

Buscando o Uso Operacional de Realidade Virtual em Grandes Modelos de Engenharia

Eduardo T. L. Corseuil¹, Alberto B. Raposo¹, Ismael H. F. dos Santos^{1,2},
Marcelo Gattass¹, Marcio H. G. Pinto¹

¹TeCGraf – Grupo de Tecnologia em Computação Gráfica/PUC-Rio

²CENPES – Centro de Pesquisas e Desenvolvimento da Petrobras

{thadeu, abraposo, ismael, gattass, mhgpinto}@tecgraf.puc-rio.br

Abstract. *This paper approaches the problems related to the use of Virtual Reality in large industrial engineering models coming from CAD (Computer Aided Design) tools. The main obstacles of that process are indicated and solutions developed to overcome some of those obstacles are presented. Some academic and commercial solutions are also analyzed, allowing us to have a sketch about the main problems that have to be solved in order to facilitate the use of the VR technology for companies that use large engineering models. In this work we analyze the necessary aspects to produce a virtual model, derived from the CAD model that allows the user to navigate through it in real-time and have a reasonable immersive sensation. Two Petrobras platform designs, P40 and P43, were used as models for the tests.*

Resumo. *Este artigo aborda os problemas relacionados ao uso de Realidade Virtual (RV) em grandes modelos industriais de engenharia provenientes de ferramentas CAD (Computer Aided Design). São identificados os principais obstáculos desse processo e apresentadas soluções próprias desenvolvidas para alguns desses obstáculos. Algumas soluções acadêmicas e comerciais também são analisadas, o que nos permitiu traçar um quadro representativo dos principais problemas encontrados para a viabilização do uso de RV em empresas que utilizam grandes modelos de engenharia. Neste trabalho analisamos os aspectos necessários para a criação de um modelo virtual, a partir do modelo CAD, que permita a sua navegação em tempo real com a devida sensação de imersão. Foram utilizados como modelo de testes os projetos das plataformas P40 e P43 da Petrobras.*

1. Introdução

A pesquisa relacionada a sistemas de Realidade Virtual (RV), que já teve o seu reconhecimento nas áreas de jogos e entretenimento e em aplicações militares, nos últimos anos passou a ser cada vez mais relacionada a outras indústrias. Áreas como Engenharia, Medicina e Geociências têm buscado soluções integradas com RV para melhor representação, percepção e análise do espaço físico envolvido.

Utilizam-se recursos de RV para visualizar desde camadas geológicas, forma e composição de reservatórios naturais, até grandes estruturas de engenharia, de forma completa ou parcial, inclusive com o detalhamento de algumas peças.

Em todas estas áreas o uso de RV permite um ganho nos processos de análise quando é conjugada com processos de simulação. A visualização em tempo real dos efeitos de uma mudança no modelo representado facilita a compreensão do processo. Pode-se, por exemplo, a partir de dados geo-

estatísticos, mostrar a evolução de seções geológicas ao longo do tempo e prever o comportamento destas seções, em uma simulação para o futuro. Modificações de projeto em estruturas já construídas também podem ser simuladas, verificando-se previamente a eficácia e a viabilidade da mudança.

Atualmente o volume de dados em sistemas onde se busca utilizar RV é considerável, sendo compartilhado por diferentes grupos de trabalho, que podem inclusive estar dispostos fisicamente em diferentes lugares [Singhal and Zyda 1999]. Nos últimos anos o uso de RV vem se estendendo através da criação de Ambientes Virtuais Colaborativos, cujo objetivo é permitir a integração de equipes que trabalhem de forma cooperativa sobre um mesmo projeto. A junção da Colaboração com a RV amplia a capacidade de compreensão destes grupos sobre o processo e os dados envolvidos.

Neste contexto, um dos grandes objetivos dos departamentos de engenharia de grandes empresas, como as da indústria automobilística, aeroespacial, de petróleo, e outras de tal porte, é a criação de sistemas de informação integrados para controlar os projetos de seus produtos. Buscam-se sistemas computacionais que, além de acessar a base de dados com informações de projeto, possuam recursos para visualização em 3D dos modelos com realismo suficiente para serem utilizados nas atividades de revisão de projeto (Design Review), manutenção de equipamentos (Change Management Systems) e treinamento de pessoal.

Sistemas de Realidade Virtual vêm sendo desenvolvidos para atender a estas finalidades. Para um sistema deste tipo, maior será a eficiência quanto maior for o grau de imersão apresentado e também quanto maior for a interatividade, o que requer uma resposta do sistema em tempo real às solicitações demandadas pelo usuário e/ou pelo modelo.

Os problemas encontrados neste processo derivam basicamente do fato de que o modelo de engenharia não é construído com a preocupação de ser utilizado para visualização em tempo real. Em alguns casos os modelos são representações visualmente simplificadas, servindo apenas como representação esquemática das características analisadas. Por outro lado, há outros casos em que os modelos são muito grandes e complexos para a visualização em tempo real. Este artigo tratará especificamente de modelos de CAD que se encaixam neste último grupo, onde muitos dos componentes, que são importantes no projeto de engenharia, não o são no contexto de RV. Mesmo os componentes que se aplicam à visualização, em grande parte precisam ser otimizados para isto, pois apresentam um detalhamento excessivo e caro para este processo.

Os principais pontos deste trabalho são: a) apresentar uma avaliação do problema, apontando os principais obstáculos do processo (seção 2); b) apresentar soluções desenvolvidas pelos autores para alguns desses obstáculos (seção 3); c) ilustrar outras soluções, acadêmicas e comerciais, reforçando os pontos sensíveis do processo (seção 4).

O presente trabalho foi realizado baseado no estudo de caso para a indústria de exploração e produção de petróleo: o projeto da P43 – uma estrutura tipo FPSO (Floating, Production, Storage and Offloading) da Petrobras – um modelo de plataforma de exploração que permite também o armazenamento do produto, pois é adaptada sobre um casco de um navio petroleiro (Figura 1). Os dados utilizados foram fornecidos pela Petrobras e gerados pelo PDS – Plant Design System [Intergraph 2003], sistema de projeto utilizado pela empresa. O conjunto de dados consiste em aproximadamente 1800 arquivos gráficos vetoriais (formatos DGN e PRP) representando componentes do projeto, e aproximadamente 1200 arquivos texto (formato DRV) com dados de caracterização dos componentes provenientes da base de dados, cada um correspondente a um respectivo arquivo vetorial. Este conjunto de dados possui aproximadamente 1Gb, gerando um modelo 3D com aproximadamente

20 milhões de polígonos quando exportado em arquivos no formato VRML. Este número varia de acordo com a solução adotada por cada sistema de Realidade Virtual analisado.

Para efeito de simplificação e melhor entendimento do texto, este trabalho denomina modelo de engenharia como “modelo CAD” e o modelo de visualização 3D para interação em tempo real como “modelo RV”.

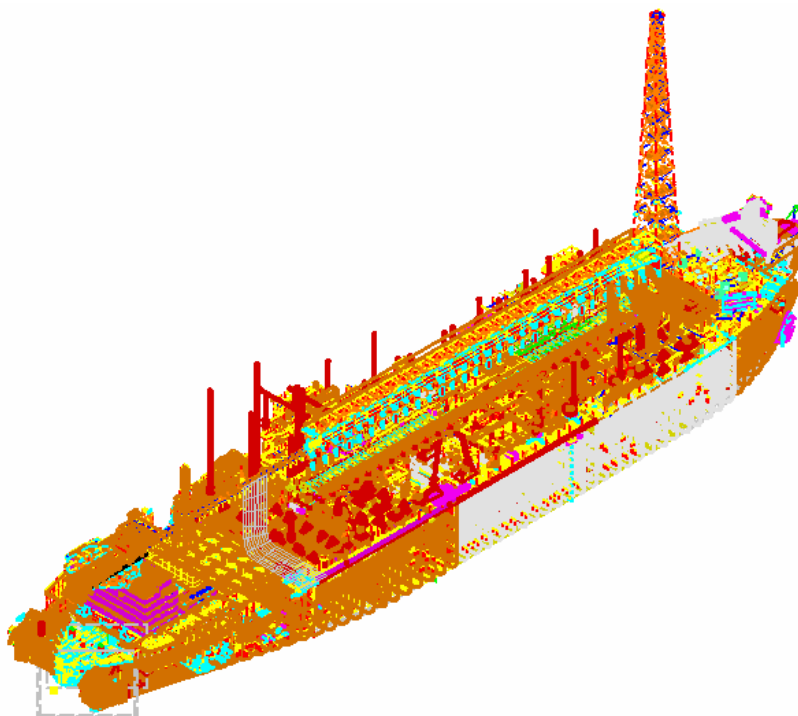


Figura 1. Modelo CAD da P43 (parte estrutural)

2. Problemas Comuns

Na busca de um caminho ótimo do processo de conversão “CAD para RV”, tanto por meio de soluções próprias como pelo uso de ferramentas comerciais foram detectados alguns obstáculos comuns no que diz respeito à visualização dos modelos. São eles:

- Baixo realismo – O modelo CAD geralmente não possui informações de textura associadas diretamente ao objeto. Aplicar texturas individualmente é um trabalho que não compensa dada a complexidade do modelo.
- Performance fraca para modelos de grande complexidade – o “frame rate” atingido é baixo quando um modelo do porte da P43 é carregado por completo e entra-se em locais de maior densidade de objetos. Os algoritmos de otimização testados ainda não são suficientes para a quantidade de objetos e detalhes em questão.
- Superfícies curvas complexas não são bem tratadas, em especial nas soluções comerciais – Algumas superfícies aparecem no modelo CAD na forma de NURBS (NonUniform Rational B-Splines), como o casco da plataforma, por exemplo. As malhas geradas para este tipo de superfície são muito ineficientes para visualização e demandam um tratamento especial.

Na verdade os problemas são originários da diversidade entre um modelo CAD e um modelo RV. Cada um atende a objetivos diferentes. No modelo CAD é realizado um detalhamento da forma para o processo de execução, sem levar em conta se isto vai dificultar a visualização 3D. Não existe

ainda um ambiente integrado onde se possa migrar do modelo CAD para o modelo RV e vice-versa, permitindo maiores interações para que ajustes necessários sejam feitos. Os processos de conversão existentes consistem em traduções diretas e imprecisas do CAD para o RV, e normalmente são fortemente dependentes do formato de CAD traduzido.

Na falta de um padrão de “CAD para RV”, as ferramentas de RV buscam soluções parciais, limitadas a um formato específico gerado pela ferramenta de CAD. Um exemplo da inadequação do modelo CAD para a visualização é o fato de que na maioria dos modelos CAD os objetos são agrupados de acordo com a função – por exemplo, um determinado tipo de equipamento. Na engenharia, a divisão temática por funcionalidade é fundamental. No entanto, para a visualização a localização é mais importante do que a função. É necessário um reagrupamento hierárquico dos objetos para alcançar melhor performance na visualização.

3. Soluções Apresentadas

Para atacar alguns dos problemas mencionados foram realizados esforços cujos resultados serão apresentados a seguir. Devido a dificuldade em se analisar os melhoramentos possíveis de serem empregados para diminuir os gargalos identificados nas soluções comerciais (“caixas-pretas”), optamos pelo desenvolvimento de um sistema próprio para a conversão do modelo CAD para o modelo RV.

No início buscou-se este caminho de uma forma direta, baseado apenas na conversão pura entre formatos existentes. No entanto, ficou evidente a necessidade de se otimizar os dados e o processo. Foi implementado então um sistema que está atualmente subdividido em dois módulos principais: um leitor de formato CAD – no caso do PDS o formato é o DGN – e um visualizador que recebe os dados exportados pelo leitor DGN. A Figura 2 ilustra como era o processo sem estas ferramentas e como ele está atualmente. Na primeira vez em que a P43 foi representada em um modelo RV, o processo foi feito via conversão para os formatos VRML e 3DS. Era um processo manual e com pequenas otimizações para a leitura destes formatos. Nas subseções seguintes serão detalhadas as etapas do processo atual.

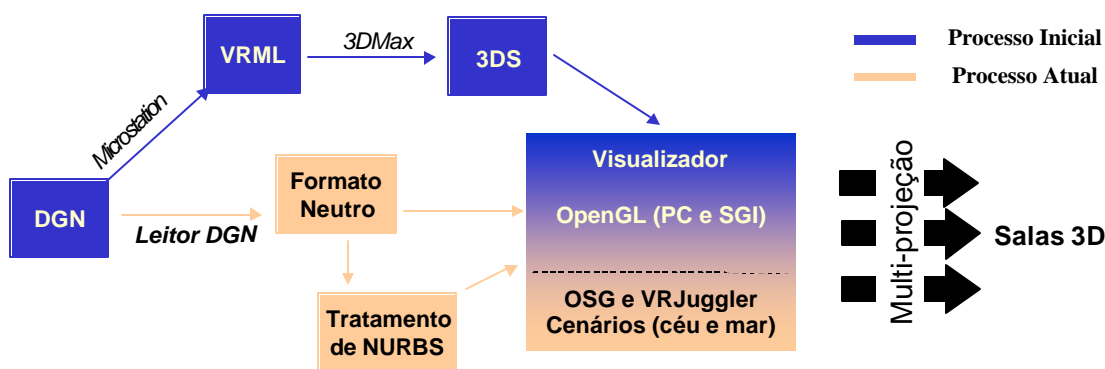


Figura 2. Pipeline de Visualização: CAD para RV

3.1. Leitor DGN

Um passo necessário em todo o processo é justamente obter um melhor entendimento do modelo CAD. Tendo como objetivo interpretar o modelo da P43, foi feito um estudo do formato DGN e de sua estruturação interna [Bentley 1995].

Paralelamente começou-se a estudar e testar ferramentas disponíveis para acesso ao arquivo DGN e captura dos dados via programação. O MicroStation versão 7.1 (compatível com o DGN gerado pelo PDS), oferece algumas APIs para acessar a estrutura interna de um modelo DGN. Primeiramente

foi utilizada uma biblioteca de macros disponível em linguagem basic (MicroStation Basic) [Bentley 2003a] para acessar alguns componentes dos modelos. Foram feitos alguns testes com sucesso e gerados arquivos de saída com dados descritivos dos componentes do modelo DGN. No entanto as macros mostraram limitações quando foi preciso acessar elementos complexos na estrutura do DGN, como alguns tipos de sólidos e grupos hierarquizados de objetos. Foi então alterada a ferramenta de desenvolvimento para a biblioteca MDL (MicroStation Development Language) [Bentley 1999a,b,c] em linguagem C. Com a MDL foi possível desvendar toda a estrutura do DGN. As principais dificuldades encontradas foram com a manipulação de sólidos gerados pelo ParaSolid [EDS 2003], que é o modelador de sólidos do MicroStation. A documentação escassa foi outro empecilho no processo. Mesmo assim, a grande maioria dos objetos da P43 foram traduzidos e exportados.

Também foram entendidas e estabelecidas relações entre as informações do arquivo DGN (parte gráfica) e as informações vindas da base de dados do sistema de projeto. Estes dados estão em arquivos à parte (formato DRV), relacionados por códigos identificadores aos elementos dos arquivos DGN.

3.2. Simplificação de NURBS

Na sua maior parte o processo de design é altamente otimizado, seguindo várias normas e tabelas para caracterizar objetos da planta de processo (Process Plant). O sistema utilizado pela Petrobras – PDS – é próprio para isto e modela bem várias partes do modelo. Justamente as partes que não são geradas pelo sistema, como o casco, é que apresentam necessidade de otimização para a visualização. Pôde-se constatar que a superfície do casco é gerada com número excessivo de curvas de formação, dificultando a conversão do modelo. Outras superfícies apresentam problemas semelhantes na conversão.

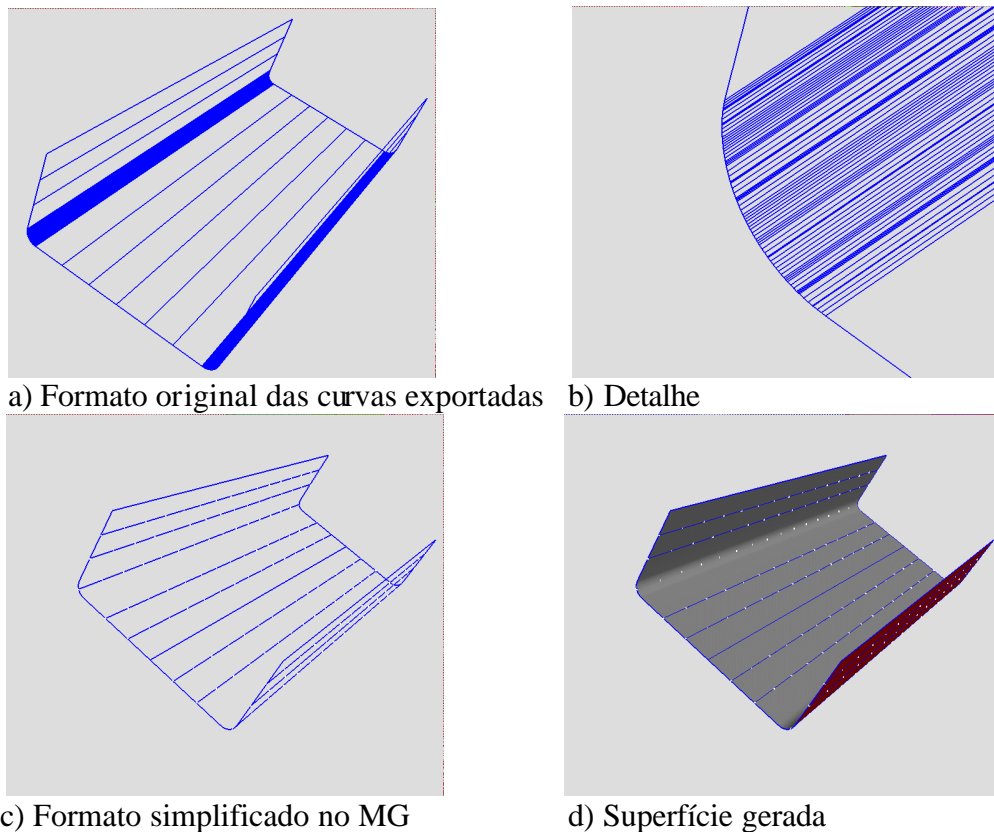


Figura 3. Casco da P43: processo de simplificação das curvas geradoras de superfície no MG

Para reduzir o problema de conversão de superfícies em geral – e em particular do casco da P43 – foi feita uma ponte entre o exportador DGN e a aplicação de visualização, utilizando-se uma outra aplicação que lê e edita superfícies, o MG (Mesh Generator) [Lira et al. 2002, TeCGraf 1996]. A superfície é carregada neste editor, e as curvas de geração são reduzidas substancialmente, para depois a superfície ser novamente exportada para o visualizador. Esta operação não altera a geometria da superfície para efeito de visualização, e o rendimento melhora muito (Figura 3).

Apesar de existirem funções no MicroStation para a simplificação e redefinição de superfícies (rebuild surface), elas nem sempre garantem o resultado desejado visto que o processo é de reconstrução total da superfície baseada em novos parâmetros. O MG oferece um controle maior sobre a superfície resultante na operação de redefinição, pois permite a edição separada das curvas que compõem a superfície. Desta forma, o processo de simplificação de superfícies, feito no MG, funciona como um pré-processamento para a visualização, mantendo inalterado o modelo CAD.

Independente do processo de simplificação da forma das superfícies e de qual o editor usado para isto, o problema da representação de superfícies do tipo NURBS na visualização foi tratado [Pie g and Tiller 1999] e foi utilizada no visualizador uma biblioteca para visualização de NURBS baseada no OpenGL (NURBS++) [Lavoie 1999]. Esta biblioteca permite a leitura de superfícies complexas no espaço parametrizado e gera as malhas que servem como base para a visualização.

3.3. Visualizador

Em alguns casos, pela dificuldade de se obter dados de formação de certos objetos, estes são exportados do leitor DGN para o visualizador diretamente na forma de malha de visualização. Como estas malhas são geradas diretamente pelas funções internas do MicroStation, não existe um controle de do grau de eficiência da malha. Analisando as malhas geradas, constatou-se que elas poderiam ser otimizadas para visualização.

Foi implementado e aplicado no visualizador um algoritmo de LOD (Levels-Of-Detail) com o objetivo de simplificar e otimizar as malhas para visualização. Para objetos baseados em primitivas simples como cilindros, esferas, linhas, curvas e extrusões e revolução dessas primitivas, o algoritmo de LOD consiste em transmitir para o visualizador as informações paramétricas de geração dessas formas (raio, posição e altura, no caso de um cilindro, por exemplo) e deixar que o visualizador, de acordo com a posição do objeto, gere a malha de triângulos necessária para a visualização. O uso de objetos paramétricos permite uma economia no consumo de memória – objetos longe o suficiente da câmera vão ocupar apenas o espaço necessário para guardar os poucos parâmetros de geração do objeto – e também elimina a necessidade de pré-processamento presente em outras técnicas. Por outro lado, o custo de CPU usado para a geração da malha ideal em cada quadro pode tornar o processo inviável.

Alguns objetos não têm uma representação paramétrica fácil de se trabalhar. Nestes casos o algoritmo implementado consiste em gerar diferentes níveis de detalhamento para a malha de cada objeto, cada um com a quantidade certa de triângulos para uma determinada distância da câmera. A cada quadro, é selecionada a melhor representação, ou seja, a representação com a menor quantidade de vértices que represente fielmente o objeto (Figura 4).

Além da boa representação do modelo, outro componente importante em realidade virtual é o encaixe do modelo representado em uma cena dotada de realismo. Neste sentido, foram integradas representações visualmente dinâmicas de céu e de mar [Araujo and Celes 2002]. Com isso evita-se uma simples representação do modelo no espaço vazio.

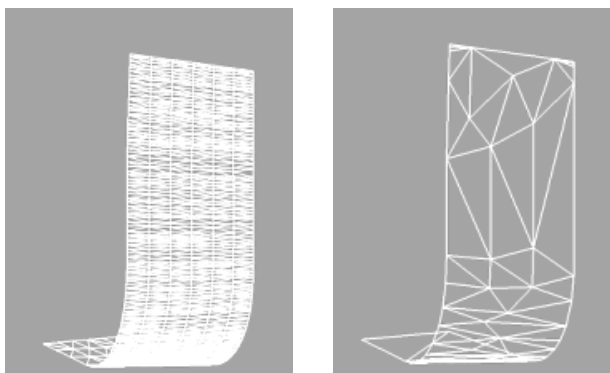


Figura 4. Malha otimizada: de 1315 polígonos para 59 polígonos

Outro componente que aumentou o realismo e a eficiência do visualizador foi o uso de ferramentas como o VRJuggler[Bierbaum 2000] e o OpenSceneGraph [Osfield and Burns 2003], descritas a seguir.

O VRJuggler é um conjunto de ferramentas desenvolvidas para abstrair o programador de tarefas comuns em RV, com especial atenção para abstrações de display e controle. A aplicação que usa o VRJuggler não precisa ser desenvolvida levando em consideração o tipo ou o número de projetores ou telas usadas para a exibição, nem da disposição delas. Do mesmo modo, a aplicação não precisa ser previamente preparada para ser controlada por um determinado dispositivo. Se em algum momento houver a necessidade de se trocar um simples Joystick por um sistema de sensores de posição, tudo o que precisa ser feito é re-configurar o VRJuggler para aceitar esse novo dispositivo.

Quando os primeiros testes de exibição da plataforma começaram a ser feitos com o VRJuggler sobre o OpenGL, surgiu a necessidade do uso de uma estrutura de cena mais robusta e eficiente que ajudasse na organização do que estava sendo desenvolvido e melhorasse o desempenho da aplicação. Decidiu-se utilizar o OpenSceneGraph, que oferece um pacote completo com grande parte do que se pode esperar de um grafo de cena. Ele traz implementadas várias otimizações importantes para um programa que trabalhe com modelos grandes, dentre elas o descarte de objetos fora do cone de visão (frustum culling) e técnicas de redução de trocas de estados do OpenGL (lazy-state-change).

O OpenSceneGraph ainda tem a vantagem de ter todo o seu código aberto. Em diversas experiências dos autores, o OpenSceneGraph tem se mostrado extremamente expansível. Modificações como o suporte a objetos paramétricos foram feitos sem que fosse alterada a estrutura do grafo de cena.

3.4. Avango/Performer

Outra opção que está sendo estudada é a utilização das ferramentas OpenGL/Iris Performer [Rohlf and Helman 1994] e Avango [Tramberend 1999]. A combinação Avango/Performer é uma alternativa semelhante à combinação VRJuggler/OpenSceneGraph. O Avango é um framework para a criação de aplicações distribuídas interativas em Ambientes Virtuais (Virtual Environments). A distribuição dos dados é obtida através da replicação transparente do grafo de cena entre os processos participantes da aplicação distribuída, criando o conceito de grafo de cena compartilhado (distributed shared scene graph).

No momento estão sendo iniciados testes com o Avango/Performer sobre plataformas SGI e também clusters de PCs. Os primeiros resultados estão expostos na seção 5.

4. Soluções Existentes

4.1. Soluções Acadêmicas

Entre as soluções acadêmicas propostas para a visualização em tempo real de grande volume de dados, uma que merece destaque pela grandeza do projeto é o GigaWalk [Baxter et al. 2002]. Esse sistema usa técnicas de reorganização do grafo de cena, HLOD (*hierarchical level of detail*) e descarte por oclusão para conseguir boas taxas de visualização em modelos CAD com dezenas de milhões de polígonos.

Na fase de pré-processamento, o sistema possui um algoritmo para reorganizar o grafo de cena do modelo CAD, através de uma combinação de particionamento e reagrupamento de objetos, com o objetivo de gerar objetos de tamanhos e localizações uniformes. Ainda na fase de pré-processamento também são gerados os níveis de detalhe para cada nó do novo grafo de cena. Por causa de toda esta reorganização que se mostra necessária, o sistema possui uma forte parte de pré-processamento do modelo para visualização. No entanto, isto gera ressalvas de parte das pessoas ligadas às soluções comerciais pois demanda muito tempo (dezenas de horas) para tratar um modelo com tamanho e complexidade equivalentes aos da P43.

Testes em uma máquina SGI com 2 pipelines gráficos, 3 CPUs 300 MHz R12000, 16GB de memória principal e placas gráficas Infinite Reality mostraram taxas variando de 11 a 50 quadros por segundo para um modelo com cerca de 82 milhões de polígonos [Baxter et al. 2002].

Outro sistema com objetivos similares é o REVIEW (Real-time Virtual Environment Walkthrough) [Shou et al. 2001], que combina técnicas “convencionais” de navegação – por exemplo, o descarte de objetos fora do campo de visão – com técnicas de otimização de I/O para grandes bases de dados.

4.2. Soluções Comerciais

Soluções comerciais também foram testadas para avaliação e comparação de performance e confirmação dos principais gargalos no processo. Dos produtos existentes na área de RV para o modelo em estudo, houve a oportunidade de avaliar mais intensamente dois deles: o Division Reality [PTC 2003a] e o WalkInside [VRContext 2003]. Embora apresentem enfoques diferentes, ambos se propõem a encarar o mesmo desafio: gerar o modelo RV a partir do modelo CAD (Figura 5).

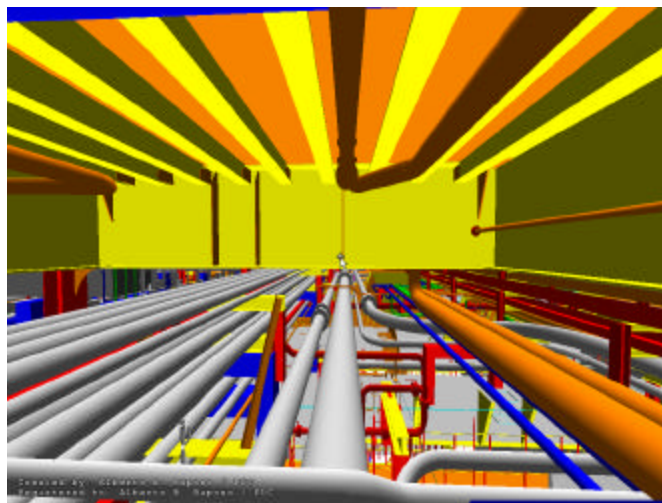


Figura 5. Imagem interna do modelo da P43 no WalkInside

Ambas as soluções apresentaram muitas dificuldades no processo de conversão, em especial o Division Reality. Porém, mesmo o WalkInside, que apresenta uma facilidade maior devido a sua integração com o formato original, ainda possui limitações para a visualização de modelos do porte da P43. Seu processo de conversão é feito diretamente do MicroStation [Bentley 2003b] – ferramenta de CAD cujo formato (DGN) é o gerado pelo PDS.

Nos dois casos foram necessárias seguidas intervenções do corpo técnico das empresas responsáveis para que a conversão fosse realizada. Ficou claro que para modelos com esta complexidade o processo de conversão está longe de ser automático. Demanda uma intervenção forte no modelo CAD, com otimizações sendo feitas “manualmente” para que o modelo fique viável para o sistema.

Avaliando a visualização no Division Reality e no WalkInside, e comparando as respectivas performances, do ponto de vista técnico, sem considerar o custo econômico, pôde-se notar que:

- O WalkInside possui um rendimento melhor nas taxas de frames por segundo (fps), sendo visualmente explícita a simplificação que ele faz na representação do objeto de acordo com a distância do observador. Ele também possui interface bastante simplificada, baseada em interface de jogos, com poucos comandos e uso intuitivo das teclas e dispositivos de interação.
- Uma grande vantagem do WalkInside é o fato de ele apresentar modos com simulação de gravidade e detecção de colisão, que juntamente com outras funcionalidades disponíveis – medição de distâncias, marcador de tempo, configuração do avatar para tamanho e velocidade de caminhada ou corrida – permitem fazer estudos de ergonomia, e definição de rotas de fuga.
- O Division Reality apresenta vantagens na manipulação direta dos objetos – fundamental para as fases de manutenção e treinamento – permitindo simulações e estudo de viabilidade de operações como troca de equipamento.
- Ambos permitem o trabalho colaborativo. No Division Reality usuários remotos, além de visualizar os demais usuários, podem compartilhar a mesma visão e realizar alterações no modelo. No WalkInside a colaboração se limita a colocar vários usuários no mesmo modelo, sem compartilhamento de visão.
- O WalkInside está limitado para uso em desktops e não serve para ambientes altamente imersivos (por exemplo, CAVEs), enquanto o Division Reality é apropriado para este tipo de ambiente e co-existe com uma outra aplicação derivada para rodar em desktops, denominada Product View [PTC 2003b].

Existem outras soluções comerciais com finalidades semelhantes, como o IC:IDO [ICIDO 2003] que não chegaram a ser avaliadas.

5. Testes de performance

Alguns testes foram registrados para uma comparação de performance, de acordo com variações no hardware e também no modelo convertido. Não representam uma avaliação completa em relação às possibilidades apresentadas neste trabalho, mas confirmam algumas das dificuldades e melhorias aqui citadas para se obter um modelo RV com performance satisfatória. A seguir os números que ilustram os testes (os fps obtidos são um valor médio em uma região específica da plataforma):

- a) Para um modelo simplificado da plataforma (aproximadamente 300.000 polígonos), e utilizando o visualizador apresentado neste trabalho (com VRJuggler, OpenSceneGraph e soluções próprias de otimização), foram registrados testes com as seguintes configurações:

- Pentium IV, 2.5 GHz, 512Mb RAM, placa Matrox Parhelia (128 Mb) – 65 fps (mono) e 32 fps (stereo)
 - Pentium IV, 2.5 GHz, 3Gb RAM, placa de vídeo Quadro FX1000 (128 Mb) – 85 fps (mono) e 44 fps (stereo)
 - SGI Onyx 2, 300 MHz (dual), 2 Gb RAM, 2 pipes (Infinity Reality IV) – 5 fps (mono)
- b) Sobre o mesmo modelo (300.000 polígonos), e utilizando o Avango/Performer, também foram registrados testes com duas configurações diferentes. Cabe ressaltar que esses são valores preliminares, visto que as soluções apresentadas ainda não foram implementadas no *Performer*.
- Cluster GenLock/SwapLock (1 BootManager, 1 Master e 4 Slaves) de Pentium IV xeon, 2GHz, 2Gb RAM, placa de vídeo ATI FireGL 4 FrameLock (128Mb) - 10 fps (stereo)
 - SGI Onix IR2 R1200, 9Gb RAM, 4 pipes, 4RM – 3 fps (stereo)
- c) Variando-se a complexidade do modelo, retorna-se ao visualizador apresentado neste trabalho com a configuração de um Pentium IV, 2.5 GHz, 3Gb RAM, placa de vídeo Quadro FX1000 (128 Mb):
- 300.000 polígonos – 85 fps (mono)
 - 600.000 polígonos – 42 fps (mono)
 - 900.000 polígonos – 28 fps (mono)
 - 1.200.000 polígonos – 24 fps (mono)
- d) Novamente variando-se a complexidade do modelo, desta vez utilizando uma das soluções comerciais existentes – o WalkInside – foram registrados testes com partes do modelo da plataforma. A configuração utilizada foi um Pentium IV, 2.5 GHz, 3Gb RAM, placa de vídeo Quadro FX1000 (128 Mb).
- 25.000 polígonos – 100 fps (mono)
 - 270.000 polígonos – 170 fps (mono)
 - 1.100.000 polígonos – 18 fps (mono)
 - 1.125.000 polígonos – 14 fps (mono)
 - 1.200.000 polígonos – 8 fps (mono)

Neste último caso, pode-se constatar que o rendimento não depende apenas do número de polígonos gerados para a visualização, pois este número depende também da complexidade de cada elemento representado (repare que o segundo resultado possui fps melhor que o primeiro, que tem muito menos polígonos). Por exemplo, trechos definidos por superfícies do tipo NURBS geram, no WalkInside, muito mais polígonos do que trechos definidos por objetos com geometria definida por poucos dados paramétricos. Desta forma não se pode afirmar que a performance do visualizador desenvolvido seja melhor do que a do Walkinside, pois os números apresentados variam de acordo com a técnica de definição de polígonos de cada visualizador. Por exemplo, em um outro modelo ou mesmo em outras regiões da plataforma o desempenho do WalkInside pode ser melhor.

6. Conclusões e próximos passos

Os problemas apresentados no caminho do CAD para RV e o fato de que ainda não foram equacionados na sua totalidade por nenhuma solução comercial ou acadêmica, limitam mas não impedem o uso destes sistemas nas grandes empresas. O processo ainda é feito por partes, sendo algumas destas partes intervenções não automatizadas e particularizadas para o modelo a ser trabalhado. Mesmo assim, já existem ganhos com processos de simulação de construção e manutenção

de equipamentos. Além disto, comercialmente existe um ganho nestas empresas que usam tecnologias de realidade virtual sobre o modelo real de engenharia, através da disponibilidade de uma maquete eletrônica dotada de alto grau de realismo e imersividade, fazendo com que o cliente tenha uma noção melhor do produto, inclusive interagindo com ele.

Os problemas apresentados serão minimizados quando existirem padrões estabelecidos na área de transição do CAD para RV. Por enquanto o esforço apresentado neste trabalho, bem como outras pesquisas acadêmicas que atuam paralelamente, são válidos na busca destas soluções. Com relação ao trabalho aqui apresentado, alguns passos já estão definidos para o seu prosseguimento, a saber:

- Meios para aplicação de textura ao modelo vindo do CAD de uma forma automatizada. É um trabalho complementar à integração de cenários como céu e mar, para se criar um ambiente de realidade virtual dotado de grande realismo;
- Busca de uma configuração de hardware que resulte em melhor performance. Devem ser feitos testes mais detalhados comparando-se as diferentes soluções disponíveis para hardware – desktops com novas placas de visualização, utilização de clusters, plataformas SGI e outros. Cabe ressaltar que este deve ser um trabalho constante devido a velocidade do desenvolvimento dos equipamentos nos dias de hoje.
- Definição de um modelo semântico dos objetos provenientes do CAD para padronizar a leitura e organização para efeito de visualização. Devido a limitações na interpretação do formato DGN, os dados exportados para o visualizador são passados em representações diferentes – objetos parametrizados, formas geométricas e malhas de visualização. Isto dificulta o processo na hora de se montar a cena para visualização.
- Novas otimizações no visualizador, como por exemplo, um processo de culling hierárquico – criação de uma hierarquia entre objetos para permitir que haja descarte de objetos por grupos, sem ter que percorrer todos os objetos individualmente.
- Aspectos relacionados à interação com o modelo e o uso de técnicas e ferramentas de colaboração.

Além destes passos já detalhados, um importante salto de qualidade que deve ser buscado em um futuro próximo pelos autores é a integração com outros sistemas ou ambientes. Trata-se de pesquisar a possibilidade de trazer para o modelo RV outras classes de dados que permitiriam a montagem de um ambiente mais complexo. Um exemplo seria encaixar, dentro de um ambiente de realidade virtual integrado, o modelo que vem do CAD sobre outro modelo vindo de um sistema GIS que representaria a superfície de uma região, podendo-se também anexar a este novo conjunto dados sísmicos. No caso específico dos dados utilizados neste trabalho, seria ter um ambiente integrado, com o modelo da P43 colocado na bacia de exploração e tendo representações do reservatório subterrâneo, do terreno do fundo do mar, de equipamentos submersos e também a da superfície do mar e a área externa. Este ambiente integraria diversas funções e análises da área de exploração e produção de petróleo, reforçando o grande potencial da realidade virtual como ferramenta operacional na indústria.

Agradecimentos

A pesquisa em Realidade Virtual do Tecgraf/PUC-Rio é apoiada primordialmente pelo projeto AmbVirt patrocinado pela FINEP/PETROBRAS. Especiais agradecimentos às pessoas da Petrobras envolvidas neste projeto: Álvaro Maia, Heitor Araújo Filho e Ênio Russo (CENPES), Paulo Roberto Araújo de Oliveira, Daniel Mandarino e João Carlos Pereira (UN-Rio). Também agradecemos a todos do TeCGraf que colaboraram: todo o grupo de visualização, e colegas de linhas de pesquisa afins, em especial William Wagner e Vinícius Almendra. Alberto Raposo é bolsista de produtividade em

pesquisa do CNPq, processo nº 305015/02-8. Ismael Santos está atualmente como pesquisador visitante no IMK/FhG, Bonn, Alemanha.

Referências

- Araujo, M.P. and Celes, W. (2002) “Uma Estratégia para a Visualização do Céu e seus Elementos em Jogos”, I Brazilian Workshop in Games and Digital Entertainment.
- Baxter III, W.V., Sud, A., Govindaraju, N.K. and Manocha, D. (2002) “GigaWalk: Interactive WalkThrough of Complex Environments”. Eurographics Workshop on Rendering.
- Bentley Systems Incorporated (1995) “Intergraph Standard File Formats”. MicroStation 95 Ref. Guide chapter 18.
- Bentley Systems Incorporated (1999a) “MDL Function Reference Manual”.
- Bentley Systems Incorporated (1999b) “MicroStation MDL - Programmers Reference Guide”.
- Bentley Systems Incorporated (1999c) “MDL Solids Modeling Reference”.
- Bentley Systems Incorporated (2003a) “MicroStation BASIC Technical Support Documentation”, <http://selectservices.bentley.com/technotes/product9.htm>
- Bentley Systems Incorporated (2003b), “MicroStation”, <http://www.bentley.com>
- Bierbaum, A. (2000) “VR Juggler: A Virtual Platform for Virtual Reality Application Development”, Master Thesis, Iowa State University.
- Electronic Data Systems – EDS (2003), “ParaSolid”, <http://www.eds.com/products/plm/parasolid/index.shtml>
- IC:IDO (2003) “ICIDO VR”, <http://www.icido.de>
- Intergraph (2003) “Plant Design System –PDS”, <http://ppo.intergraph.com/pds>
- Lavoie P. (1999) “NURBS++: The Nurbs Package - User’s Reference Manual - Version 3.0”, <http://libnurbs.sourceforge.net>, Ottawa.
- Lira, W.W.M., Cavalcanti, P.R., Coelho L.C.G. and Martha, L.F. (2002) “A Modeling Methodology for Finite Element Mesh Generation of Multi-Region Models with Parametric Surfaces”, Computers & Graphics, v. 16, n. 6, p. 907-918.
- Osfield, R. and Burns, D. (2003) “Open Scene Graph”, <http://www.openscenegraph.org>
- Parametric Technology Corporation – PTC (2003a), “Division Reality”, <http://www.ptc.com>
- Parametric Technology Corporation – PTC (2003b) “Division Product View”, <http://www.ptc.com>
- Piegl L. and Tiller W. (1999) “The Nurbs Book”, 2nd ed. Ottawa: Springer.
- Rohlf, J. and Helman, J. (1994) “IRIS Performer: A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics”, Proc of the SIGGRAPH’94.
- Shou, L., Chionh, J. et al. (2001) “Walking Through a Very Large Virtual Environment In Real-time”. Proc. 27th Int. Conf. on Very Large Data Bases, pp. 401-410.
- Singhal, S. and Zyda, M. (1999) “Networked Virtual Environments”, Addison Wesley, USA.
- TeCGraf - Grupo de Tecnologia em Computação Gráfica (1996) “MG: Manual do Usuário - Versão 3.0”, Pontifícia Universidade Católica do Rio de Janeiro, <http://www.tecgraf.puc-rio.br/~lula/mg>
- Tramberend, H. (1999), “Avocado: A Distributed Virtual Reality Framework”, Proc of the IEEE Virtual Reality, <http://www.avango.org>
- VRContext (2003), “WalkInside”, <http://www.walkinside.com>