquisas em duas direções: para desenvolver novas arquiteturas em hardware e para reduzir as taxas de processamento dos algoritmos.

4.3.1 Buscando eficiência em hardware

Silicon Graphics Inc. está constantemente desenvolvendo equipamentos para melhorar a performance do processamento em tempo real e a qualidade das imagens produzidas. **Onyx2 InfiniteReality** é o equipamento mais utilizado para cenários virtuais, possui um sistema gráfico chamado InfiniteReality que é formado por três a seis placas conectadas à CPU do Onyx2; quando uma aplicação está sendo executada a CPU transfere os comandos gráficos para o sistema InfiniteReality liberando a CPU para outro processamento. O InfiniteReality tem a seguinte estrutura (Figura 4.4): o *Geometry Engine* faz transformações geométricas, cálculo de iluminação e funções de processamento de imagem; o *Raster Manager* faz as operações com pixels, cor e transparência e ainda armazena texturas; o *Digital Manager* converte uma imagem digital em um sinal de vídeo analógico.

Figura 4.4: InfiniteReality Pipeline

Este sistema gráfico é capaz de gerar mais de 11 milhões de polígonos por segundo e 896 milhões de pixels por segundo, permite trabalhar com múltiplos monitores de alta resolução e suporta resoluções de cores altíssimas para operações de rendering com textura (imagens) usando luminância. Pela qualidade gerada e capacidade de processamento, este sistema vem sendo considerado o sistema gráfico de melhor performance (Dare, 1998).

Outras pesquisas voltadas para este enfoque:

- **Pixel flow: High-speed rendering**

  Pixel Flow é uma arquitetura para acelerar a geração de imagens e tentar superar a arquitetura de hardware para rendering convencional. Este trabalho usa a técnica de composição de imagens, distribuindo a tarefa do rendering em vários conjuntos, onde cada um vai calcular uma imagem completa, mas contendo apenas uma fração das primitivas da cena. Uma rede de composição de imagens de alta performance compõe estas imagens em tempo real para produzir a imagem final da cena. As vantagens citadas são: taxa de crescimento linear com um modelo simples de programação e sub-imagens sendo calculadas de forma independente proporcionando um paralelismo natural transparente ao programador (Molnar *et al.* , 1992).

- **Just-in-time pixels**

  Baseado no fato das telas de computador exibirem as imagens de forma sequencial, existindo um tempo de espera para a exibição de cada pixel, os valores gerados para os pixels já não correspondem à posição real dos objetos no mundo no momento de sua exibição, desde que a posição e orientação da câmera e a posição do objeto no mundo não se mantenham fixos durante o período de tempo requerido pelo display. E este fato é comumente ignorado nas animações geradas por computador. O objetivo deste trabalho é conseguir exibir os pixels na tela de maneira que eles correspondam exatamente à posição e orientação verdadeiras da câmera virtual e à posição de todos os objetos no referencial mundo, no momento da exibição dos pixels. O sistema foi desenvolvido com a idéia inicial de usá-lo com um *Head-Mounted Display* para reduzir os erros de registro entre objetos virtuais e o mundo real. Just-in-time pixels em tempo real requer o conhecimento de quando um pixel será exibido e para onde o usuário estará olhando no tempo, isto implica na execução simultânea de outras funções: sincronismo da imagem gerada com a imagem exibida e determinação da posição e orientação da cabeça do usuário no momento da exibição de cada pixel (Mine & Bishop, 1993). Em cenários virtuais esta segunda condição corresponderia a conhecer a posição e orientação da câmera real no estúdio.

4.3.2 Buscando eficiência em software

Segundo (Dévai, 1997) rendering em tempo real vai além dos limites da máquina e tem como requisitos básicos as transformações geométricas, pré-seleção de parte da cena a ser exibida - *clipping* - e cálculo da visibilidade, e apresenta a seguinte análise: o tempo de processamento das transformações geométricas é linearmente proporcional ao número total N de arestas do modelo; a taxa de crescimento do clipping é de $N \log N$ no pior caso, considerando-se o método tradicional de clipping de polígonos, onde N é o total de arestas na cena; o cálculo da visibilidade apresenta uma taxa quadrática de crescimento. Portanto o cálculo da visibilidade é considerado por Dévai o *bottleneck* do rendering em tempo real, tornando-se mais sério quanto mais complexa for a cena.

Dévai faz em seu trabalho um estudo de eficiência das etapas do rendering, apresentando detalhada análise de complexidade dos algoritmos. Outros trabalhos com o objetivo de aumentar a eficiência do rendering são apresentados a seguir:

- **Overview of Parallel Photo-realistic Graphics**

  Muitas pesquisas são direcionadas para acelerar e obter resultados em tempo real usando os algoritmos de Raytracing e Radiosidade, uma das propostas é o *processamento paralelo*. O objetivo é definir quais as tarefas que são independentes e executá-las em diferentes processadores. O algoritmo raytracing calcula um pixel de forma independente de outro pixel e portanto muito propício ao processamento paralelo por este aspecto, mas em contrapartida gera problemas na distribuição dos dados. Em oposição ao raytracing, o algoritmo de radiosidade tende a ser problemático nos dois aspectos - distribuição e interdependência dos dados pois o cálculo da visibilidade de um *patch* depende do cálculo de outros *patchs*. Uma análise detalhada desta abordagem pode ser encontrada em (Reinhard *et al.* , 1998) onde vários algoritmos com o mesmo objetivo são apresentados e discutidos.

- **Temporal Coherence Using Predictive Rendering**

  Este trabalho introduz um novo método para reduzir o custo do processo de rendering baseado na previsão de movimentos na cena, usando simples transformações. As primitivas da cena são agrupadas em nós em uma estrutura de árvore, estes nós são renderizados com diferentes taxas baseados em previsões de movimento na cena. Os nós com movimento lento ou primitivas estáticas são renderizados menos vezes que os nós que contém primitivas com movimentos rápidos. Cada porção da cena é finalmente combinada para formar o frame final. Desta forma *Predictive Rendering* só redesenha as primitivas que sofreram movimento em cada frame, melhorando a velocidade do rendering (Fearing, 1996b).

- **Importance Ordering for Real-Time Depth of Field**

Figura 4.5: Depth of Field

Profundidade de campo - *Depth of Field* - é um efeito que acrescenta realismo às imagens sintéticas, desempenha um papel importante na credibilidade da imagem mas exige muito esforço computacional. É um dado importante também para outras áreas como visão computacional e realidade aumentada. Fearing apresenta um algoritmo cuja peculiaridade é reconhecer que existe semelhanças entre frames sequenciais de uma animação e usar esta continuidade para evitar de recalcular áreas da cena que não mudaram entre os frames. Os pixels que necessitam ser recalculados são processados na ordem em que são percebidos visualmente. Depth of Field em tempo real permite tentativas e erros no ajuste de foco da câmera, proporcionando resultados mais realísticos (Fearing, 1996a).

4.4 Image-based Rendering

Os sistemas para cenários virtuais que operam em tempo real apresentados no Capítulo 2 usam Model-based rendering como processo de produção dos cenários.

Image-based rendering não segue o processo tradicional do rendering e usa imagens pré-renderizadas de uma cena ou fotografias como primitivas básicas, combinando-as para compor o cenário.

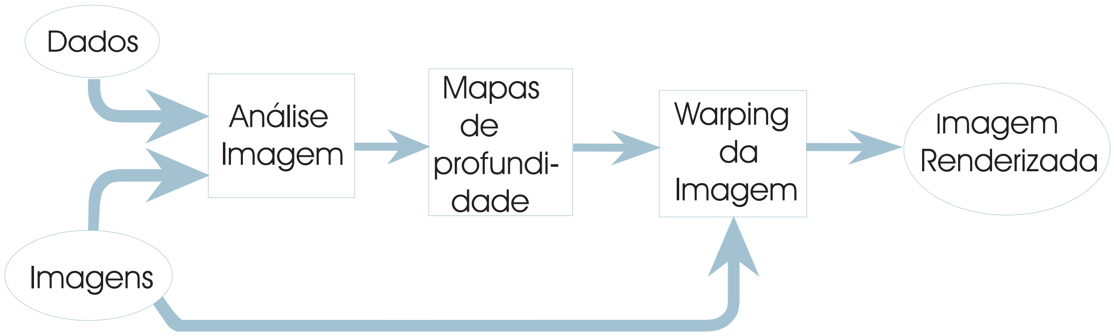


Figura 4.6: Image-based Rendering

O processo de image-based rendering pode ser aplicado a partir de várias fotos obtidas de diferentes pontos de vista. Precisa-se então determinar a profundidade z real de cada pixel da imagem correspondendo à sua localização $3D$ determinando-se assim o *mapa de profundidade* de cada imagem, e a partir destas informações estas imagens podem ser reprojetadas com diferentes perspectivas, em outro plano de projeção. As imagens são tratadas por técnicas de warping e as operações de câmera podem ser simuladas por transformações perspectivas em regiões dessa imagem. Uma abordagem mais conceitual desta área encontra-se em (Gomes *et al.* , 1998).

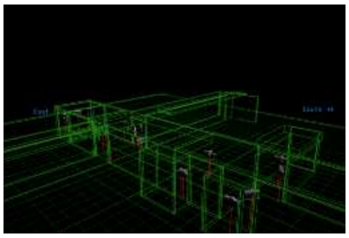
Uma vantagem desta técnica é que a complexidade de cálculo está relacionada ao pixel da imagem, e com isto possibilita o cálculo de cenas complexas com um número constante de operações por pixel. Uma limitação é a dificuldade de gerar um ambiente virtual que seja livremente navegável, pois a aquisição de muitas fotos de um ambiente pode-se tornar inviável. Como consequência existem limitações no movimento da câmera de forma a não envolver áreas que estariam fora da imagem.

Soluções Híbridas

As duas técnicas, Image-based e Model-based, podem ser usadas de forma conjunta onde um modelo geométrico simples é usado com imagens para compor uma cena. As imagens são usadas para orientar a construção da geometria do modelo e também como texturas dependentes do ponto de visão - *view-dependent textures*.

Reconstrução e Rendering de interiores (Debevec *et al.* , 1996) é um trabalho interessante que combina as duas abordagens do rendering. Neste trabalho são pesquisados métodos para rapidamente reconstruir e renderizar, como modelos, interiores de prédios existentes na vida real. A Figura 4.7 ilustra resultados deste trabalho (Healey *et al.* , 1998).

O interior do prédio é scaneado com um laser de alta velocidade e os valores de profundidade dos pixels são usados para reconstruir uma malha $3D$ e modelar a cena. Um sistema de image-based rendering é então usado para mapear fotografias do ambiente como



(a) Modelo geométrico

(b) Foto usada como textura

(c) Foto usada como textura

(d) Resultado do image-based rendering

Figura 4.7: Reconstrução e Rendering de interiores

texturas no modelo. Esta técnica na realidade troca geometria do modelo por textura, para acrescentar realismo.

Embora o trabalho Reconstrução e Rendering de interiores não tenha sido desenvolvido com o objetivo de cenografia virtual, estes resultados obtidos a partir de um sistema de rendering híbrido são muito interessantes para esta área, pois pode-se pensar na reprodução de ambientes famosos e sofisticados como museus, igrejas, etc para filmagens e na construção de um banco de dados destes lugares, o que causaria uma grande economia em filmagens.

Capítulo 5

Composição

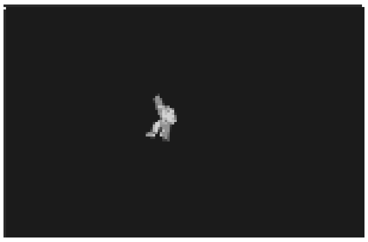
É necessário combinar as imagens reais do cenário real com as imagens do cenário virtual, por isso todos os processos para gerar cenários virtuais englobam Composição de Imagens sob diversas abordagens.

"A combinação é o único método possível para juntar elementos de imagens geradas por processos distintos, por exemplo, cenas reais com cenas geradas por computação gráfica. A importância dessa operação está na flexibilidade que ela propicia ao processo de geração de imagens" (Gomes & Velho, 1994).

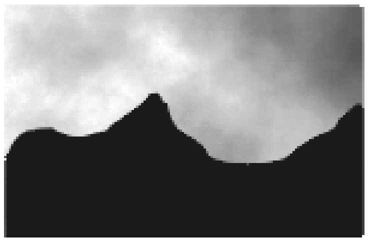
5.1 Composição de imagens

A indústria cinematográfica desenvolveu ao longo dos anos vários métodos analógicos de combinação de imagens, primeiro pela necessidade de filmar pessoas em lugares de difícil acesso e depois por vantagens comerciais, pois as técnicas causavam redução de custos e favoreciam cada vez mais a produção de efeitos especiais.

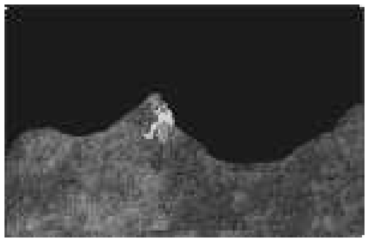
Tradicionalmente usava-se fazer uma tomada de cena onde parte do frame da câmera era encoberto para não haver exposição do filme nesta área. A parte não exposta era depois usada para filmar uma ação em foreground, enquanto a que havia sido exposta era então encoberta com uma película opaca chamada *matte* para proteger a imagem já gravada. Esta técnica era conhecida como *matte painting* e usada para simular distâncias, perigos ou locais imaginários (ver Figura 5.1). Esta técnica possibilitava o uso de maquetes, miniaturas, pinturas para compor as cenas e deixava livre da preocupação de combinar sol e sombras nas diferentes tomadas, porque seriam retocados na montagem final. Como maior desvantagem tinha-se a impossibilidade de atores ou quaisquer outros elementos de foreground movimentarem-se diante do background. Algumas soluções intermediárias surgiram, como por exemplo, aplicar a cena de fundo a um cilindro que girava enquanto o ator representava em frente, técnica denominada "carrocel" (Erland & Dorney, 1982).



(a) primeira tomada do ator



(b) uso de matte nas montanhas



(c) uso de matte para o céu

(d) cena final

Figura 5.1: *Matte Painting*

Como extensão do carrocel surgiram as técnicas *Back Projection* (Figura 5.2) - a cena de background filmada previamente em qualquer lugar é projetada em uma tela e o ator encena diante desta tela, no estúdio - e *Front Projection* (Figura 5.3) - o ator fica diante de uma tela reflexiva na qual a imagem de background está sendo projetada a partir de um espelho em frente da cena.

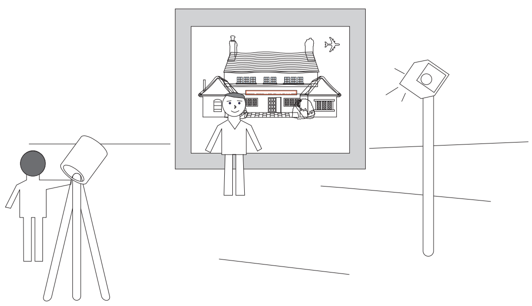


Figura 5.2: *Back Projection*

As duas técnicas produzem praticamente os mesmos efeitos e vantagens, mas *back pro-*

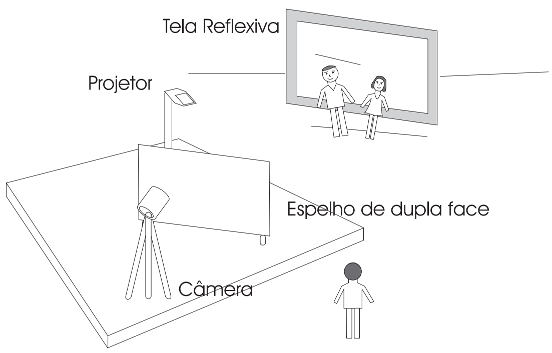


Figura 5.3: *Front Projection*

jection só podia ser usada em grandes produções por envolver o uso de equipamentos de alto custo, por isso *front projection* foi desenvolvida como uma forma alternativa para modestas produções. Principais vantagens destas técnicas são: permitir ao diretor visualizar o resultado da composição no momento da filmagem, o ator poder ver o cenário e por isso atuar melhor e dar condições de efetuar com segurança algumas operações de câmera, como pan e tilt por exemplo, pelo fato da imagem de fundo estar sendo visualizada simultaneamente (Fielding, 1985).

Em contrapartida, estas soluções exigiam sincronismo entre o projetor e a câmera, uniformidade na iluminação entre a imagem de frente (com o ator) e a imagem projetada. A necessidade de ter um background previamente filmado aumentava muito o custo de uso do estúdio, além de causar perda na qualidade da imagem de fundo quando refotografada. Com todos esses ônus, ainda faltava um aspecto importante que era a necessidade de liberdade para movimentos básicos diante do cenário, tanto para o ator como para a câmera, motivando a criação de mattes que podiam mover-se frame a frame, chamados *travelling mattes*, que eram geralmente feitos à mão ou mecanicamente, no início. Depois outros recursos foram desenvolvidos para fazê-los fotografados. Com a chegada do filme colorido surgiu o processo de travelling matte, dominante na indústria de filme até hoje, chamado *bluescreen* (Erland & Dorney, 1982).

Neste caso, o ator encena diante de uma tela azul gerando uma imagem de foreground. Pelo canal de cor é possível gerar um matte em preto e branco e separar os elementos de foreground e background (azul), dando grandes condições de controle sobre estes elementos para posterior composição com um background diferente, e assim obter a cena final.

5.2 Composição Digital

As primeiras técnicas de composição digital apenas reproduziam no computador os efeitos usados convencionalmente pelos produtores de filmes, porém os métodos utilizados diferem passando de um tratamento manual para um processamento matemático.

Uma imagem é definida como uma função bidimensional $f : U \subset R^2 \to C$, onde C é um espaço de cores. Uma operação de composição entre duas imagens só é possível se as imagens pertencem ao mesmo espaço de imagens $I = \{f : U \subset R^2 \to C\}$ e os valores de cor pertencem ao mesmo espaço de cor.

Uma operação simples de combinação de imagens chama-se *dissolve*, onde dadas duas imagens a e b, a imagem resultante é obtida por

$$d_t = dissolve_t(a, b) = (1 - t)a + tb$$

sendo t um número real no intervalo $0 \leqslant t \leqslant 1$. Então, para $t = 0$ a imagem resultante é a e para $t = 1$ o resultado é b, os valores intermediários de t geram uma imagem que é a mistura das duas. O processo equivale a interpolar linearmente a cor das imagens a e b, pixel a pixel, também denominado mistura de imagens.

Em computação gráfica utiliza-se muito fazer a composição dos elementos de imagens, isso é possível através da decomposição frente-fundo do domínio de cada imagem f_i em $U^i = \{U^i_f, U^i_b\}$. No caso de duas imagens f_1 e f_2, determina-se a decomposição frente-fundo $\{U_f, U_b\}$ de cada imagem e a operação de composição denominada *superposição* resulta em uma imagem $f = over(f_1, f_2)$ dada por

$$f(x, y) = \begin{cases} f_1 & \text{se } (x, y) \in U_f; \\ f_2 & \text{se } (x, y) \in U_b. \end{cases}$$

Esta operação não é comutativa, pois a ordem das imagens determina a ordem em que cada elemento aparece na imagem final, podendo-se portanto obter diferentes resultados a partir das mesmas imagens.

Toda a formalização detalhada para os processos de composição de imagens, assim como o exposto acima, encontram-se em (Gomes & Velho, 1994).

Duas técnicas para combinar imagens são muito importantes em cenários virtuais e serão discutidas a seguir: *Canal alfa* e *Composição com profundidade*.

5.2.1 Canal Alfa

Os processos dissolve e superposição trabalham com camadas de imagens retangulares e equivalem ao processo tradicional de combinar mattes. Para combinar várias imagens duas a duas ainda era necessário repetir o processo de rendering para cada novo par de imagens. Para evitar isso surgiu um novo conceito que não apenas repete as técnicas tradicionais. *Canal alfa* permite dar forma e transparência a elementos de uma imagem colorida agregando os conceitos de cor e opacidade. A imagem passa a ser tratada como um conjunto de quatro informações: red, green, blue e opacidade (R, G, B, α) (Smith, 1996).

"Em termos de filme, o canal alfa é exatamente o matte necessário para compor duas imagens" (Smith, 1996). Só que, neste caso, não existe a segunda faixa de filme. A informação de transparência é armazenada em um quarto canal da imagem, e isso faz com que formas possam ser definidas e combinadas com outras imagens. Imagens que possuem um canal alfa podem assumir a forma dos objetos que a compõem porque os pixels em torno destas formas podem ser completamente transparentes ou completamente opacos.

A fórmula para composição pode ser reescrita em função do canal alfa, e a imagem final é obtida por

$$h = \alpha f + (1 - \alpha)g,$$

onde α tem valor 0 quando o pixel da imagem é completamente transparente e α é 1 quando o pixel é opaco, os valores intermediários representam o percentual de contribuição de cor de cada imagem para o pixel, na composição final.

Os componentes RGB contém apenas informação de cor, a operação de superposição do pixel com o background é dada pelo canal alfa que funciona como um fator de composição que controla a operação de interpolação entre as cores do foreground e background. Isso faz com que as imagens deixem de ser representadas apenas por retângulos opacos e possam representar seus elementos opacos com formas próprias assim como outras regiões com transparência. A Figura 5.4 mostra uma imagem com seu canal alfa. A imagem da esquerda contém um objeto isolado não combinado com um background, e cujos pixels tem valor de $\alpha > 0$, que (Smith & Blinn, 1996) chamam de *image sprite* ou apenas *sprite*.

Figura 5.4: Imagem Digital e seu canal alfa

Para combinar duas imagens, onde cada pixel da imagem f_1 possui a representação de cor e seu canal alfa $c_1 = (r_1, g_1, b_1, \alpha_1)$ e da imagem f_2 em $c_2 = (r_2, g_2, b_2, \alpha_2)$, e o pixel final será $c_f = (r_f, g_f, b_f, \alpha_f)$, pode-se escrever $c_f = c_1$ **op** c_2, onde **op** é um operador de composição que vai determinar diferentes formas de compor as imagens.

Por exemplo, o operador **over** faz a imagem de frente se sobrepor à imagem de fundo. O operador **inside** faz a imagem f_2 agir como uma máscara para f_1, e f_1 mostra somente a área onde é visível em f_2. Esses operadores fazem um total de 12 operações, pois a maioria deles não é comutativa e são também denominados *Alpha Blending*. Os detalhes dessas operações encontram-se em (Gomes & Velho, 1994).

O canal alfa terminou com o conceito de "matte painting" e levou a uma solução de "travelling matte" digital.

5.2.2 Cálculo do Canal Alfa

O método usado para o cálculo do canal alfa depende do tipo da imagem. Para o caso de cenários virtuais, tem-se maior interesse nas imagens sintéticas e nas imagens de vídeo. Nas imagens sintéticas obtém-se o canal alfa de forma natural, pois este é calculado pelo processo de rendering, já que as informações geométricas estão disponíveis no modelo. Neste caso, o cálculo do canal alfa não constitui um problema.

As imagens digitais não possuem informação de canal alfa disponível, sendo esta informação obtida de um processo de análise da imagem. Então a técnica *bluescreen* é usada para auxiliar na determinação do canal alfa em imagens digitais. No domínio digital o bluescreen permite criar o matte, que é o canal alfa, para ser posteriormente reutilizado na composição, e usa a cor de fundo azul como informação de crominância para separar o que é background e foreground na imagem. Como obter então o canal alfa de uma imagem digital usando a técnica de bluescreen?

Bluescreen ou Chroma key

"A indústria de vídeo usa os termos *key* e *keying* assim como *Chroma keying* no lugar de *matte* e *matting* usados na indústria de filmes" (Smith & Blinn, 1996).

(Smith & Blinn, 1996) apresentam o problema de chroma key da seguinte forma: o problema refere-se a extrair o canal alfa de objetos de foreground, conhecendo-se apenas uma imagem combinada que contém esses objetos. Em uma imagem combinada a cor de qualquer ponto desta imagem pode ser definida como $C = [R\ G\ B\ \alpha]$, função de C_f - cor do elemento de foreground - e C_b - cor do elemento de background - do ponto correspondente. C_f é a combinação de um background especial de cor constante C_k e um foreground C_0 que é o objeto em si isolado de qualquer background transparente ou parcialmente transparente, mas cuja cor não pode ser vista através do elemento C_0. Então C_f é expressa como uma composição de C_k e C_0 em cada ponto na forma: $C_f = f(C_0, C_k)$ considerando-se $\alpha_k = 1$ para C_k. f é então a função **over** que pode ser escrita como:

$$C_f = C_0 + (1 - \alpha_0)C_k.$$

O problema consiste em determinar C_0, já que C_f e C_k são conhecidos, para então obter-se a cor final da composição pela fórmula

$$C = C_0 + (1 - \alpha_0)C_b,$$

para todos os pontos que C_f e C_b têm em comum. Então C_0 que é a cor do objeto de foreground, incluindo alfa, é a solução do problema de chroma key.

Precisa-se determinar então R_0, G_0, B_0 e α_0. Cada elemento de cor, R_f por exemplo, é resultado de uma interpolação das duas imagens R_k e R_0; portanto tem-se três equações com quatro incógnitas caracterizando um problema indeterminado com infinitas soluções. Diversas soluções podem ser encontradas a partir de hipóteses de valores para R_0, G_0 e B_0.

A solução muito usada é supor que C_0 não tem azul e que C_k só tem azul, que é o caso do bluescreen.

(Smith & Blinn, 1996) afirmam que existe uma infinidade de soluções para o problema e não existe um método algorítmico para extrair o *sprite* de uma imagem real. Em seu trabalho são discutidas várias soluções a partir de hipóteses de valores de cor para o fundo da imagem e feitas algumas generalizações para o problema a partir de estudo de casos especiais.

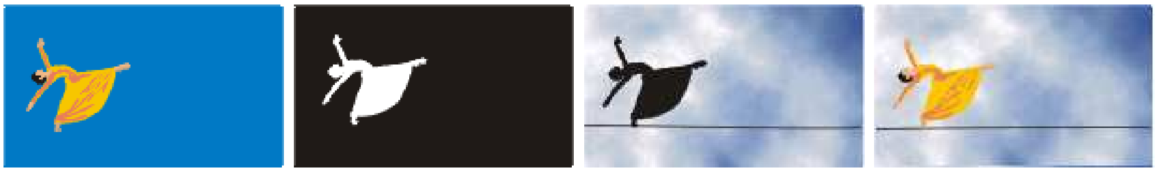


Figura 5.5: *Bluescreen* com Canal Alfa

A Figura 5.5 mostra um ator filmado diante de um fundo azul, seu canal alfa e a composição final.

5.2.3 Composição com profundidade

A informação de profundidade z da cena representa a distância entre a câmera e o objeto da cena no ponto (x, y). Neste caso as informações armazenadas para cada pixel da imagem são do tipo (R, G, B, z), onde z é o valor usado para determinar como os objetos se sobrepõem na imagem final. Dadas duas imagens f e g, a imagem h resultante da composição será:

$$h(x,y) = \begin{cases} f(x,y) & \text{se } Z_f(x,y) < Z_g(x,y); \\ g(x,y) & \text{se } Z_f(x,y) > Z_g(x,y); \\ f(x,y) & \text{se } Z_f(x,y) = Z_g(x,y). \end{cases}$$

A imagem final armazena apenas o valor de cor do ponto correspondente (Gomes & Velho, 1994).

Combinação usando Z-buffer

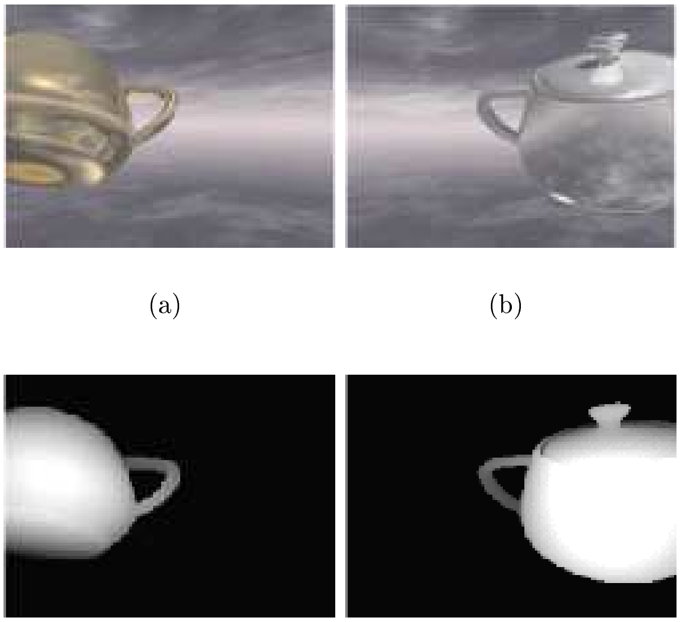
O algoritmo de superfícies escondidas - *Z-buffer* - compara as profundidades entre os elementos analisados e armazena o mais próximo, determinando assim quem é visível na cena.

Sendo h a combinação de duas imagens f e g, usando z-buffer, tem-se:

$$h(p) = \begin{cases} f(p), & \text{se } f_z(p) < g_z(p) \\ g(p), & \text{nos outros casos} \end{cases}$$

para cada pixel p (Gomes *et al.* , 1998).

Este algoritmo não requer uma pré-ordenação e nem faz uma comparação entre os objetos. Todo processo resume-se a uma procura nos conjuntos de valores das imagens, comparando o valor de z para cada par fixo de (x, y).



(a) (b)

(c) (d)

Figura 5.6: Imagens originais (a)(b) e seus Z-buffers (c)(d)

A Figura 5.6 mostra duas imagens com seus respectivos mapas de profundidade, obtidos pelo algortimo z-buffer.

A Figura 5.7 mostra o resultado da composição através dos mapas de profundidade. Observe-se que não há sombras de um objeto em relação ao outro após a composição, o que seria uma falha para cenários virtuais.

O exposto acima é aplicável quando já se conhece o valor de z de cada ponto da imagem. Para as imagens sintéticas a informação de z também pode ser armazenada durante o processo de rendering. O processo de obtenção da informação de z constitui um problema para as imagens digitais, pois esta informação não encontra-se disponível na imagem e deve ser obtida durante a geração da mesma.

A informação de profundidade nas imagens de vídeo pode ser obtida através de câmera com sensor, do uso de técnicas de estéreo ou através de estimativas de valores fazendo-se medidas no estúdio.

Figura 5.7: Composição usando Z-buffers

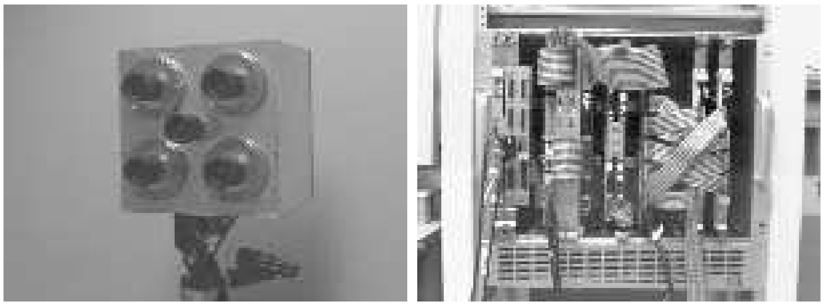


Figura 5.8: Câmera com cinco lentes

A Figura 5.8 mostra um conjunto com cinco câmeras usado com um processador de vídeo, desenvolvidos na *Carnegie Mellon University*, para obter mapas de profundidade (Kanade *et al.* , 1996). As lentes formam a configuração de um X, onde a câmera do meio é a câmera de base e as outras quatro formam quatro pares de estéreo.

5.3 Composição e cenários virtuais

Chroma keying assume que um objeto do mundo real sempre vai estar na frente (foreground) da cena sintética (background). Em cenários virtuais este processo não atende às necessidades pois para o ator parecer imerso no ambiente virtual ele precisa poder ocupar qualquer posição dentro da cena. Precisa-se acrescentar a informação de profundidade nas imagens para realizar esse efeito, e a informação de chroma key é necessária para separar o ator (sprite) do ambiente do estúdio (fundo azul).

Chroma key ainda é a técnica básica utilizada para realizar a composição de cenas em um estúdio virtual, mas novos componentes tornam-se essenciais para a obtenção deste efeito. É necessário que atores e cenário possam ser combinados de forma que um possa encobrir o outro em qualquer ordem. Então o conceito de foreground e background já não é simples, pois o cenário sintético, a princípio denominado cena de fundo, precisa poder colocar elementos à frente de atores quando isto for desejado (exemplo: mesas diante dos atores). Este tipo de composição não é possível usando apenas a informação de chroma key. A denominação foreground continua sendo usada para referenciar a imagem obtida do cenário real, onde está o ator e background para referenciar o cenário sintético virtual, mas não mais no sentido de estar totalmente na frente e totalmente atrás. Além disso, a cena a ser combinada como ambiente é sintética, um ambiente virtual, trazendo então o problema de combinar sombras e iluminação. Como ponto principal para as transmissões de televisão ao vivo tem-se ainda a necessidade de composição em tempo real, eliminando uma ferramenta auxiliar fundamental que é o recurso da pós-produção.

O conhecimento da informação de profundidade para os atores e para o cenário, ou seja, imagens reais e imagens virtuais, torna-se imprescindível para efetuar a composição na ordem não convencional dos elementos.

Projeto Monalisa

O projeto Monalisa (Monalisa, 1994) desenvolveu um sistema para composição sensível à informação de profundidade, chamado *Z-Mixing*, aplicado a transmissões ao vivo usando cenários sintéticos. A imagem do cenário, referenciada como imagem $3D$, contém informação adicional da distância entre a câmera virtual e cada pixel da imagem gerada, chamado *valor de Z* ou *profundidade*. Para a imagem do estúdio obtida de uma câmera comum não há informação de profundidade disponível em tempo real, então usou-se estimar a distância entre o ator e a câmera real e usar este valor como constante para toda a imagem dita "foreground". Quando o ator move-se é necessário um novo cálculo de Z, porém apenas os pixels da imagem que correspondem ao ator são atualizados e inseridos na imagem de background, os demais são descartados pela técnica de chroma key e canal alfa, descritas nas seções anteriores, gerando assim o sinal para composição que permite separar o ator do fundo azul.

Figura 5.9: Efeito de *Fade* entre dois cenários

Neste projeto são usados dois tons de azul para o fundo do estúdio, um para a informação de crominância usada na técnica de chroma key e o outro para informações de coordenadas do ator para sincronismo de câmera (ver Capítulo 6). O sistema Z-mixing efetua alguns efeitos de composição em tempo real como *cut, dissolve e fade*. A Figura 5.9 mostra o efeito de fade entre dois cenários.

O sistema tem as seguintes restrições: só é possível um ator no cenário de cada vez ou, se houver mais de um, eles devem ter a mesma distância em relação à câmera. Não pode haver interação entre atores e elementos do cenário sintético.

Z Keying

Z Keying é um método baseado em informação de profundidade, que tem o mesmo objetivo de ocludir cenas reais com objetos virtuais e vice-versa, este trabalho está citado no Capítulo 2, Seção 2.4.2. Para cada pixel da imagem, Z key compara a profundidade z das imagens real e sintética. O mapa de profundidade da imagem real é obtido através de uma máquina estéreo que funciona como um sensor, desenvolvida na própria CMU (Figura 5.8), e produz imagens de 256 x 240 pixels com profundidade a uma taxa de 30 frames por segundo em tempo real (Kanade *et al.* , 1996).

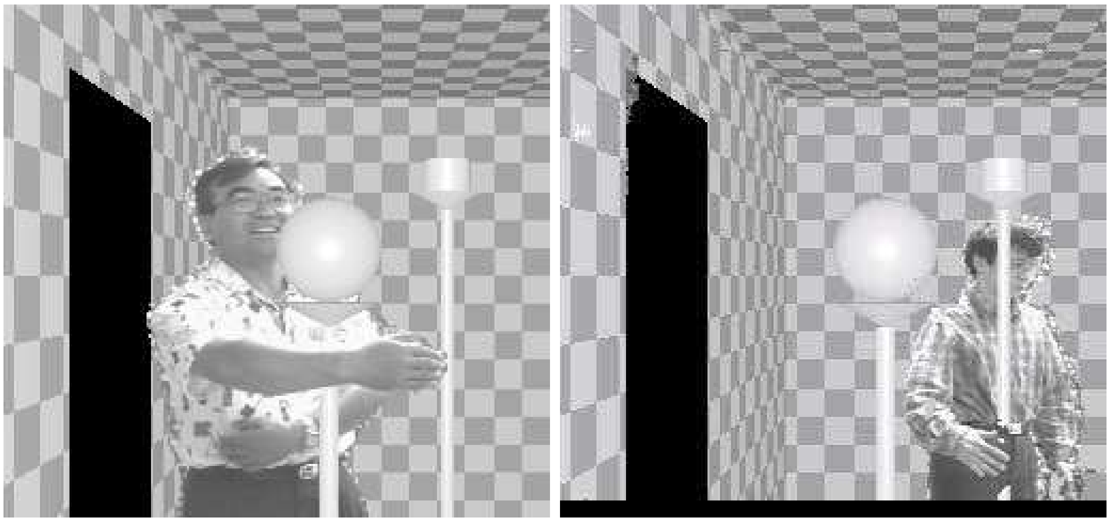


Figura 5.10: Composição com oclusão

Necessidades e Limitações

Os resultados do Z keying apresentam falhas no contorno das pessoas, pois percebe-se partes do fundo da imagem real formando um contorno na pessoa (ver Figura 5.10). O método não usa chroma key para separar a pessoa (imagem real) do fundo da cena e apresenta-se como uma forma de substituição do chroma key.

Este trabalho não foi desenvolvido com o objetivo de cenários virtuais e sim para composição de imagens com oclusão. Para aplicá-lo em cenografia virtual, parece resolver muitos problemas de oclusão, mas precisaria somar-se à técnica de chroma key para obter resultados mais realísticos. O projeto Monalisa não é tão versátil na questão da oclusão, pois não usa câmera como sensor e sim estimativa por algoritmo para obter o mapa de profundidade, mas combina sua técnica com chroma key e consegue resultados mais satisfatórios do ponto de vista de transmissões para televisão.

Em termos de composição para cenários virtuais é como se hoje estivessemos no estágio digital dos "travelling mattes", buscando ampliar os movimentos livres para o ator e para a câmera, só que agora de forma automatizada, pois já não satisfaz mais simplesmente colocar o ator em frente a um cenário gerado por computador, é necessário passar a impressão de imersão, e tudo isso em tempo real. O que falta? Falta ainda a idéia do ator poder ver o cenário sintético, ou melhor, ambiente virtual, para sentir mais integração na interpretação. Interatividade e Imersao do ator? Deseja-se que o ator "interaja" ou pareça ter contato físico com atores e objetos do ambiente sintético para parecer mais natural. Conseguir resultados satisfatórios e mais realísticos é um processo em evolução, portanto o processo de composição ainda precisa sofrer melhorias para que os mundos real e virtual pareçam realmente integrados.

Capítulo 6

Sincronismo de Câmera

O sincronismo consiste em obter uma perfeita correspondência entre os parâmetros da câmera real e os da câmera virtual, isto equivale a dizer que no referencial virtual a posição e orientação da câmera virtual devem ser as mesmas da câmera real no mundo real, como se uma estivesse sobrepondo a outra, correspondência esta que precisa ser mantida durante toda a filmagem, ao longo do tempo.

Este sincronismo é fundamental para que qualquer movimento efetuado pela câmera real gere exatamente o mesmo movimento na câmera virtual, e não haja incoerência no resultado final, pois isso passaria ao telespectador a percepção de uma imagem não natural e sem realismo. A projeção correta da imagem obtida do rendering depende da precisão do sincronismo. Para produzir o cenário virtual com a perspectiva exata em relação à cena real, o sistema de cenário virtual deve, antes de tudo, conhecer de forma precisa a posição e orientação da câmera no estúdio. Com isso, espera-se que o sistema atenda a um critério de performance necessário, isto é, precisão no registro entre a imagem real e virtual.

Sincronismo e Calibração

Inicialmente o posicionamento da imagem real e da imagem gráfica precisam ser ajustados até obter-se o alinhamento desejado entre as imagens. Isto só é possível conhecendo-se a posição e orientação (parâmetros extrínsecos), e distância focal e distorção das lentes (parâmetros intrínsecos) da câmera real, que serão fixados. A isto chama-se *calibração da câmera*. Estes parâmetros são obtidos em função das informações sobre os objetos na imagem e das informações sobre estes objetos no espaço. Manter o sincronismo equivale a ter que fazer uma calibração das câmeras a cada instante.

Sincronismo e Tracking

Na prática a câmera não deve ficar fixa, pois isso ocasiona uma total limitação na direção da cena. É fundamental permitir movimento da câmera e manter sincronismo, em tempo real.

Isto é possível através do *tracking* da câmera, que é o acompanhamento do seu movimento a cada instante. É necessário dispor de técnicas de tracking para manter atualizadas as informações sobre os objetos na imagem, e sobre os objetos correspondentes no espaço.

Existem atualmente algumas técnicas em uso tentando atingir este resultado, mas ainda não foi dada uma solução ideal que atenda a todos os tipos de movimentos que se quer controlar e às diferentes configurações de sistemas (Radamec, 1995). As soluções para o sincronismo envolvem a captura dos parâmetros da câmera real por um mecanismo de *tracking*, para determinar os parâmetros da câmera virtual, a cada instante.

Portanto, o sincronismo em cenário virtual requer sincronismo temporal obtido pelo tracking dos objetos e câmeras ao longo do tempo e sincronismo espacial obtido através da calibração de câmeras, no instante t.

6.1 Calibração de câmera

O problema de calibração de câmera é estudado de forma acentuada na área de visão computacional, onde pretende-se de uma imagem 2D extrair os parâmetros intrínsecos e extrínsecos da câmera que gerou a imagem. Esta abordagem é trazida para a área de cenário virtual como uma ferramenta para sincronismo de câmera.

(Tommaselli & Tozzi, 1991) apresenta um estudo detalhado de vários métodos de calibração de câmera para aplicações na área de visão. As técnicas de calibração discutidas vão desde as clássicas até algumas mais recentes, usando objetos e linhas como pontos de apoio para a calibração. Alguns métodos de calibração, que estudam apenas a determinação dos parâmetros extrínsecos, são denominados de *Resseção espacial*, ou ainda, métodos de determinação de localização, e podem ser aplicados quando os parâmetros intrínsecos não variam com freqüência, ou quando esta variação pode ser determinada em função da distância focal.

Dentre os métodos expostos, destacam-se o *Método de Tsai*, usando pontos de apoio e o *Método de Liu, Huang e Faugeras*, usando retas, como métodos mais práticos.

- **Método de Tsai** - caracteriza-se pela divisão da calibração em duas etapas, para evitar uma otimização não linear típica do método clássico. A maior vantagem apresentada é o tempo de processamento, permitindo o uso para aplicações em tempo real. Como desvantagem tem-se a propagação de erro, que é atribuída ao fato dos parâmetros serem separados para cálculo.

- **Método de Liu, Huang e Faugeras** - este algoritmo é a adaptação de um algoritmo inicial de Liu e Huang, que permite o cálculo dos parâmetros extrínsecos em duas etapas. Na primeira etapa são calculadas as rotações com a resolução de um sistema de ordem 3. Na segunda etapa são calculadas as translações através de um modelo linear com 3 incógnitas. O uso deste algoritmo tem mostrado que com 4 retas obtém-se bons resultados.

"A escolha do método de calibração dependerá da aplicação. Devem ser considerados: necessidade de tempo real, precisão exigida, tipo de câmera e natureza dos parâmetros (feições) encontrados no ambiente e que serão usados como apoio" (Tommaselli & Tozzi, 1991).

Um exemplo de calibração de câmera

Juiz Virtual (Carvalho *et al.* , 1998) apresenta um método de calibração de câmera em duas etapas. A primeira etapa, onde é aplicado um processo de otimização, resulta em uma transformação projetiva, que ainda não é a transformação de câmera procurada. A segunda etapa determina a câmera verdadeira, através do reajuste dos parâmetros obtidos da etapa anterior. O método é baseado na formulação de modelos lineares de mínimos quadrados e é usado para analisar lances duvidosos em partidas de futebol, a partir de uma imagem estática obtida do jogo.

Figura 6.1: Juiz Virtual

A feição usada neste método são pontos de apoio especificados pelo usuário. Então, dada uma imagem contendo pontos conhecidos, precisa-se determinar uma transformação T, que relacione como pontos em uma cena (referencial $3D$) são mapeados em pixels da imagem (referencial $2D$). Invertendo-se esta transformação, pode-se determinar a posição $3D$ dos objetos na cena. A partir desta informação, os parâmetros de visualização podem ser modificados para posicionar a câmera sintética em diferentes posições, permitindo ver a mesma cena de outros ângulos.

O trabalho tem duas principais contribuições: uma nova técnica de calibração de câmera e ainda, como integrar um ambiente de modelagem como OpenGL com informações de uma imagem e produzir cenas sintéticas que correspondem a esta imagem (Figura 6.1).

Esta é uma aplicação pós-processada e que utiliza apenas imagens estáticas.

6.2 Tracking

"O tracking é também usado em quase todas as tomadas de efeitos especiais para estabilizar e suavizar o movimento dos objetos, para combinar várias camadas de imagens obtidas de fontes diferentes, adicionar objetos gerados por computador às cenas reais e ainda para mapear texturas ou reflexos sobre objetos em movimento" (Lando, 1997).

6.2.1 Requisitos básicos para sistemas de tracking

Medidas de posição e orientação

O sistema de coordenadas deve ser definido em função da área de trabalho. Em uma aplicação real, os seis parâmetros (3 de rotação e 3 de posição) são necessários, pois a imagem que é capturada pela câmera é dependente de todos estes parâmetros.

A função principal do sistema de tracking é adquirir a posição e orientação da câmera real e passar para o computador que controla o sistema. A posição é dada na forma de um vetor tridimensional, com base em um sistema de coordenadas cartesianas tridimensional, que pode ser definido localmente em relação a um ponto de referência no próprio estúdio. A orientação pode também ser indicada por um vetor tridimensional, e seguir diferentes especificações de referência. Um critério de orientação muito usado tem como base os ângulos de Euler - *roll, pitch e yaw*, cuja especificação é feita por três rotações distintas e relativas a um sistema de referência local. Outra especificação de orientação que supera alguns problemas gerados pelos ângulos de Euler é denominada *quatérnio*. Esta foi a estrutura de orientação de câmera utilizada no estudo feito neste trabalho e encontra-se bem definida no Capítulo 7.

Latência

É o tempo que o sistema gasta para informar ao computador de controle uma mudança ocorrida na posição e orientação. O tempo de latência deve ser o menor possível, considerando-se que, para aplicações em tempo real, este é um requisito de extrema importância.

Precisão

É a medida do erro na posição e orientação informada pelo *tracker*. O sistema de tracking deve ser sensível à mínima mudança nos parâmetros de posição para garantir a precisão das informações. Mas a medida de orientação é ainda mais importante, pois o menor desvio de ângulo na câmera pode causar uma grande modificação na imagem. (Oschatz, 1994) descreve um exemplo de cálculo de erro na medida do ângulo yaw, e também acrescenta que no caso de sistemas com sensores, estes devem ser acoplados a uma certa distância da câmera. Este fato não influencia diretamente na estimativa de posição, mas causa erro na medida de orientação.

Taxa de atualização

O sistema de tracking necessita de um tempo entre a medida de um ponto na imagem e o processamento do mesmo pelo computador de controle.

6.2.2 Tipos de tracking

(Bhatnagar, 1993) apresenta um estudo dos tipos de sistemas de tracking com importantes características. As quatro principais categorias de tracking abordadas são: magnético, acústico, ótico e mecânico. O magnético difere muito da abordagem deste trabalho e está bem descrito na referência citada, portanto não será detalhado aqui.

Tracking Acústico

Tracking acústico usa ondas ultrasônicas para determinar posição e orientação do objeto em mira. Os meios de transmissão usados para medidas são paredes entre dispositivos emissores e receptores, ecos ou fortes fontes de ruídos. A velocidade do som no estúdio deve ser medida em relação a uma distância conhecida usada como referencial, pois os parâmetros de som sofrem influência de fatores como temperatura e umidade do ar. Estes sistemas têm a desvantagem de necessitar de um período relativamente longo para medidas devido à velocidade do som.

Tracking Mecânico

De forma genérica, o tracking mecânico mede a posição e orientação do objeto alvo que fica acoplado à extremidade de um braço móvel mecânico. Este braço é ancorado a um ponto de referência fixo e possui articulações que permitem translação e rotação. Estes movimentos são medidos por garras ou potenciômetros.

Tracking Ótico

Os tipos de sistemas de tracking ótico apresentados em (Bhatnagar, 1993) referem-se todos a técnicas baseadas em recursos de iluminação: sinalizadores óticos, diodos, transmissores de luz, refletores de luz, etc. Na literatura de cenários virtuais esta denominação é dada a sistemas que usam técnicas de reconhecimento de padrões nas imagens, recebendo, portanto, a mesma denominação para abordagens diferentes.

6.2.3 Sistemas de Tracking em cenário virtual

Nos sistemas existentes para cenários virtuais, encontram-se dois tipos de soluções para tracking: os que usam processos ótico-mecânicos, através de sensores - que são codificadores óticos, e câmeras robotizadas, etc. Outros utilizam padrões nas paredes do estúdio real que possam ser reconhecidos na imagem, ter suas coordenadas determinadas e, assim, proporcionar a calibração da câmera.

Na literatura existente, estes primeiros são denominados *tracking eletromecânico* e os outros denominados *tracking ótico*. Esta denominação, entretanto, não é adequada para as soluções de cenários virtuais, pois não tem coerência nas suas características. Portanto, apresentamos a seguinte classificação para tracking em cenário virtual: **Tracking monitorado** e **Tracking algorítmico**.

Tracking monitorado

São os que utilizam quaisquer dispositivos óticos, mecânicos, etc. que sejam alheios ao cenário natural, e que desempenhem a função de controlar e capturar o movimento da câmera.

Os sistemas *monitorados* surgiram primeiro, são ainda os mais usados e utilizam mecanismos para controlar a câmera por diferentes processos, através do uso de sensores presos à câmera ou a diferentes lugares no estúdio. Os sensores podem funcionar de forma ativa ou passiva. Quando ativo, os controles (sensores) são usados para guiar a câmera. No caso passivo, sensores são acoplados à câmera para detectar seus movimentos.

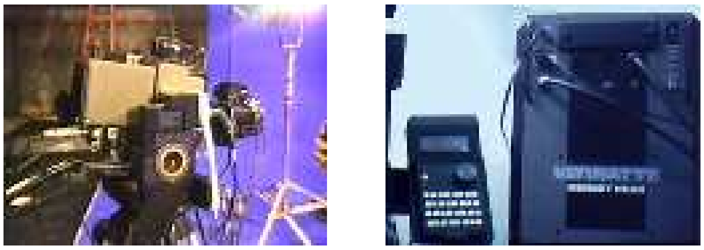


Figura 6.2: Ultimate MEMORY Head

Para cenários virtuais, o objeto alvo de medida é a câmera real. Esta precisa estar acoplada firmemente ao sistema de tracking, que passa as informações ao computador de controle. Como vantagem, este sistema permite medidas rápidas e com precisão. Nos estúdios de TV a câmera precisa estar sobre plataforma móvel para que o acoplamento físico não represente um problema.

Um exemplo de sistema comercial para tracking monitorado é o *Ultimate MEMORY Head of the Ultimate Corp.*, Figura 6.2. Este sistema utiliza “motores de passo” de alta precisão para medir e enviar para o computador a elevação, os ângulos de pan, tilt e roll da câmera, assim como informações sobre distância focal e ajuste de foco (Corp., 1993).

Tracking algorítmico

São os processos que usam apenas algoritmos para o controle e recuperação de parâmetros da câmera. Estes sistemas permitem o uso de qualquer câmera na filmagem, até mesmo câmeras manuais, e são baseados em reconhecimento de padrões.

Utilizando-se técnicas de processamento de imagens pode-se extrair os parâmetros de posição, orientação e campo de visão da câmera que gerou a imagem, usando pontos de referência ou ainda marcas de grade no fundo azul do estúdio de filmagem - azul ou verde, são os mais padronizados. Dentro da teoria de reconhecimento de padrões para processamento de imagens existem diferentes procedimentos que podem ser aplicados ao tracking não-monitorado, pelo uso de diferentes algoritmos desenvolvidos para esta aplicação. O tempo de cálculo dos algoritmos envolvidos é o responsável pelo tempo de resposta destes sistemas, podendo-se, portanto, atingir tempos curtos. Com a evolução contínua dos computadores estes tempos tendem a diminuir. Em contrapartida, estes sistemas exigem calibração da câmera para obter-se resultados precisos. Além disso, a grande quantidade de iluminação dos estúdios pode interferir na obtenção dos dados, dificultando o processo.

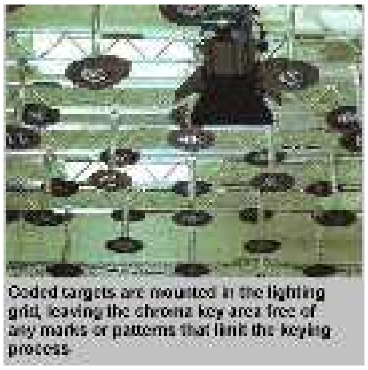
Os que desenvolveram sistemas baseados em tracking monitorado afirmam ser extremamente difícil conseguir resultados com a precisão necessária e em tempo real sem o auxílio de sensores. Por outro lado, os que usam técnicas algorítmicas garantem suas vantagens em relação ao anterior como, por exemplo: os modelos de câmera utilizados podem ser comuns e dispensam o uso de câmeras robotizadas, não sendo necessário nenhuma instalação especial no estúdio. Em ambas as soluções, entretanto, ainda existem muitos problemas a serem tratados.

6.3 Algumas Soluções para Sincronismo

A seguir serão descritas algumas técnicas e equipamentos utilizados nas soluções de tracking em sistemas existentes para a construção de cenários virtuais.

Free-d Tracking System

Free-d é um módulo integrante do sistema Virtual Scenario (Radamec, 1995) que utiliza tracking monitorado. O sistema Free-d fabricado pela Radamec usa uma quantidade de codificadores distribuídos na grade de iluminação do estúdio, deixando a área de fundo azul do cenário sem qualquer marca. Cada codificador é identificado por um código de barra circular e o uso de material retro-reflexivo assegura que eles ficam visíveis em quaisquer condições de iluminação do estúdio (ver Figura 6.3.a). A imagem gerada pela câmera é



(a)

(b)

Figura 6.3: Free-d tracking system

processada pelo Free-d para calcular a posição e orientação exatas da câmera no estúdio a partir de uma análise da imagem, em tempo real, que identifica cada codificador, que tem sua posição conhecida fisicamente e assim possibilita determinar a posição da câmera com alta precisão, segundo a Radamec. O eixos de zoom e foco são monitorados por sensores óticos acoplados às lentes. Esta informação é combinada com a proveniente da análise dos codificadores para finalmente determinar a posição exata da câmera no estúdio. Estes dados são enviados ao sistema para consequente cálculo da imagem sintética a partir do cálculo da posição da câmera virtual.

Monalisa camera tracking

O projeto (Monalisa, 1994) desenvolveu um algoritmo de estimativa de movimento que é capaz de medir movimentos globais na imagem com um alto grau de precisão, de forma que a câmera possa ser movimentada sem a ajuda de sensores mecânicos, o que caracteriza um sistema de tracking algorítmico.

Em muitas aplicações usa-se um background uniforme, pois a identificação da cor chave é feita pela análise dos valores do RGB, mas isso não dá ferramentas necessárias para analisar movimentos de câmera. Para conseguir medir variações de translação e escala a partir da imagem, foi necessário inserir detalhes no background. Por isso, Monalisa usa como background um padrão com vários tons de azul, como uma forma de gerar um sinal-chave. No sistema, um sinal único é formado permitindo ver todas as variações de azul do background. Aplica-se um filtro de passa-baixa a este sinal, obtendo-se um sinal-chave e com isso é possível ignorar os objetos de foreground, que passarão a ser desconsiderados até o final do processo. Ao invés de analisar valores do RGB (cor-chave), é feita a análise de um sinal-chave.

Figura 6.4: Tracking algorítmico - Projeto Monalisa

O método implementado provê meios de remover objetos de foreground da imagem antes do processo de estimativa de movimento, de forma que o movimento dos atores, por exemplo, não interfira nos cálculos. O principal parâmetro do algoritmo é o número de pontos da imagem, sendo necessário a medida de 900 pontos na imagem, arranjados em uma grade retangular de 30 x 30 pontos para obter-se a precisão necessária.

Cyberset - Orad's Virtual Set

Os parâmetros extrínsecos e intrínsecos da câmera são extraídos em tempo real usando uma tecnologia de reconhecimento de padrões, e dispensando o uso de sensores mecânicos, constituindo-se em outro exemplo de tracking algorítmico. O algoritmo que efetua a captura dos parâmetros em tempo real foi implementado em um processador de vídeo - DVP-100 (Orad, 1995).

Figura 6.5: Padrão utilizado para reconhecimeto na imagem - Orad's Virtual Set

O sistema usa uma parede de fundo com um padrão de grade que auxilia na extração da posição, orientação e distância focal da câmera em tempo real. O background utilizado é composto de dois tons de azul, o azul de fundo é usado para composição das imagens por chroma-key (ver Capítulo 5), o outro tom mais claro que forma a grade é usado para fazer medidas das coordenadas da imagem, efetuando o tracking e permitindo sincronismo das câmeras. O tamanho deste painel é adaptável em função do tamanho do estúdio de filmagem.

Análise

Os métodos de calibração de câmera, em geral, são estáticos. Para cenários virtuais a meta é poder mover a câmera livremente ou o ator poder agir livremente. As técnicas de calibração consideram os parâmetros calculados em um determinado instante, supondo-se que estes se mantém constantes por um determinado período, após o qual necessita ser feita uma nova calibração. Este fato torna difícil a obtenção de um movimento contínuo de câmera. Alguns métodos usam interpolar as posições intermediárias para obter movimento.

Os sistemas existentes para cenários virtuais, como foi exposto, apresentam soluções com câmeras robotizadas, que requerem equipamentos de alto custo e difícil operabilidade e outros usam reconhecimeto de padrões que recaem nas limitações de calibração citadas acima. Esta motivação levou a buscar o estudo de uma técnica para sincronismo que pudesse ser incremental, ambicionando superar diversas limitações apresentadas pelas demais. O ideal é trabalhar com uma técnica que dê condições para que o controle sobre a câmera virtual seja equivalente ao controle do *câmera-man* na câmera real, o que denominamos de *Paradigma do câmera-man*.

No Capítulo 7 será apresentado um estudo de uma técnica que consideramos adequada para este objetivo.

Capítulo 7

Estudo de Caso

Em *Through-the-Lens Camera Control* (Gleicher & Witkin, 1992) os autores introduzem um conjunto de técnicas que permitem ao usuário manipular uma câmera virtual pelo controle e restrições de propriedades na imagem vistas através das lentes da câmera, exemplificando com aplicações em composição de imagens e manipulação de cenas.

O presente estudo quer mostrar o potencial destas técnicas visando sincronismo, em tempo real, das câmeras real e virtual.

7.1 Controle de Câmera Virtual

Posicionamento e controle de câmera virtual têm desempenhado um papel importante na composição de imagens e animação por computador, pois a forma como uma cena $3D$ é vista numa tela de computador é função direta do modelo de câmera utilizado e como esta foi especificada. Por isso, grandes esforços têm sido canalizados para o desenvolvimento de modelos de câmera para computação gráfica. A maioria das formulações de câmera são construídas com base num modelo de projeção perspectiva, onde qualquer vista $3D$ é especificada dados o centro de projeção, o plano de visão e o volume de visualização. O modelo de câmera mais utilizado é o *Look-at/Look-from/View-up*, que está bem descrito em (Foley *et al.* , 1992).

Parametrização é uma representação específica das condições de um objeto. O fato de não ter uma parametrização única que atenda a todas as necessidades de controle de câmera gera dificuldades em usar parâmetros da câmera diretamente como controles.

Os modelos de câmera são inflexíveis. Para mudar os controles, deve-se escolher um modelo diferente pré-existente ou deduzir e implementar um novo. Se essa inflexibilidade pudesse ser retirada, os esforços dedicados ao controle de câmera poderiam ser reduzidos (Gleicher & Witkin, 1992).

Through-the-Lens Camera Control apresenta uma solução geral para esse problema. Ao invés de dados fixos, são dados ao usuário controles interativos no espaço da imagem e no espaço do mundo que podem ser aplicados em diferentes combinações. Os pontos $2D$ da tela podem ser controlados pelo usuário movendo-os para uma nova posição, e então todas as mudanças necessárias aos parâmetros da câmera são calculadas de maneira que os pontos $3D$ correspondentes sejam projetados na nova posição especificada. Com esta abordagem, não existe realmente necessidade de saber como cada parâmetro muda a visão da cena e nem quais parâmetros são responsáveis pela obtenção de um determinado efeito de câmera.

Uma visão genérica da técnica será exposta a seguir.

7.1.1 Controle de objetos gráficos visto como uma solução de equações diferenciais

Pela abordagem diferencial introduzida em (Gleicher, 1994), o controle de objetos gráficos deve ser visto como um problema de *Otimização diferencial com restrição*. (Para uma melhor compreensão do conceito de objetos gráficos ver (Gomes *et al.* , 1998)).

Segundo esta técnica, controla-se objetos gráficos especificando-se o que acontece aos valores de atributos selecionados. Estes atributos são chamados *controles*, e podem também ser manipulados diretamente.

Controles são definidos por funções do tipo

$$v_c = f(v_e), \tag{7.1}$$

onde v_c - vetor de valores dos controles, v_e - vetor de estado dos objetos e f - função que define os controles.

O *vetor de estado* é um conjunto de parâmetros com valores reais que descrevem a configuração do objeto. Os atributos devem ser determinados como funções destes parâmetros.

Não é simples resolver a equação $v_c = f(v_e)$ para v_e, dado v_c. Pois para determinar os valores do vetor de estado a partir dos valores dos controles, é necessário fazer uma inversão da função f, quase sempre não linear.

Então, ao invés de especificar valores para os controles, aqui será especificado como eles estão mudando no tempo. Para um determinado instante no tempo, dados o valor para o vetor de estado v_e neste instante e os valores desejados para a taxa de mudança dos controles, $\dot{v_c}$, pode-se calcular a taxa de mudança necessária para o vetor de estado, $\dot{v_e}$.

Para trabalhar com a variação no tempo dos controles e do vetor de estado é necessário derivar cada lado da equação 7.1 em relação ao tempo, tendo-se

$$\dot{v_c} = \frac{\mathrm{d}v_c}{\mathrm{d}t} = \frac{\mathrm{d}f(v_e)}{\mathrm{d}t}$$

Aplicando a regra da cadeia obtém-se

$$\dot{v_c} = \frac{\partial f}{\partial v_e}\frac{\mathrm{d}v_e}{\mathrm{d}t}$$

De forma genérica, a derivada da função que define os controles em relação ao vetor de estado, $\partial f/\partial v_e$, é uma matriz chamada *Jacobiano*, e será representada por $\mathbf{J}$. Com esta notação, a expressão acima fica

$$\dot{v_c} = \mathbf{J}\dot{v_e} \tag{7.2}$$

Com isso, a relação não linear da eq. 7.1 fica substituída por uma relação linear dada pela eq. 7.2.

No entanto, existem vários valores para $\dot{v_e}$ que satisfazem o valor desejado para $\dot{v_c}$. Precisa-se determinar uma forma de variação adequada para estes parâmetros. A escolhida foi minimizar a taxa de variação das variáveis.

O valor de $\dot{v_e}$ de menor magnitude é o que está na mesma direção do gradiente de f. Qualquer componente de $\dot{v_e}$, não ao longo desta linha, não vai estar ajudando a atingir os controles desejados. Para determinar este valor particular de $\dot{v_e}$, deve-se resolver o problema

$$\text{minimizar } \; E = g(\dot{v_e}) \text{ , sujeito a } \; \dot{v_c} = \mathbf{J}\dot{v_e},$$

onde E é uma função objetivo de $\dot{v_e}$, sujeito à restrição linear a que os controles estão submetidos.

Otimização com Restrição

Os problemas de otimização com restrição constituem uma categoria grande de problemas na matemática para os quais existem uma gama de técnicas que podem ser aplicadas. Uma delas é o método de *Multiplicadores de Lagrange*.

Usando o cálculo diferencial para minimizar ou maximizar uma função sujeita a uma restrição, pode-se dizer que: dada uma função $g(x)$ sujeita a uma restrição $f(x)$, uma nova função pode ser formada igualando-se a zero a restrição, multiplicando-a por λ (o multiplicador de Lagrange) e somando o produto à função original. Assim

$$g(x, \lambda) = g(x) + \lambda f(x),$$

onde $g(x, \lambda)$ é a função de Lagrange, $g(x)$ é a função original ou função objetivo e $f(x)$ é a restrição.

O multiplicador de Lagrange, λ, aproxima o efeito, na função objetivo, da variação de uma unidade na constante da função restrição. Se λ é positivo, para cada acréscimo de uma unidade (decréscimo) na constante da função restrição, a função objetivo sofrerá um decréscimo (acréscimo) de um valor aproximadamente igual a λ. Se λ é negativo, para cada acréscimo (decréscimo) na constante da função restrição, a função objetivo sofrerá um acréscimo (decréscimo) de um valor aproximadamente igual a λ.

Voltando à notação anterior, com base no exposto acima, pode-se restringir $\dot{v_e}$ para ser múltiplo do gradiente, ou seja, $\dot{v_e}$ será expresso como um fator de escala vezes o gradiente.

Se existirem múltiplos controles, cada um dará uma contribuição para $\dot{v}_e$. Para cada controle a contribuição é algum múltiplo do seu gradiente. Tem-se então um vetor de multiplicadores de Lagrange, λ.

Associando ao vetor de estado os parâmetros da câmera, e os pontos da imagem aos valores dos controles, fica formulado o problema de calcular derivadas no tempo dos parâmetros da câmera como uma simples otimização com restrição. Obtidas as derivadas, e usando-se estes valores para atualizar os parametros da câmera, o problema se reduz a resolver uma equação diferencial de primeira-ordem com valor inicial.

A Seção 7.2 mostra a abordagem diferencial na solução específica do controle de câmera.

7.2 Through-the-Lens Camera Control

Primeiramente, deve-se definir a relação entre um ponto no sistema de coordenadas do mundo e sua projeção no espaço da imagem, expressa em termos de um modelo de câmera genérico.

Seja $p \in R^2$ um ponto no espaço da imagem, suas coordenadas são dadas por

$$p = h(Vx), \tag{7.3}$$

onde $x \in R^4$ é o ponto, em coordenadas homogêneas, no sistema de coordenadas do mundo que é projetado em p. V é a matriz de visualização, em coordenadas homogêneas, representando a combinação das transformações geométricas e projetivas. $h{:}R^4 \rightarrow R^2$, é a função que converte o ponto transformado Vx, de coordenadas homogêneas para coordenadas $2D$ da imagem, dada por:

$$h(Vx) = \left[\frac{Vx_1}{Vx_4}, \frac{Vx_2}{Vx_4}\right], \tag{7.4}$$

A matriz V é função dos parâmetros de câmera do modelo escolhido. Estes parâmetros passarão a ser referenciados como um vetor $\mathbf{c}$, de tamanho n.

Observe que h é uma função não linear e, portanto, a eq. 7.3 estabelece uma relação não linear entre o ponto p e os parâmetros de câmera $\mathbf{c}$, não sendo simples determinar V dado p, a partir desta equação, isto é, determinar a matriz de visualização a partir de um ponto da imagem.

Movimento da câmera a partir da velocidade de um ponto

Considere $x \in R^4$ um ponto fixo no espaço do mundo e p, como definido anteriormente, uma função dos parâmetros da câmera $\mathbf{c}$. Pela abordagem diferencial exposta na Seção 7.1.1, precisa-se obter a variação dos controles no tempo. A expressão para a velocidade do ponto, $\dot{p}$, é obtida aplicando-se a regra da cadeia à eq. 7.3:

$$\dot{p} = h^{'}(Vx)\left(\frac{\partial Vx}{\partial \mathbf{c}}\right)\dot{\mathbf{c}}, \tag{7.5}$$

onde

$h^{'}(Vx)$ é a matriz que representa a derivada da função $h(Vx)$, dada por

$$h^{'}(Vx) = \begin{bmatrix} \frac{1}{Vx_4} & 0 & 0 & -\frac{Vx_1}{Vx_4^2} \\ 0 & \frac{1}{Vx_4} & 0 & -\frac{Vx_2}{Vx_4^2} \end{bmatrix} \tag{7.6}$$

$\partial(Vx)/\partial(\mathbf{c})$ é a matriz 4 x n que representa a derivada do ponto transformado Vx em relação a $\mathbf{c}$, parâmetros da câmera, e $\dot{\mathbf{c}}$ é a derivada no tempo de $\mathbf{c}$.

Pela eq. 7.2 associada a eq. 7.5, o jacobiano $\mathbf{J}$ fica definido por uma matriz 2 x n, igual a

$$\mathbf{J} = h^{'}(Vx)\frac{\partial(Vx)}{\partial \mathbf{c}}, \tag{7.7}$$

considerando-se que o controle é um único ponto $2D$ no espaço da imagem, de coordenadas (x, y).

A eq. 7.5 fica reescrita como

$$\dot{p} = \mathbf{J}\dot{\mathbf{c}}. \tag{7.8}$$

Para m pontos na imagem, o jacobiano será uma matriz $2m$ x n, sendo n o número de parâmetros da câmera, que serão explicitados na Seção 7.2.1.

Agora que está definido que o ponto p, $2D$, da imagem é o vetor de valores dos controles, a função que define estes controles é a eq. 7.3, os parâmetros de câmera são o vetor de estado e a relação linear entre as suas velocidades está estabelecida pela eq. 7.5, pode-se então formular a solução de otimização diferencial usando a técnica dos multiplicadores de Lagrange, apresentada na Seção 7.1.1.

Recalculando os parâmetros da câmera

Existem vários valores para $\mathbf{c}$ que podem determinar o mesmo valor para p. Recaindo então no problema de otimização com restrição, iremos minimizar a taxa de variação dos parâmetros da câmera, a partir de um valor inicial de $\dot{p}$ (velocidade inicial do ponto na imagem) que será denotado por $\dot{p_0}$. Este valor deverá ser fornecido inicialmente ao sistema.

Então trata-se de minimizar a magnitude de $\dot{\mathbf{c}}$ a partir de um valor $\dot{c_0}$. A função objetivo será

$$E = \frac{1}{2}(\dot{c} - \dot{c_0}) \cdot (\dot{c} - \dot{c_0}),$$

que é a integral de

$$\frac{\mathrm{d}E}{\mathrm{d}\dot{c}} = \dot{c} - \dot{c_0}, \text{ sujeita a } \dot{p} - \dot{p_0} = 0.$$

Com isso estaremos minimizando a taxa de variação dos parâmetros da câmera. $\dot{\mathbf{c}}$ deve atender a restrição $\dot{p_0} = \mathbf{J}\ \dot{\mathbf{c}}$.

Teorema 1 *Supondo que x^* é um ponto de extremo local de f sujeito a $h(x) = 0$ e que x^* é um ponto regular destas restrições, então existe um $\lambda \in R^m$ tal que*

$$\nabla f(x^*) + \lambda \nabla h(x^*)^T = 0, \tag{7.9}$$

onde $\nabla h(x^*)^T$ é a matriz m x n definida por

$$\nabla h(x^*) = \left[\frac{\partial h_i(x)}{\partial x_j}\right]$$

Pelo teorema 1, $\nabla h(x) = \mathbf{J}$ e $\nabla f(x) = \frac{\mathrm{d}E}{\mathrm{d}\dot{c}} = \dot{c} - \dot{c_0}$. Portanto,

$$\dot{c} - \dot{c_0} = \mathbf{J}^T \lambda, \tag{7.10}$$

onde λ é o vetor $2D$ dos multiplicadores de Lagrange. Donde conclui-se que a taxa de variação dos parâmetros da câmera é obtida por:

$$\dot{c} = \dot{c_0} + \mathbf{J}^T \lambda \tag{7.11}$$

Mas para calcular $\dot{\mathbf{c}}$, precisa-se antes determinar λ. A eq. 7.10 diz que o gradiente de E deve ser uma combinação linear dos gradientes das restrições. Portanto, multiplicando-se esta equação por $\mathbf{J}$, tem-se:

$$\mathbf{J}\dot{c} - \mathbf{J}\dot{c_0} = \mathbf{J}\mathbf{J}^T \lambda$$

Aplicando-se a restrição, temos:

$$\dot{c_0} - \mathbf{J}\dot{c_0} = \mathbf{J}\mathbf{J}^T \lambda \tag{7.12}$$

e portanto, resolvendo-se este sistema de equações lineares, determina-se λ.

Obtendo-se então os valores das variações de $\dot{\mathbf{c}}$ pela eq. 7.11, deve-se atualizar os valores da câmera $\mathbf{c}$. O que recai em um problema de equação diferencial com valor inicial, cuja solução pode ser obtida a partir da aplicação de diferentes métodos numéricos.

A taxa inicial $\dot{c_0}$ é sugerida em (Gleicher & Witkin, 1992), como função de coordenadas do mouse, o que foi denominado pelos autores de *soft control*, sendo obtida de

$$\dot{c_0} = k_c \mathbf{J}^T (p_c - p),$$

onde p_c é a posição do mouse e k_c é uma constante.

Todo o processo de cálculo foi descrito até aqui usando uma matriz de visualização $\mathbf{V}$ genérica, independente do modelo de câmera. A seguir será descrito o modelo de câmera utilizado.

7.2.1 Modelo de câmera de quatérnio

Uma característica importante em through-the-lens camera control é dar uma nova conotação ao papel desempenhado pela parametrização da câmera. Esses parâmetros não são mais os controles. Isso permite uma liberdade de escolha para o modelo de câmera com bases no comportamento numérico ou por conveniências na implementação deste modelo.

O modelo de câmera escolhido foi baseado em rotações de quatérnio. Este modelo é pobre para os critérios convencionais, pois os quatro componentes do quatérnio podem tornar-se muito difícil para controlá-los diretamente. Por outro lado, ele é ideal para esta técnica porque permite liberdade nas rotações de câmera, sem singularidades. Além disso o método evita as interpolações de quatérnio tão conhecidas pelas suas dificuldades.

Uma breve explicação do que é quatérnio faz-se necessário.

Quatérnio

Os números complexos têm a forma $c = a + b\ i$, onde a e b são números reais, e as seguintes multiplicações são definidas:

$$1^2 = 1 \ \ e \ \ i^2 = -1.$$

Este conjunto de números define um plano onde um eixo é real e o outro é imaginário.

O espaço das rotações no plano é representado pelo círculo unitário $S^1 = \{x \in R^2; |x| = 1\}$. Um ponto do círculo unitário pode ser escrito sob a forma de um número complexo. Com alguns cálculos pode-se provar que as operações com rotações do plano podem ser reduzidas a operações com números complexos unitários.

Quatérnio é um elemento matemático cuja notação é:

$$\mathbf{q} = q_x + q_y + q_z + q_w,$$

$\mathbf{q} \in R^4$, onde q_w é chamado a *parte real* do quatérnio e q_x, q_y e q_z é a *parte vetorial* com eixos x, y e z. Pode-se ter a seguinte notação compacta: $\mathbf{q} = (q_w, \vec{q})$.

O quatérnio tem no R^4 uma estrutura de multiplicação semelhante à dos números complexos no R^2. Considerando a base canônica do R^4: $\vec{1} = (1,0,0,0)$, $\vec{i} = (0,1,0,0)$, $\vec{j} = (0,0,1,0)$ e $\vec{k} = (0,0,0,1)$, pode-se identificar o subespaço $R \cdot \vec{1}$ com o conjunto dos números reais e o subespaço $R \cdot \vec{x} + R \cdot \vec{y} + R \cdot \vec{z}$ com o espaço euclidiano R^3, que é chamado de *espaço dos quatérnios puros*.

Um quatérnio de norma 1 é chamado de *quatérnio unitário*. O conjunto dos quatérnios unitários é a esfera unitária S^3 do espaço R^4, $S^3 = \{x \in R^4; |x| = 1\}$. O espaço das rotações no R^3 é representado por uma esfera unitária. Tem-se então uma analogia entre números complexos e rotações no R^2 com quatérnios e rotações no R^4. Para que um quatérnio represente uma rotação tem-se a seguinte restrição $|\mathbf{q}| = \sqrt{q_x^2 + q_y^2 + q_z^2 + q_w^2} = 1$.

Para um estudo mais detalhado, demonstração dos resultados apresentados sobre quatérnios e rotações e dedução da matriz de quatérnio utilizada para rotações da câmera ver (Gomes & Velho, 1998).

A câmera de quatérnio

A matriz de visualização **V** é utilizada na definição do controle de câmera; o modelo escolhido usa os parâmetros de translação t_x, t_y e t_z para especificar a posição da câmera no espaço de coordenadas do mundo, e os parâmetros do quatérnio para dar orientação aos eixos do sistema da câmera. A matriz de projeção coloca a câmera a uma distância f do plano de projeção, ao longo do eixo z, paralela ao plano xy e apontando para a origem, conforme figura 7.1. O vetor de parâmetros da câmera, **c**, fica portanto caracterizado por um vetor com 8 elementos $(f, t_x, t_y, t_z, q_x, q_y, q_z, q_w)$.

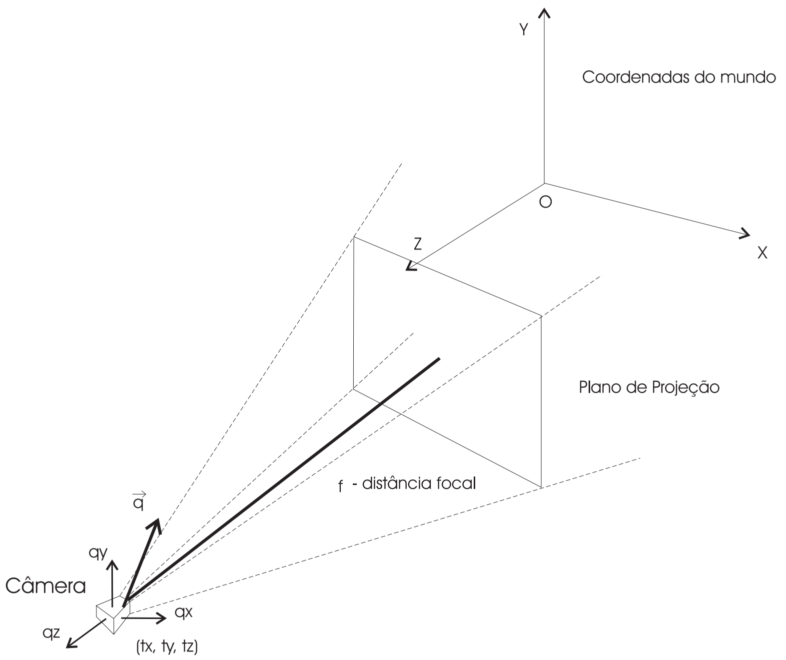


Figura 7.1: Câmera de Quatérnio

Especificando-se a matrix **V**, ela é a combinação da matriz de projeção **P**, matriz de translação **T** e **Q**, que é a matriz de *rotação de quatérnio*.

V:$R^4 \rightarrow R^4$, é dada pelo produto das transformações

$$V = P\ T\ Q, \tag{7.13}$$

a matriz de projeção $\mathbf{P}(f)$, com distância focal f, é dada por

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/\mathrm{f} & 0 \end{bmatrix}$$

$\mathbf{T}(t_x, t_y, t_z)$ a matriz de translação,

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

e a matriz $\mathbf{Q}(q_x, q_y, q_z, q_w)$ é composta dos seguintes elementos

$$\mathbf{Q} = 2 \begin{bmatrix} \frac{|q|^2}{2} - q_y^2 - q_z^2 & q_x q_y + q_w q_z & q_x q_z - q_w q_y & 0 \\ q_x q_y - q_w q_z & \frac{|q|^2}{2} - q_x^2 - q_z^2 & q_w q_x + q_y q_z & 0 \\ q_w q_y + q_x q_z & q_y q_z - q_w q_x & \frac{|q|^2}{2} - q_x^2 - q_y^2 & 0 \\ 0 & 0 & 0 & \frac{|q|^2}{2} \end{bmatrix}$$

Durante o movimento da câmera os valores obtidos para os quatérnios não correspondem a quatérnios puros, para isto o quatérnio precisa ser normalizado. Na integração numérica não é suficiente normalizar o quatérnio entre as iterações, pois as derivadas não levariam em conta a restrição de norma igual a 1 e geraria valores incorretos. Para resolver este problema, a normalização do quatérnio foi embutida na própria matriz de quatérnio $\mathbf{Q}$ (Gleicher & Witkin, 1992). É por isso que a matriz de quatérnio apresentada difere da matriz de quatérnio definida em (Gomes & Velho, 1998) e (Shoemake, 1985).

Com a matriz $\mathbf{V}$ calculada, determina-se o jacobiano pela eq. 7.7, derivando-se o vetor transformado Vx em relação a cada parâmetro da câmera $\mathbf{c}$. Para um ponto de controle, o jacobiano é uma matriz 2 x 8.

Determinado o jacobiano, calcula-se os multiplicadores de Lagrange através de um método de soluções de sistemas lineares. Com isso, as taxas de variação da câmera podem ser calculadas. Os novos parâmetros da câmera no espaço são resultantes do processo de integração numérica.

O problema de controle de câmera fica então formulado sob uma abordagem diferencial: dadas as derivadas no tempo de pontos da imagem, determinam-se as derivadas no tempo dos parâmetros da câmera. Com esta solução, a inversão da função f é evitada.

As áreas de fotogrametria, visão computacional e robótica tratam problemas que envolvem a recuperação de parâmetros da câmera a partir de medidas da imagem fazendo a inversão destes parâmetros. Isto exige uma calibração de câmera a cada nova posição, daí a dificuldade de obter movimentos contínuos da câmera utilizando-se estes recursos.

O uso de controle diferencial não tem como objetivo posicionar diretamente a câmera em saltos globais, e sim prover meios precisos e robustos de transformar ajustes contínuos dos controles em movimentos contínuos da câmera.

7.3 Implementação

O experimento realizado na dissertação consiste em arrastar a projeção p na imagem, de um ponto x do espaço $3D$, especificando-se sua velocidade $\dot{p}_0$ a cada instante, ao invés de fornecer a posição final desejada. A partir desta informação determina-se a nova posição da câmera no espaço que corresponde à nova projeção do ponto na imagem.

O ponto da imagem está sob controle iterativo e a velocidade inicial $\dot{p}_0$ é obtida do movimento de arrasto do mouse. O *soft control* foi implementado para ajudar a prender o ponto na trajetória determinada.

A tela da Figura 7.2 mostra o deslocamento do ponto no plano da imagem após o arrasto. Existe um ponto *dummy* - ponto fictício - que é inicialmente projetado na mesma posição do ponto controle. Este dummy é arrastado para que a câmera se ajuste no espaço $3D$ e projete o ponto de acordo com a trajetória determinada pelo dummy. Durante o deslocamento a tela mostra um pequeno vetor, no centro do polígono, que indica a direção do vetor velocidade.

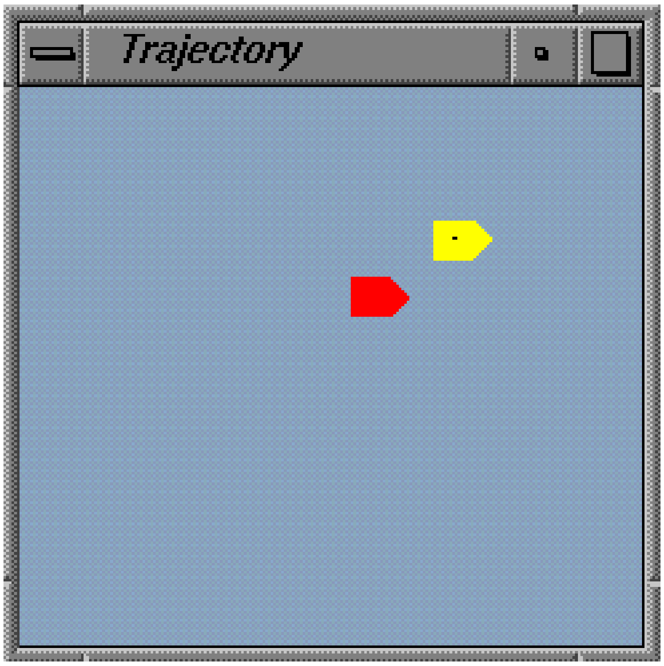


Figura 7.2: Trajetória do ponto no plano da imagem

O polígono serve para auxiliar a visualização do ponto. O controle é o centro do polígono e os demais pontos são desenhados no plano da imagem, em função do controle. Por isso o polígono se desloca sobre o plano de projeção sem sofrer deformações.

A tela da Figura 7.3 mostra um cubo gerado a partir do ponto x do espaço, que é um vértice do cubo.

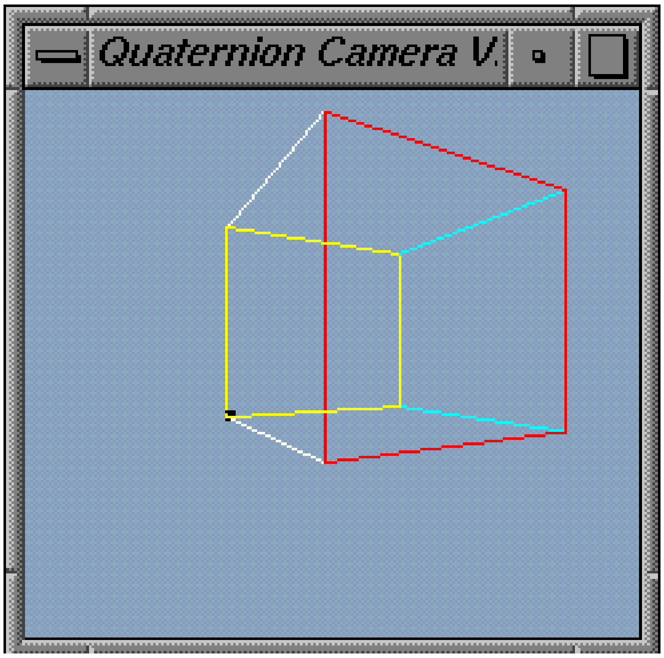


Figura 7.3: Projeção da Câmera de quatérnio

O cubo é calculado no espaço $3D$ e projetado pela câmera de quatérnio ao longo da trajetória. Todos os vértices do cubo sofrem as transformações da câmera, mas apenas o controle p, projeção de x, contribui para o jacobiano. Isso faz com que o cubo mostre, através de sua projeção, os movimenos que a câmera faz no espaço para ajustar-se à projeção desejada. É possível ver as rotações, deslocamentos e variações de foco feitas pela câmera.

O ponto dummy nesta tela está representado por um ponto preto coincidente com o ponto controle p.

A tela da Figura 7.4 mostra a origem do sistema de coordenadas do mundo, a câmera de quatérnio, seu plano de projeção com a projeção do cubo, o ponto x do espaço, e a linha de visão ligando a câmera ao ponto, vistos por uma outra câmera. Nesta janela pode-se acompanhar o movimento da câmera no espaço para ajustar sua projeção à trajetória.

O método utilizado para solução do sistema de equações lineares foi *Gauss-Jordan*, que determina os multiplicadores de Lagrange. O método *Runge-Kutta* de 4^a ordem foi utilizado para a integração numérica necessária no cálculo dos novos parâmetros de câmera, por ser suficientemente estável.

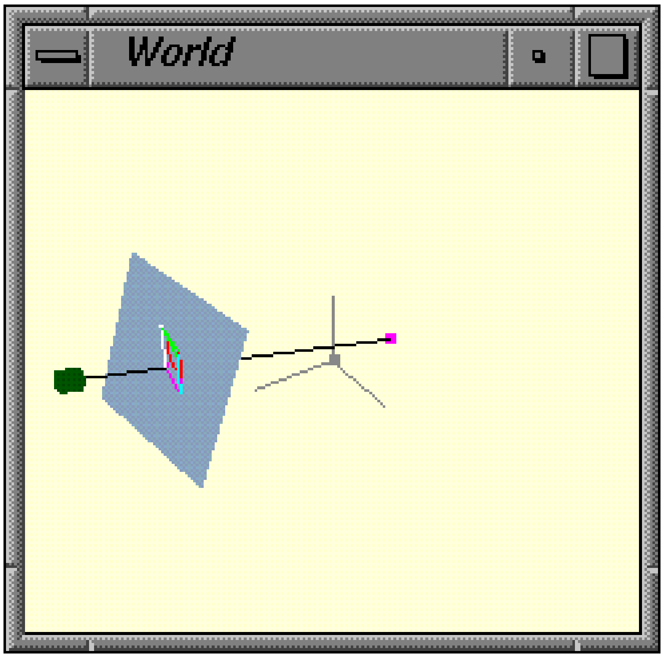


Figura 7.4: Movimento da câmera no espaço

Foi utilizada a linguagem *C* para programação e *OpenGL* para recursos gráficos, com o uso da biblioteca *Glut* para gerenciamento de janelas. O hardware utilizado foi uma estação Iris Indigo Silicon Graphics.

O controle dos parâmetros da câmera efetuado pela técnica de otimização diferencial apresenta duas grandes vantagens: lineariza o problema de otimização e calibra a câmera de forma incremental, sendo pois, adequada para movimentos contínuos de câmera.

Pelos resultados obtidos, constata-se que é bastante relevante a extensão deste estudo, ampliando-se o número de pontos de controle da câmera, visando atingir uma técnica de sincronismo de câmera baseada em tracking algorítmico.

Capítulo 8

Contribuições e Conclusões

Este capítulo resume as principais contribuições apresentadas ao longo do trabalho, faz conclusões e dá diretrizes para desenvolvimentos de novos trabalhos relacionados.

A dissertação foi desenvolvida com dois objetivos:

- apresentar uma conceituação de cenários virtuais gerando uma documentação para suprir a escassez de literatura técnica nesta área. Posicionar cenários virtuais como uma área de pesquisa e mostrar as fronteiras com áreas correlatas. Identificar os principais problemas destes sistemas dentro da computação gráfica, mostrar as soluções existentes e problemas em aberto.

- realizar um estudo de caso baseado em otimização diferencial visando chegar a uma técnica para sincronismo de câmera, que é parte integrante e de grande importância em um sistema para cenários virtuais.

Os objetivos foram atingidos através das pesquisas realizadas porque a área de cenários virtuais ficou caracterizada, teve seus conceitos bem definidos, o que permitiu obter uma visão global da área. Dentro do contexto da computação gráfica, foram identificadas e discutidas todas as tecnologias relevantes para cenografia virtual. Ficou constatada a importância do sincronismo de câmera nestes sistemas e foi proposto a extensão do estudo realizado como uma nova tecnologia para sincronismo de câmera.

Primeiramente, o trabalho contribui com uma análise entre áreas correlatas que mostrou as diferenças tecnológicas entre cada linha de pesquisa, ressaltou os pontos em comum e destacou as fronteiras existentes, deixando claro que cenários virtuais constitui uma área de pesquisa. Na literatura disponível esta conceituação está confusa e as denominações se sobrepõem em trabalhos com características técnicas diferentes e pontos em comum. Muitas pesquisas aplicáveis a cenários virtuais são denominadas realidade aumentada aplicada ao entretenimento, e não fazem distinção na característica de imersão que diferencia as duas áreas.

Uma vez esclarecido o que é cenário virtual, foi apresentada uma conceituação de sistemas de cenários virtuais, inexistente na literatura, com uma descrição específica, que mostrou quais os usuários, as demandas, objetivos e vantagens destes sistemas.

O estudo conceitual da área permitiu identificar as três tecnologias relevantes no uso de computação gráfica em cenografia virtual, que são: Rendering, Composição de Imagens e Sincronismo de câmera. Estes componentes são subsistemas que desempenham funções importantes e determinantes dentro de um sistema para cenários virtuais. Com isso, muitos trabalhos que não estão direcionados para cenários virtuais podem representar soluções para estes sistemas e em contrapartida, soluções dadas a esses sistemas servem também para solucionar problemas das áreas correlatas, adaptando-se às necessidades de cada abordagem.

Para cada tecnologia identificada, o trabalho apresentou os principais problemas e aspectos relevantes tecendo críticas e conclusões.

Em rendering ficou claro a necessidade de fotorealismo e resultados em tempo real. Das duas abordagens apresentadas, sugeriu-se o uso de Image-based rendering para a área de cenários virtuais, aliado às técnicas de Model-based rendering, processados em tempo real. Uma técnica híbrida que permitirá a reprodução de ambientes reais e famosos, ampliando o enfoque de cenografia virtual.

Depois o trabalho descreveu a evolução das técnicas de composição de imagem, dando um embasamento dos conceitos e necessidades que levaram às técnicas de cenário virtual atuais, deixando claro o desenvolvimento tecnológico que a distingue das técnicas tradicionais de bluescreen. O estudo esclarece o uso de duas tecnologias para composição: decomposição frente-fundo e composição por profundidade. Concluiu-se que para cenários virtuais, a combinação destas duas técnicas se faz necessário para ampliar os resultados na composição e resolver o problema de oclusão.

Na área de sincronismo constatou-se que os conceitos de calibração e tracking encontrados na literatura estão apresentados de forma confusa. A terminologia usada para cenários virtuais não está de acordo com a usada nos trackings aplicados à realidade virtual, e estes não são adequados para estúdios de cenários virtuais devido à proximidade de luzes e equipamentos de vídeo, causando problemas de imprecisão e instabilidade.

Foi então proposta uma denominação para os trackings: **tracking monitorado** - com uso de qualquer tipo de sensor (ótico ou mecânico) para informar coordenadas de posição da câmera - e **tracking algorítmico** para as soluções por algoritmo e que usam câmeras comuns para a filmagem.

A análise feita na dissertação permitiu avaliar os recursos existentes para sincronismo de câmera usados em cenários virtuais e também levou a concluir que é fundamental encontrar uma solução para o sincronismo de câmera que seja incremental, pois as técnicas que dependem de uma calibração a cada instante geram problemas de continuidade no movimento, por ser a calibração um processo estático. Ficou claro que o sincronismo de câmera é um problema ainda não totalmente resolvido e que desempenha papel fundamental na habilidade destes sistemas.

Isto motivou a efetuar um estudo de caso de uma técnica de otimização diferencial aplicada a controle de câmera virtual apresentada em (Gleicher & Witkin, 1992) e (Gleicher, 1994), com a visão de estendê-la a uma técnica incremental para sincronismo de câmera. Foi feito um experimento desta técnica para um ponto de controle. Os resultados obtidos para o movimento contínuo de câmera sob o controle de um ponto da imagem foram bons, embora um único ponto não seja suficiente para o controle efetivo de câmera.

A partir do experimento realizado, propõe-se a extensão deste estudo para que este método possa vir a efetuar um sincronismo de câmeras real e virtual, através deste controle diferencial. Além disso, para cenários virtuais isto caracteriza uma nova solução nos sistemas de sincronismo, já que os existentes utilizam sensores acoplados à câmera ou técnicas de reconhecimento de padrões de imagens.

A implementação realizada usou linguagem C com um modelo de câmera próprio, e o ambiente de OpenGL foi usado para a interface e desenho. Este aspecto apresentou outra contribuição importante que é a integracao de uma técnica de controle de câmera com um ambiente de modelagem como o OpenGL, usando um modelo de câmera que não é o que está implementado no OpenGL, e sim uma câmera de quatérnio externa ao ambiente.

Com todo o aspecto dissertativo e global dado ao assunto apresentado, o trabalho contribui também para dar um conhecimento global necessário ao desenvolvimento de um sistema para cenários virtuais. É uma área promissora que envolve tecnologia e que exige renovação constante, aspecto muito importante em pesquisa. Além do mais, a sua importância ressalta-se também pela correlação com outras áreas da computação gráfica, pois solucionar um problema para cenários virtuais pode significar resolver um problema para a área médica através da realidade aumentada, por exemplo, ou para a melhoria das comunicações através da área de ambientes virtuais colaborativos. Por isso é muito importante que estas informações, apesar de estarem focando mais uma aplicação específica, sejam vistas de forma mais abrangente, podendo ser ponte para muitas outras pesquisas e soluções.

Diretrizes para problemas em aberto

Existem muitos problemas em aberto para cenários virtuais, que são limitações dos atuais sistemas existentes. Ao longo da dissertação estes problemas foram colocados. Tem-se a seguir os principais deles, apresentados de forma resumida.

1. Composição

    - Resolver o problema de oclusão adicionado à técnica de chroma key, para encontrar uma solução de composição mais genérica.
    - A própria técnica de chroma key ainda demanda soluções para determinar o canal alfa de imagens de vídeo como está discutido em (Smith & Blinn, 1996).
    - Permitir mais de um ator no cenário com profundidades de campo diferentes, tornando o tratamento da composição mais complexo. Este problema também está relacionado ao sincronismo, pois lida com o problema de fazer o tracking de mais de um ator ao mesmo tempo.
    - Interação de atores reais com atores virtuais, em tempo real. Este tratamento também pode ser enriquecido através do uso de *motion capture* para passar movimentos mais reais aos atores sintéticos. Outro recurso está na área de inteligência artificial, usada para dar comportamento ao ator virtual. Já existem trabalhos nesta linha, mas ainda não operam em tempo real. É importante ressaltar que a inclusão de um ator sintético em cena também sobrecarrega a área de rendering e controle da câmera virtual.

2. Rendering

    - As sombras dos atores aparecendo no cenário virtual é ainda um problema mal resolvido. Em algumas apresentações, o ator parece estar flutuando no cenário. Isto também está relacionado ao tratamento da profundidade na cena, sendo simultaneamente um problema de rendering (pela geração das sombras) e um problema de composição.
    - Aplicar Image-based rendering combinado com Model-based rendering para cenários virtuais. Esta é uma abordagem interessante que pode ser mais explorada para a reprodução de ambientes reais.

3. Sincronismo

    - Continuar o estudo da técnica "Through-the-Lens", estendendo-a para mais pontos de controle. Integrar o controle da câmera virtual com a câmera real para torná-la uma técnica incremental de sincronismo de câmera.
    - Ampliar a capacidade de movimento dos atores e da câmera real para efeitos mais complexos, mantendo sincronismo em tempo real.

Bibliografia

Albuquerque, Antonia Lucinelma Pessoa. 1998. Texturas em Computação Gráfica. *PUC-RioInf. MCC36/98 - Departamento de Informática, PUC-Rio.*

Bhatnagar, Devesh Kumar. 1993. Position trackers for Head Mounted Display systems: A survey. *Technical Report, University of North Carolina at Chapell Hill.*

Bourke, Paul. 1992. Computer Rendering and Visualisation Techniques - Examples. *http://www.mhri.edu.au/~pdb/rendering/rendervis/.*

Bowman, D. 1996. Conceptual Design Space - Beyond Walk-through to Immersive Design. *in Bertol, D., Designing Digital Space, John Wiley and Sons, New York.*

Carvalho, Paulo Cezar, Szenberg, Flavio, & Gattass, Marcelo. 1998. Image-based Modelling Using a Two-step Camera Calibration. *SIBGRAPI'98 e http://www.visgraf.impa.br/juizvirtual.*

Corp., Ultimate. 1993. Memory Head Operations Manual. *http://www.audiovideo.pt/ibertelco/Ultimatte/MemoryHead_i.html.*

Costa, B., Darsa, L., Gomes, J., Roma, P., Velho, L., & Zonenschein, R. 1996. Cenário Virtual e Composição de Imagens. *SIBGRAPI'96 - http://www.visgraf.impa.br/Projects/virtualset/virtualset.html.*

Crow, Franklin C. 1989. *Computer Graphics Techniques - Theory And Practice.* Rogers, David F. and Earnshaw, Rae A.; - Editors.

Cruz-Neira, C., Sandin, D., & DeFanti, T. 1992. The CAVE : A Virtual Reality Theater. *http://www.evl.uic.edu/pape/CAVE/oldCAVE/CAVE.html.*

Cruz-Neira, C., Sandin, D., & DeFanti, T. 1994. ImmersaDesk. *http://www.evl.uic.edu/EVL/VR/ImmersaDesk.shtml.*

Cruz-Neira, C., Sandin, D., & DeFanti, T. 1995. Virtual Reality Systems. *http://www.evl.uic.edu/EVL/VR/systems.shtml#idesk.*

Dare, Alan. 1998. Perspectives on Image Quality in the Onyx2TM InfiniteReality Graphics System. *Silicon Graphics, http://www.sgi.com/apps/geospatial_imaging/irquality_wp.html.*

Debevec, Paul, Taylor, Camillo, & Malik, Jitendra. 1996. Modelling and rendering architecture from photographs: A hybrid geometry and image-based approach. *SIGGRAPH'96.*

DeFanti, T., Foster, I., Papka, M. E., Stevens, R., & Kuhfuss, T. 1996. Overview of the I-WAY: Wide Area Visual Supercomputing. *International Journal of Supercomputing Applications, 10(2) - http://evlweb.eecs.uic.edu/EVL/RESEARCH/PAPERS/PAPKA/intro.html.*

Dévai, Frank. 1997. On the Computational Requirements of Virtual Reality Systems. *Eurographics'97, State of the Art.*

Erland, Jonathan, & Dorney, Roger. 1982. Reverse or Negative Bluescreen Traveling Matte Process. *Society's 124th Conference, New York - paper No. 124-58, SMPTE.*

Ettinger, G.J., Leventon, M.E., Grimson, W.E.L., Kikinis, R., Gugino, V., Cote, W., Sprung, L., Aglio, L., Shenton, M., Potts, G., & Alexander, E. 1997. Experimentation with a Transcranial Magnetic Stimulation System for Functional Brain Mapping. *In CVRMED/MRCAS, Grenoble, France, 1997. - http://www.ai.mit.edu/projects/medical-vision/surgery/surgical_navigation.html.*

Fearing, Paul. 1996a. Importance Ordering for Real-Time Depth of Field. *Proceedings of Third International Conference on Computer Science, held in Hong Kong, 11-13 of December.*

Fearing, Paul. 1996b. Predictive Rendering. *Master Thesis, University of British Columbia.*

Feiner, S., MacIntyre, B., & Seligmann, D. 1993. KARMA - Knowledge-based Augmented Reality for Maintenance Assistance. *Communications of the ACM, 36(7), July 1993, 52-62.*

Feiner, S., Webster, T., Krueger, T., MacIntyre, B., & Keller, Ed. 1995. Architectural anatomy. *In Presence, 4(3), Summer 1995, 318-325. - http://www.cs.columbia.edu/graphics/projects/archAnatomy/architecturalAnatomy.html.*

Fielding, Raymond. 1985. *The Technique of Special Effects Cinematography.* Focal Press.

Foley, J., van Dam, A., Feiner, S., & Hughes, J. 1992. *Computer Graphics Principles and Practice.* Addison-Wesley Publishing Company, Inc.

Fuchs, H., Whitton, Mary C., State, A., Livingston, M. A., Garrett, W. F., Hirota, G., & Pisano, E. D. 1996. Technologies for Augmented-Reality Systems: realizing Ultrasound-Guided Needle Biopsies. *Proceedings of SIGGRAPH 96 (New Orleans, LA, August 4-9, 1996). In Computer Graphics Proceedings, Annual Conference Series 1996, ACM SIGGRAPH, pgs. 439-446.*

Future-Reality. 1995. Discreet Logic - Virtual Sets. *http://www.future.com.au/dlnews/discreet_news_v1i6/page3.html.*

Gibbs, Simon, & Baudisch, Patrick. 1996. Interaction in the Virtual Studio. *German National Research Center for Information Technology.*

Gleicher, M., & Witkin, A. 1992. Through-the-Lens Camera Control. *SIGGRAPH'92.*

Gleicher, Michael. 1994. A Differential Approach to Graphical Interaction. *PhD Thesis, Carnegie Mellon University, CMU-CS-94-217.*

Gomes, J., & Velho, L. 1994. *Computação Gráfica: Imagem.* IMPA.

Gomes, J., & Velho, L. 1998. *Computação Gráfica, Volume 1.* Série de Computação e Matemática - IMPA.

Gomes, J., Velho, L., Costa, B., & Darsa, L. 1998. *Warping and Morphing of Graphical Objects.* Morgan Kaufmann Publishers, Inc.

Graphics, Silicon. 1998. Onyx2 System Hardware. *http://www.sgi.com/onyx2/sys_hardware.html.*

Hayashi, M., Fukui, K., & Itoh, Y. 1996. Image Compositing System Capable of Long-Range Camera Movement. *ACM Multimedia 96.*

Healey, Christopher G., Debevec, Paul, & Séquin, Carlo. 1998. Rendering Building Interiors. *http://www.cs.berkeley.edu/~healey/ibr/ibr.shtml.*

Hodges, L.F., Rothbaum, B.O., Kooper, R., Opdyke, D., Meyer, T., North, M., Graaff, J.J., & Williford, J. 1995. Virtual environments for treating the fear of heights. *IEEE Computer 28,7 (1995), pp. 27-34.*

Hoffman, Hunter. 1996. VR Treatment for Burn Pain. *University of Washington, HITLab - http://www.hitl.washington.edu/projects/burn/.*

Hoffman, Hunter. 1997. Desensitization in Virtual Environments. *University of Washington, HITLab - http://www.hitl.washington.edu/projects/exposure/.*

Honda, Y., Matsuda, K., Rekimoto, J., & Lea, R. 1995. Virtual Society: Extending the WWW to support a Multi-user Interactive Shared 3d Environment. *Sony Computer Science Laboratory Inc.*

Jancène, P., Meilhac, C., Neyret, F., Provot, X., Tarel, J. P., Vézien, J. M., & Verroust, A. 1996. Réalité Enrichie par Synthèse. *INRIA.*

Kanade, T., Oda, K., Yoshida, A., Tanaka, M., & Kano, H. 1996. Z-Key: A New Method for Creating Virtual Reality. *http://www.cs.cmu.edu/afs/cs/project/stereo-machine/www/z-key.html.*

Kruger, W., Frohlich, B., Wesche, G., & Strauss, W. 1995. The Responsive Workbench. *http://viswiz.gmd.de/VMSD/PAGES.en/projects.workbench.html.*

Lando, Maria. 1997. Motion Tracking for Special Effects in the Film Industry. *SIGGRAPH'97.*

Mandeville, J., Furness, T., Kawahata, M., Campbell, D., Danset, P., Dahl, A., Dauner, J., Davidson, J., Kandie, K., & Schwartz, P. 1995. GreenSpace: Creating a Distributed Virtual Environment for Global Applications. *IEEE Proceedings of the Networked Reality Workshop.*

Metzger, P. J. 1993. Adding Reality to the Virtual. *Proceedings of the IEEE 1993 Virtual Reality Annual International Symposium : 7-13.*

Milgram, Paul, Rastogi, Anu, & Grodski, Julius J. 1995. Augmented Telerobotic Control: a visual interface for unstructured environments. *http://vered.rose.utoronto.ca/people/anu_dir/papers/atc/atcDND.html.*

Mine, Mark, & Bishop, Gary. 1993. Just-in-Time Pixels. *Technical Report - University of North Carolina, Chapel Hill.*

Molnar, S., Eyles, J., & Poulton, J. 1992. Pixel Flow: High Speed Rendering Using Image Composition. *SIGGRAPH'92.*

Monalisa. 1994. Modelling Natural Images for Synthesis and Animation. *http://www.race.analysys.co.uk/race/pl4/overview/r2052.htm.*

Orad. 1995. Virtual Sets. *http://www.orad.co.il/virsets/index.htm.*

Oschatz, Sebastian. 1994. Grundlagen eines echtzeitfähigen Systems zur Verwendung virtueller Fernseh-Studio-Kulissen. *http://www-cui.darmstadt.gmd.de/visit/Activities/Vist/Diplomarbeit.oschatz/.*

Radamec, Broadcast Systems. 1995. Virtual Scenario Studio System. *http://website.lineone.net/~radamec_broadcast/virt1.html.*

Reinhard, E., Chalmers, A. G., & Jansen, F. W. 1998. Overview of Parallel Photo-realistic Graphics. *Eurographics'98, State of the Art.*

RT-Set. 1994. Virtual Studio Systems. *http://www.rtset.co.il/.*

Shoemake, Ken. 1985. Animating Rotations with Quaternion Curves. *Computer Graphics, 19(3):245-254.*

Siegmann, K. 1994. Virtual Sets. *IRIS Universe - Silicon Graphics.*

Silicon Graphics, Inc. 1994/95. Open Inventor. *http://www-europe.sgi.com/Technology/Inventor/.*

Silicon Graphics, Inc. 1997a. OpenGL. *http://www.sgi.com/software/opengl/manual.html.*

Silicon Graphics, Inc. 1997b. OpenGL-based Real-Time Shadows. *http://reality.sgi.com/opengl/tips/rts/.*

Smith, Alvy Ray. 1996. The Alpha Channel: A Simple Concept with Profound Implications. *http://www.microsoft.com/imagecomposer/usingic/alvy_2.htm.*

Smith, Alvy Ray, & Blinn, James F. 1996. Blue Screen Matting. *SIGGRAPH'96.*

Studios, Pixar Animations. 1997. PIXAR'S RENDERMAN. *http://www.pixar.com/products/renderman/toolkit/Toolkit/.*

Sutherland, Ivan. 1968. A Head-Mounted Three-Dimensional Display. *AFIPS Conference Proceedings, Vol. 33, Part I, 1968, pp. 757-764.*

Tommaselli, A. M., & Tozzi, Clésio Luís. 1991. Tutorial: Calibração de Câmeras usando Feições Genéricas. *SIBGRAPI'91.*

Vist. 1994. The Virtual Studio. *http://www-cui.darmstadt.gmd.de/visit/Activities/Vist/.*

Weghorst, Suzanne, & Prothero, Jerry. 1994. Parkinson's Project. *University of Washington, HITLab - http://www.hitl.washington.edu/projects/parkinsons/.*

Wright, Anthony Asher. 1997. Virtual Sets and Studios. *http://www.ashera.com/3D.html.*