

Design de uma Interface de Interação Tridimensional com Foco na Usabilidade e no Desempenho Gráfico

S.L. Dos Santos^{a,b} e F.G. Teixeira^b

^a*sergio.santos@ufrgs.br*

^b*Programa de Pós-Graduação em Design, Departamento de Design e Exp. Gráfica,
Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil*

Resumo

Este trabalho mostra o desenvolvimento de uma interface 3D para manipulação direta de objetos, visando à melhoria da interatividade, desempenho e usabilidade de um programa para projeto de produtos. São abordados os melhores métodos de manipulação direta de objetos em tempo real e de controle de câmera para programas 3D. O protótipo da nova interface mostrou bons resultados com relação ao desempenho gráfico quando comparado com a interface original e com relação a facilidade e qualidade da interação tridimensional

Palavras-chave: Usabilidade, Manipulação Direta, 3D.

Design de uma Interface de Interação Tridimensional com Foco na Usabilidade e no Desempenho Gráfico

Abstract

This work shows the development of a 3D user interface for direct manipulation of objects, aiming the improvement of interactivity, performance and usability of a software tool for product development. The different methods for direct manipulation of objects in real time and camera movement controls of the 3D software are studied. The new interface prototype has shown good results on the graphics performance when compared with the original interface, and regarding to simplicity and quality of tridimensional interaction as well.

Keywords: Usability, Direct Manipulation, 3D.

1. INTRODUÇÃO

Sistemas CAD/CAE (*Computer Aided Design / Computer-aided engineering*) tornaram-se uma tecnologia especialmente importante na área de projetos computacionais utilizados pelo design, engenharia e arquitetura, com benefícios como redução dos custos de desenvolvimento de um produto e grande redução do tempo no ciclo de projeto.

Segundo Teixeira *et al.* [12], as ferramentas CAD que oferecem recursos tridimensionais (3D) permitiram substituir os ensaios físicos por simulações virtuais, transformando o processo de projeto. Conforme Teixeira *et al.* [13], a possibilidade de realizar todas as etapas de um projeto, de uma forma totalmente virtual traz diversas vantagens, como a redução de custos e do tempo de desenvolvimento e, principalmente, a otimização do projeto em si.

Este trabalho divulga os resultados obtidos no trabalho de mestrado do autor [1] que tem como objetivo o desenvolvimento uma interface 3D para softwares de projeto considerando principalmente requisitos como Interatividade, Desempenho e Usabilidade. As técnicas e ferramentas apresentadas aqui foram desenvolvidas com este objetivo. Para a criação da interface foi utilizado, como estudo de caso, a plataforma de desenvolvimento T-CADE. Segundo Teixeira [11], o software T-CADE foi originalmente criado para geração interativa de malhas otimizadas para análise de elementos finitos. Na medida em que foi sendo desenvolvido, foram

acrescentadas novas funcionalidades como a geração de superfícies complexas e o cálculo de intersecções. O T-CADE foi implementado com programação orientada a objetos na linguagem Delphi utilizando algoritmos gráficos próprios, que resolveram muito bem os problemas de interface até então apresentados, mas têm limitado o desenvolvimento e as possibilidades de criação de novas ferramentas em função da complexidade das operações gráficas exigidas. A Figura 1 mostra a interface original do programa T-CADE com o modo de visualização "Aramado+Shade", evidencia-se neste caso a área de trabalho de interface (a parte com fundo branco onde o objeto é mostrado) que é o objeto de estudo deste trabalho.

A interface original não permite a seleção de objetos clicando sobre a superfície do objeto. A seleção só acontece quando as linhas de construção deste objeto são selecionadas. O ideal quanto à seleção seria selecionar o objeto através de sua superfície e, ainda, reconhecer as coordenadas paramétricas da parte do objeto em que o cursor está apontando (coordenadas em 3D do ponto na superfície). Com a melhoria da seleção dos objetos, novas ferramentas de edição e manipulação de objetos poderiam ser facilmente implementadas. Além da seleção unitária (um objeto de cada vez), é necessário também que se possa selecionar um grupo de objetos de uma única vez, utilizando uma janela de seleção.

A manipulação direta dos objetos também precisa ser melhorada. Uma vez selecionados deve ser possível mover e/ou copiar os objetos de forma fácil e direta visando agilizar o processo de modelagem tridimensional.

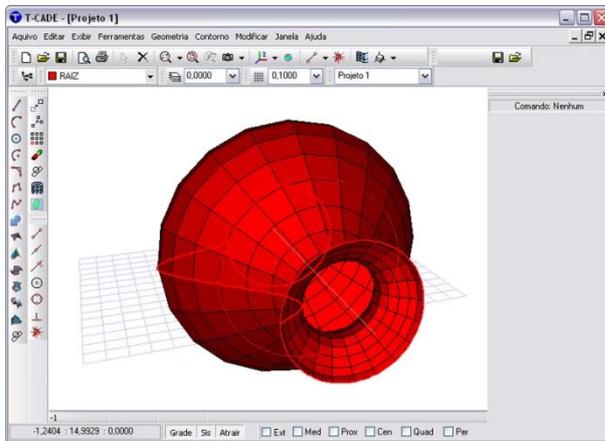


Figura 1: Superfície de Revolução no T-CADE.

O desempenho gráfico é outro limitante. Nos casos em que existem muitos polígonos representados na tela, o nível de interatividade com o usuário cai drasticamente quando o ponto de vista da cena é movimentado, dando a impressão que a tela está piscando podendo parecer que o programa não está respondendo por um curto período de tempo. Os modos de visualização da interface original também são limitados, não suportando as opções de transparência e suavização da superfície, entre outras, o que pode melhorar a representação da superfície e facilitar o entendimento da cena tridimensional.

Além disto, a interface possui poucos controles para a movimentação da câmera tornando a definição da posição de visualização da cena menos interativa.

Estas limitações podem ser agrupadas em termos de Interatividade, Desempenho e Usabilidade. A interatividade é um requisito importante para um sistema 3D eficiente. As modificações no modelo e no ambiente durante o processo de design devem ser visualizadas imediatamente, em tempo real. Para um desempenho gráfico aceitável para aplicações 3D em tempo real, deve-se tentar manter, principalmente durante as interações, uma taxa de atualização de tela igual ou superior a 30 quadros por segundo (padrão de cinema). Usabilidade é um atributo de qualidade que avalia a facilidade de uso das interfaces de usuários.

O principal objetivo deste projeto foi desenvolver uma interface 3D que resolvesse as questões de Interatividade, Desempenho e Usabilidade deste artefato digital possibilitando o desenvolvimento de novas funcionalidades nesta plataforma, ampliando significativamente seu potencial como ferramenta de projeto.

2. INTERAÇÃO HOMEM-COMPUTADOR

Em linhas gerais, a área de interação homem-computador investiga o design, avaliação e implementação de sistemas computacionais interativos para uso humano, juntamente com os fenômenos associados a este uso [3]. Interação é o processo de comunicação entre pessoas e sistemas interativos [9]. O estudo de sistemas interativos tem como foco o usuário e sua experiência no uso do artefato digital. A interface deve tornar esta experiência o mais fácil, prática e agradável possível.

De acordo com Nielsen [6], a usabilidade é a característi-

ca que determina se o manuseio de um produto é fácil e rapidamente aprendido, dificilmente esquecido, não provoca erros operacionais, oferece um alto grau de satisfação para seus usuários e, eficientemente, resolve as tarefas para as quais ele foi projetado. Existem muitos motivos porque investir ou se preocupar com a usabilidade de uma interface. A usabilidade está diretamente ligada à produtividade, em função disto tem se tornado cada vez maior o interesse das empresas em desenvolverem melhores interfaces, melhorar a usabilidade também significa diminuir custos de treinamento, aumentar a satisfação dos usuários e, por consequência, aumentar as vendas do software.

De acordo com Nielsen [6], para que o sistema tenha boa usabilidade, é necessário atender aos seguintes requisitos:

- Ser de fácil aprendizagem;
- Ser eficiente na utilização;
- Ser fácil de lembrar;
- Ter poucos erros;
- Satisfazer subjetivamente o usuário.

Para o autor, a preocupação com usabilidade deve estar presente durante todo o ciclo de projeto e não apenas quando o produto já está finalizado. Assim, podemos fazer as modificações necessárias sem que elas impliquem em um custo elevado de re-projeto.

Considerando os requisitos de usabilidade da interface, para facilitar a aprendizagem do usuário e facilitar a lembrança do funcionamento da interface 3D, metáforas são utilizadas, assim como é feito em outros softwares do cotidiano (por exemplo, um processador de texto utiliza a metáfora da máquina de escrever). Segundo Kettner [4], uma metáfora 3D explica a manipulação de uma operação 3D utilizando um dispositivo de entrada 2D. A eficácia de sistemas 3D depende das ferramentas e técnicas disponíveis ao usuário para uma rápida e amigável definição e modificação do modelo tridimensional.

3. ANÁLISE DE SIMILARES

Para a implementação de uma nova interface em um programa 3D, foram investigados os softwares 3D/CAD/CAE mais amplamente utilizados no mercado atualmente: Rhinoceros, AutoCAD e Inventor. Estes programas foram estudados, principalmente, sob o ponto de vista das ferramentas utilizadas para manipulação direta dos objetos, das ferramentas de visualização e movimentação da câmera no mundo tridimensional, e dos métodos e técnicas de trabalho voltadas à produtividade e usabilidade.

Na avaliação comparativa entre eles foram encontradas algumas semelhanças importantes:

- O uso de manipuladores para restrições de eixos para a manipulação direta de objetos, podendo-se diretamente com o uso do cursor em um único movimento, manipular os objetos com as restrições desejadas.
- O uso de teclas de atalho para facilitar e acelerar o uso das ferramentas pelos usuários mais experientes.
- O uso de vistas pré-definidas mais freqüentemente utilizadas ou mais representativas para a visualização em 3D.
- A possibilidade de se trabalhar em planos auxiliares de trabalho, como sistemas de coordenadas criados pelo usuário com uma posição no espaço distinta do sistema de coordenadas globais, bem como os sistemas locais dos objetos e das vistas (ou câmeras).

- O uso transparente dos comandos de visualização, ou seja, deve ser possível usar funções da câmera como zoom, pan, ou girar a câmera ao redor do alvo enquanto executa um determinado comando de edição, facilitando a visualização completa necessária para melhor entendimento e execução precisa da tarefa.
- Além da seleção direta de objetos com um clique sobre o objeto, apresentam a possibilidade de selecionar os objetos através de janelas de seleção, com dois métodos distintos, selecionando apenas objetos inteiramente contidos na Janela de seleção ou incluir também os objetos parcialmente contidos na janela.

Além destas observações, algumas características foram apontadas como importantes para um software de modelagem tridimensional:

- A importância de que quando um objeto é selecionado, seja possível identificar sobre a superfície deste objeto o ponto 3D apontado pelo cursor; o reconhecimento do ponto 3D, sobre a superfície que é clicada ao se selecionar um objeto, é importante para o desenvolvimento de alguns algoritmos, como o de concordância de superfícies, que atualmente em termos de estrutura de dados já poderia ser implementado, mas necessita deste reconhecimento do ponto 3D clicado na superfície para funcionar conforme o esperado.
- Deve ter um bom desempenho gráfico quando trabalhar com vários objetos. O desempenho gráfico é importante em termos de usabilidade e da garantia da interatividade com o usuário. O desempenho gráfico não pode ser limitador das ações do usuário. Uma nova interface gráfica só se justifica se ela apresentar um desempenho gráfico melhor que a interface original.
- A interatividade é fundamental na criação dos objetos, uma retroalimentação (feedback) mais natural e precisa pode ser fornecida enquanto o usuário está interagindo.
- A nova interface deve ser mais fácil de usar; deveria oferecer uma melhor usabilidade em relação à interface original.

A partir destes conceitos, foi dado início a fase de projeto da interface. Nesta fase foram utilizadas ferramentas e técnicas da área de desenvolvimento de produto que selecionaram as alternativas de ferramentas para implementação no programa.

4. IMPLEMENTAÇÃO

A variedade de hardware para computação gráfica, como placas gráficas 3D, canetas digitais e monitores, induzem ao uso de bibliotecas gráficas para utilizá-los de maneira transparente e eficiente. O GLScene [2] é uma Biblioteca gráfica baseada em OpenGL [8] e otimizada para a linguagem Delphi que fornece componentes e objetos visuais permitindo a descrição e renderização de cenas 3D de uma maneira visual e interativa, possibilitando a determinação das propriedades dos componentes em tempo de projeto e em tempo de execução.

A nova Interface foi criada para substituir a interface original trazendo melhorias na Interatividade, Desempenho e Usabilidade. Foi criada uma nova estrutura de classes de representação dos objetos 3D utilizando a biblioteca gráfica GLScene buscando a melhoria do desempenho gráfico. As novas classes gráficas são propriedades dos objetos existentes na estrutura do T-CADE. Objetos lineares (uma dimensão)

têm um tipo de representação diferente de objetos planos (duas dimensões) e objetos tridimensionais, portanto, estas classes gráficas devem atender às diferentes necessidades de representação dos objetos existentes e dos objetos ainda a serem criados.

A implementação da nova interface foi direcionada para duas questões específicas: a melhoria do desempenho gráfico e a melhoria da interatividade. A partir destes pontos buscou-se, então, uma melhor usabilidade do software.

4.1 Interatividade

No desenho de objetos, a interatividade é fundamental para o controle das ações do usuário. Uma vez implementada a nova biblioteca gráfica, foi possível também a criação de novas ferramentas gráficas que permitem uma melhor interatividade com os objetos 3D e, conseqüentemente, melhor usabilidade. Foram implementadas técnicas de manipulação direta dos objetos selecionados, permitindo a restrição da movimentação e da cópia do objeto selecionado aos eixos cartesianos, facilitando assim sua manipulação em 3D.

4.1.1 Seleção de objetos

Na OpenGL, a seleção de objetos se dá pela intersecção de vetores no espaço tridimensional. A Biblioteca GLScene já traz algoritmos de seleção de objetos baseados neste princípio. A seleção de objetos pode ser feita de três maneiras: clicando diretamente sobre o objeto, através de uma janela Cheia (*Window selection*) e através de uma Janela tracejada (*Crossing selection*). Ao se clicar sobre um objeto, este fica selecionado e sua aparência se modifica, suas linhas e suas superfícies, se houver, ficam imediatamente alaranjadas, indicando a seleção. A figura 2 mostra os dois tipos de janelas de seleção: janela cheia e janela tracejada.

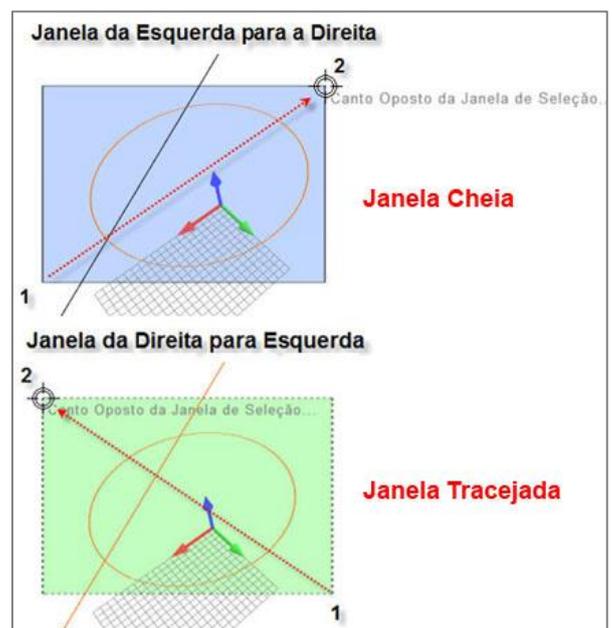


Figura 2: Janelas de seleção.

O modo de seleção de janela cheia é feito da esquerda para a direita e uma janela (quadro) azul transparente com borda de linha contínua é desenhada na tela. O primeiro clique inicia a janela e um segundo clique no canto oposto da janela termina a seleção. Neste modo, somente os objetos totalmente contidos dentro da janela são selecionados. No modo de seleção de Janela tracejada, a criação da janela é feita da direita para esquerda e uma janela verde transparente

te com borda pontilhada é desenhada na tela. Neste modo, qualquer objeto tocado pela janela é selecionado.

Estas opções de seleção, além de tornarem o processo mais ágil, aumentam o controle do usuário na interação com o programa, principalmente quando se trabalha em cenas complexas com dezenas de objetos.

4.1.2 Localização do ponto 3D

Os algoritmos de intersecção da biblioteca GLScene (RayCast) têm, entre outras características interessantes, a capacidade de saber a posição 3D do ponto clicado sobre uma superfície, o que é fundamental para a implementação de novos algoritmos com uma maior interatividade. Esta era, até então, uma limitação do T-CADE. Infelizmente esta capacidade se aplica somente a superfícies e não às linhas (uma curva qualquer).

Na interface original do T-CADE, na seleção de retas e curvas já são identificadas as posições paramétricas do ponto clicado sobre o objeto e, também, suas coordenadas globais. Usando então uma solução semelhante na nova interface, uma reta é colocada no espaço da câmera na posição do cursor e normal à tela de projeção. A partir desta reta, pode-se calcular a intersecção com qualquer outra curva no espaço 3D e saber exatamente o ponto clicado sobre o objeto. Este é um recurso importante em comandos como dividir uma linha em um determinado ponto ou, mesmo, desenhar uma nova linha a partir de um ponto qualquer pertencente à linha existente.

Para as superfícies, foi criado um algoritmo específico para a identificação do ponto 3D sobre a superfície, de tal maneira que fossem mostrados tanto o ponto 3D apontado pelo cursor quanto o vetor normal à superfície naquele ponto. Para demonstrar esta capacidade, foi criada temporariamente uma representação visual que consiste de uma esfera que é criada com centro no ponto 3D sobre a superfície e uma seta 3D indicando o sentido do vetor normal conforme mostrado na Figura 3.

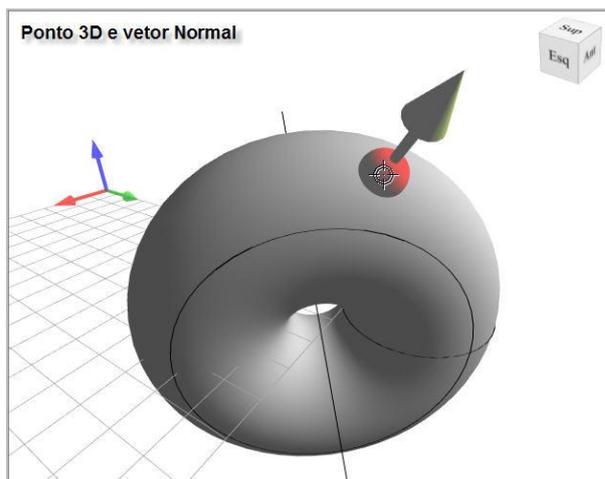


Figura 3: Esfera e Seta 3D no cursor.

4.1.3 Criação interativa de objetos

Ao desenhar uma linha, a linha deve ser mostrada durante a operação de desenho, para que usuário possa avaliar se o resultado mostrado é o resultado esperado por sua ação e então decidir aceitar ou não aquele resultado.

Um exemplo disto é a melhoria da rotina de desenho do Arco utilizando a opção “Centro-Início-Fim”. A figura 4 mostra duas situações de desenho na interface original onde os ângulos iniciais e finais são os mesmos e os resultados além

de diferentes são apenas mostrados quando o comando é finalizado.

Na interface original, somente depois de fornecidos os três pontos (início, centro e fim do arco) é que o arco era desenhado, freqüentemente causando uma surpresa ao usuário com o resultado obtido, uma vez que para os mesmos três pontos existem duas soluções possíveis (o arco de ângulo maior e o de ângulo menor).

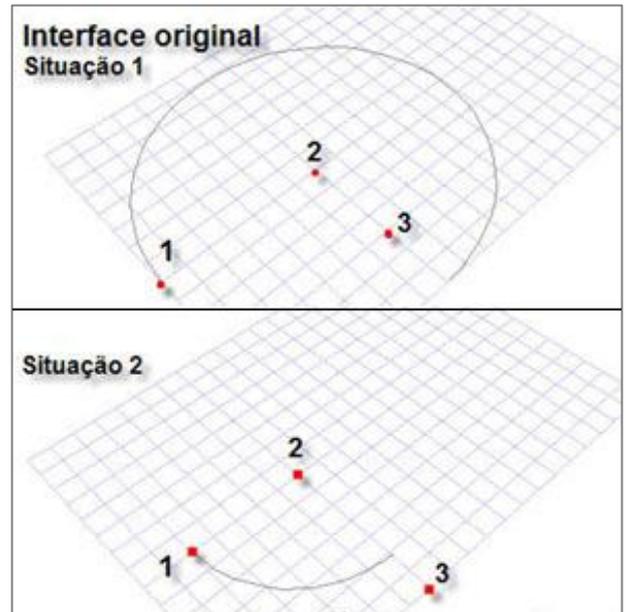


Figura 4: Arco Início, Centro e fim na Interface Original.

Na nova Interface, este problema foi resolvido através da interatividade. Para um melhor funcionamento do comando a ordem de inserção dos dados também foi alterada. Ao fornecer o primeiro ponto (centro) uma linha já fica presa ao cursor indicando a linha de início do arco (ângulo inicial), ao fornecer o segundo ponto (definindo o início do arco) o arco começa a ser desenhado acompanhando o movimento do cursor.

O sentido do arco é alterado cada vez que o arco (acompanhando o cursor) passa pelo seu início, permitindo que o usuário controle o resultado da ação. Ao fornecer o último ponto, o comando é então finalizado. A figura 5 mostra o mesmo comando na nova interface, o arco é desenhado durante o processo e acompanha o movimento do cursor.

Um dos critérios da avaliação heurística da usabilidade segundo Nielsen e Molich [7] é a visibilidade do sistema, isto significa que se deve manter o usuário informado do resultado de suas ações, dando um feedback imediato às ações do usuário, facilitando a interação e minimizando os erros.

4.1.4 Manipulação direta de objetos

Outra aplicação importante da interatividade é na modificação e manipulação direta de objetos. Atualmente o T-CADE oferece duas possibilidades de modificação de objetos, mover e copiar. Para utilizá-los, é necessário acionar o respectivo comando, selecionar os objetos, terminar a seleção, indicar o ponto de base e, finalmente, indicar o novo ponto (relativo ao primeiro) para onde o objeto vai ser movido ou copiado. Este estilo de interação é importante por uma questão de precisão e controle na modelagem de objetos, mas é um pouco frustrante quando a precisão não é tão importante quanto a agilidade do processo.

Para suprir esta necessidade de interação foi criado a possibilidade de mover e/ou copiar os objetos enquanto

selecionados, tornando a manipulação dos objetos mais ágil e direta. Por exemplo, ao clicar sobre um objeto, pode-se movê-lo mantendo o botão esquerdo do mouse pressionado e movimentando o cursor em qualquer direção. Assim, em um único movimento (clique, arrastar e soltar), o objeto é selecionado e movido para a nova posição, aumentando a produtividade do usuário.

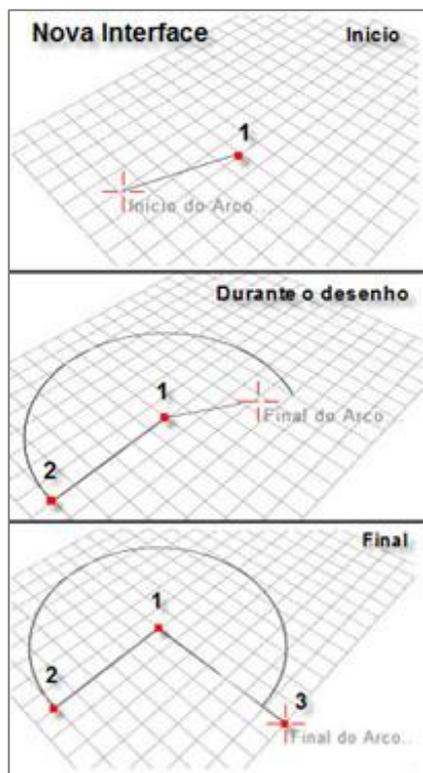


Figura 5: Arco Centro, Início e fim na Nova Interface.

O uso combinado deste movimento com teclas modificadoras para a restrição de eixos amplia seu potencial de manipulação, inclusive em termos de precisão. Pode-se mover um objeto ao longo do eixo X ou Y mantendo-se a tecla SHIFT pressionada, é desenhada uma linha de cor vermelha ou verde, indicando os eixos X e Y respectivamente, informando imediatamente ao usuário o resultado de suas ações.

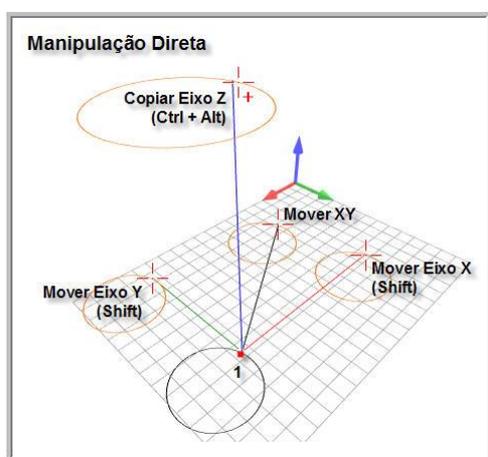


Figura 6: Manipulação direta na Nova Interface.

Pode-se ainda mover um objeto ao longo do eixo Z, utilizando para isto a combinação com a tecla CTRL, neste caso uma linha azul é desenhada indicando o movimento ao longo

do eixo Z. Para fazer uma cópia, utilizam-se os mesmos procedimentos descritos anteriormente combinados também com a tecla ALT.

Quando o modo copiar está ativado, o cursor se modifica, é desenhado um sinal de adição ao lado do cursor original. A figura 6 mostra estas diferentes situações.

Uma das ações mais intuitivas na interação tridimensional é clicar e arrastar um objeto. De acordo com Nielsen [6], a intuitividade é um dos critérios mais importantes da usabilidade. Esta técnica afeta também, além da intuitividade, a eficiência da interação tridimensional.

Assim, o uso de teclas de modificação combinados a uma interação direta com objeto selecionado, sem a necessidade de invocar comandos específicos de transformação, torna o trabalho do usuário muito mais prático e rápido, aumentando sua produtividade e, o mais importante neste caso, sua satisfação com a ferramenta digital.

4.1.5 Visualização 3D

Para o controle da visualização 3D foi criada uma versão do ViewCube [5], esta ferramenta permite a mudança de vistas pré-estabelecidas de maneira simples e intuitiva evitando a desorientação comum em ambientes abstratos 3D.

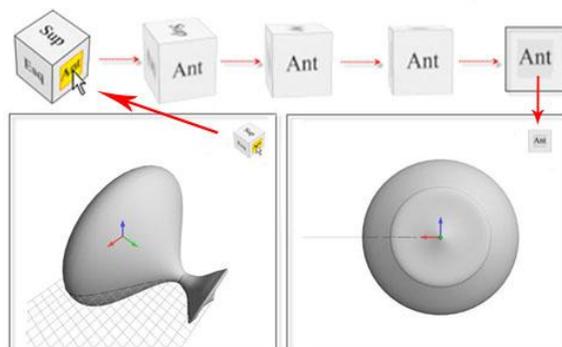


Figura 7: O Cubo de orientação: Clicando na face do cubo, tanto o cubo quanto a cena se ajustam a nova vista em 3D.

A partir de um cubo no canto superior direito da tela, com o nome das vistas nas faces, o usuário tem o controle e a indicação da orientação da cena. Quando usado como controlador da posição de visada, clicando sobre as faces, arestas ou cantos do cubo, tanto o cubo quanto a cena mudam sua orientação ajustando-se a vista correspondente. Quando a cena é modificada por outros métodos de movimentação de câmera, o cubo se ajusta para indicar a nova orientação. A transição entre as mudanças de vistas é feito de modo animado de maneira a suavizar a mudança e evitar a desorientação do usuário causada por mudanças bruscas da cena. As vistas foram chamadas de **Sup** (Superior), **Inf** (Inferior), **Ant** (Anterior), **Pos** (Posterior), **Esq** (Lateral Esquerda) e **Dir** (Lateral Direita). A figura 7 mostra o cubo e como ele funciona.

Além das vistas ortogonais nomeadas anteriormente, é possível clicar sobre as arestas do cubo para conseguir a vista intermediária entre duas vistas ou sobre um de seus vértices para obter as vistas axonométricas. A Figura 8 mostra a transição entre a posição original e nova vista escolhida em ambos os casos.

O cubo é formado por vinte e seis (26) objetos, seis (6) objetos representando as faces do cubo, oito (8) objetos representando os vértices do cubo e doze (12) objetos representando as arestas do cubo. A Cada um destes objetos esta mapeada uma determinada posição de câmera. A posição

original da câmera fica registrada em uma variável, e assim que a posição da câmera é atualizada pelo clique em um destes objetos, a interface faz a animação da posição anterior para a nova posição, minimizando a desorientação do usuário.

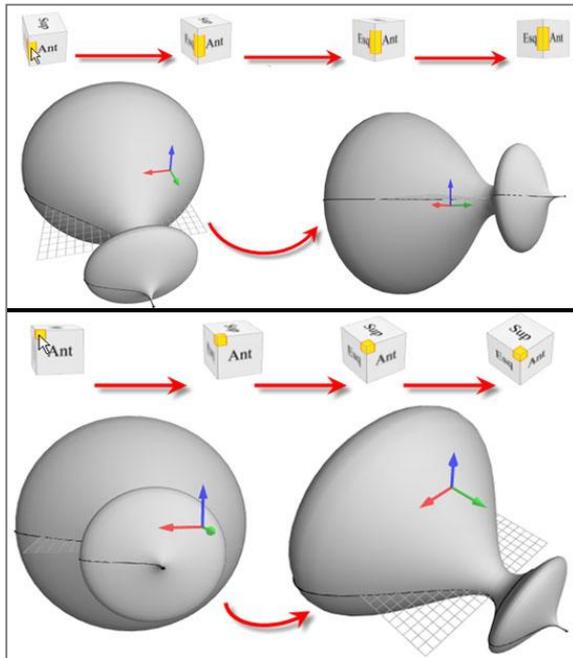


Figura 8: Vista a partir da aresta do cubo e do vértice.

Entende-se que um design interativo tende a melhorar a qualidade da experiência do usuário. Outra melhoria importante foi a criação do “QuadMenu”, uma ferramenta de controle de câmera a partir de um menu circular onde os comandos são dispostos em quadrantes, facilitando a seleção dos comandos e sua memorização. Seu funcionamento é baseado na interação gestual do mouse [14], mas com um diferencial: a seleção se dá, não através de gestos pré-definidos, mas da direção do movimento e distancia percorrida pelo cursor, inovando pela simplificação e pelo menu visual para a escolha de comandos.

Para que estas ferramentas pudessem existir, foram implementados comandos de controle de câmera como zoom interativo, zoom window, translação da câmera, rotação da câmera, entre outros. A Figura 9 mostra a ferramenta **QuadMenu** e os diferentes cursores para cada uma das ferramentas de visualização.

A nova técnica propõe que sejam concentrados os comandos comuns de visualização em um único evento do botão. Isto a facilita a memorização e agiliza a utilização da ferramenta. Além disto, a atenção do usuário permanece na área de trabalho, sem a necessidade de desviar o olhar ou procurar o botão da ferramenta correspondente àquela função.

O cursor do programa em forma de um alvo é claramente dividido em quatro quadrantes, e a zona circular definida na parte interna do cursor forma uma zona neutra onde o usuário poderá decidir que direção tomar e, portanto, qual ferramenta de visualização escolher. Uma vez que cursor seja arrastado para fora da zona neutra a ferramenta é selecionada e seus efeitos passam a serem visíveis imediatamente.



Figura 9: QuadMenu e cursores das ferramentas de câmera.

A partir de um clique com o botão direito do mouse na área de trabalho, um menu circular é mostrado com quatro opções para o controle da câmera, orbitar ao redor do alvo, mover-se sobre o plano XY, mover a altura da câmera ou mover a cena no plano da tela. Ainda com o botão pressionado o usuário leva o cursor em direção ao comando escolhido, este fica selecionado imediatamente e se o movimento continuar até que o cursor alcance o ícone do quadrante escolhido, a ferramenta aceita o comando e inicia a movimentação.

O menu de escolha só aparece depois de ¼ de segundo (250 milissegundos) que o botão seja mantido pressionado, liberando assim a função do simples clique do botão da direita para um futuro menu de contexto. Para os usuários experientes que já conhecem a posição correta (direção) das ferramentas, é possível simplesmente clicar e arrastar diretamente na determinada direção para selecionar a ferramenta, sem a necessidade de esperar que o menu apareça. Depois de acionado o comando o ícone do quadrante se transforma no cursor indicando o início da ação. Neste caso, da ação já ser escolhida diretamente, o menu nem sequer é mostrado.

A figura 10 ilustra as etapas do funcionamento deste menu de seleção dos movimentos da câmera.

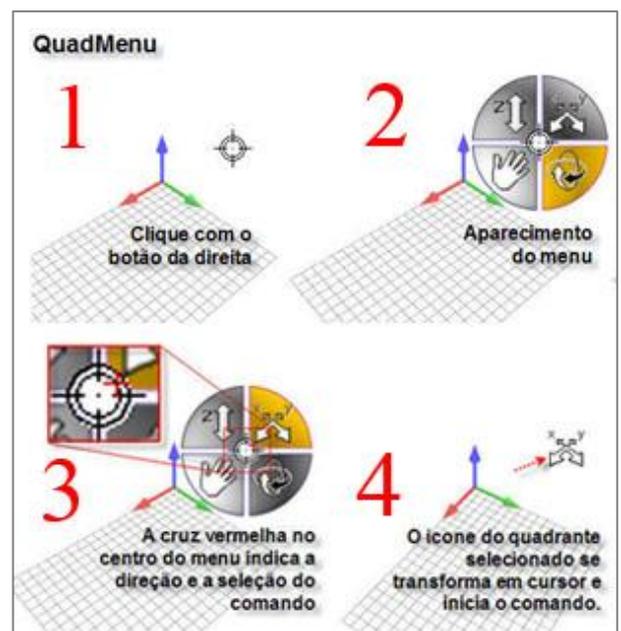


Figura 10: O QuadMenu, o comando de movimentação de camera é acionado em um único movimento.

4.1.6 Modos de visualização

A nova biblioteca gráfica permitiu que novos modos de visualização fossem implementados, aumentando o controle das opções de visualização do usuário, melhorando a clareza e o entendimento da cena tridimensional.

É possível mostrar os objetos sem a malha de visualização, em modo aramado (*WireFrame*), em modo aramado com Volume (*VolumeWire*), em modo aramado com linhas ocultas (*Hide*), em modo Volumétrico Facetado com linhas e sem linhas (*ShadedLines* e *FlatShade*), em modo Volumétrico Suavizado (*SmoothShade*) e em modo Transparente (*Ghost*).

Os diferentes modos de visualização são importantes durante o processo de modelagem tridimensional. Frequentemente, é necessário editar objetos que estão ocultos por outros objetos na cena. Quando se muda o ponto de vista para tentar corrigir este problema, outros objetos se colocam entre a câmera e o objeto em questão, dificultando a edição e aumentando o tempo de trabalho e a frustração do usuário por não conseguir realizar a tarefa.

Com diferentes modos de visualização, é possível, por exemplo, utilizar um dos modos que representam os objetos em estrutura de arame (sem malha, *wireframe* ou *volumewire*) para conseguir ver e editar o objeto através de outros objetos. O modo transparente também pode ajudar na identificação destes objetos ocultos em outros modos de visualização. Esta possibilidade melhora o entendimento da cena, mostrando a relação de distancia entre todos os objetos e a perspectiva da cena. A figura 11 mostra os quatro primeiros modos de visualização. A Figura 12 mostra os modos de visualização restantes.

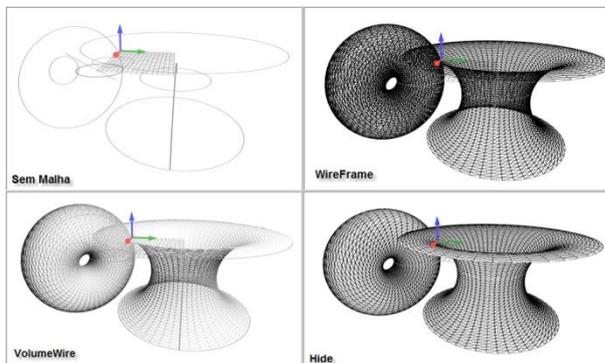


Figura 11: Modos de visualização: Sem malha, WireFrame, Volume Wire e Hide.

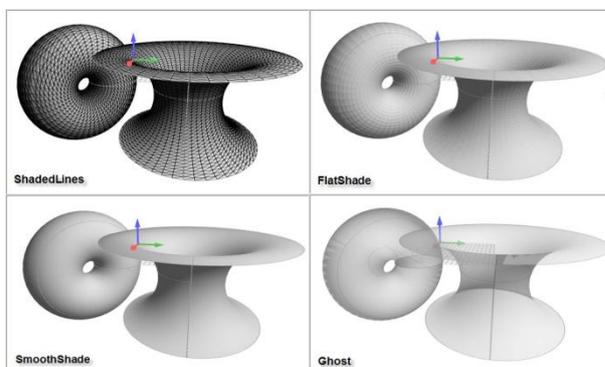


Figura 12: Modos de Visualização: Shaded Lines, FlatShade, SmoothShade e Ghost.

Quando um objeto é criado, especialmente as superfícies, sua representação 3D aparece imediatamente conforme o modo de visualização corrente e a cor do material utilizado; os diferentes modos de visualização permitem diferentes

percepções dos objetos; é possível ainda controlar a resolução da malha de visualização, controlando assim, caso necessário, o desempenho gráfico do aplicativo.

A Figura 13 mostra uma superfície de revolução com dois níveis diferentes de resolução de malha utilizando o modo de visualização Hide (remoção de linhas ocultas). Na imagem à esquerda, a superfície paramétrica é dividida em vinte (20) linhas e vinte (20) colunas, resultando em uma malha com oitocentos (800) triângulos. Na imagem à direita, a mesma superfície é representada por uma malha com cinquenta divisões em cada direção, resultando em uma malha com cinco mil (5.000) triângulos.

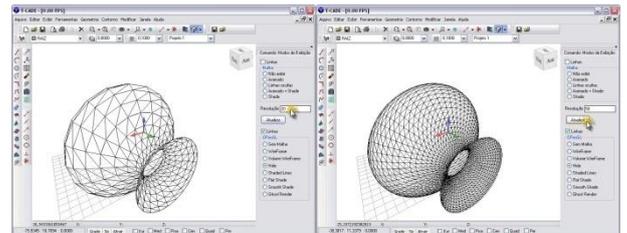


Figura 13: Mudando a resolução da malha.

Os diferentes modos de visualização têm desempenhos gráficos diferentes em função dos algoritmos específicos usados em cada implementação. Alguns modos podem facilitar a visualização de objetos que estejam escondidos atrás de outro objeto. Outros podem, também, ter grande influência no desempenho gráfico, principalmente no que diz respeito à eficiência computacional do próprio modo de visualização.

Assim, os modos de visualização são importantes para a representação dos objetos, influenciando diretamente o entendimento da cena tridimensional e o desempenho da interface.

4.1.7 Biblioteca de materiais

O T-CADE já permitia, conceitualmente, a utilização de materiais. Na interface original isto se refletia apenas na cor do objeto. Na nova interface, criou-se uma biblioteca de materiais padrão para representar corretamente os diferentes modos de visualização. Quando o programa é iniciado, esta biblioteca padrão é criada. A partir da definição de um novo material na estrutura de dados do T-CADE, uma nova biblioteca de materiais é criada para representá-lo e fica associado a ele na estrutura de dados. A Figura 14 mostra a interseção entre duas superfícies definidas com materiais diferentes.

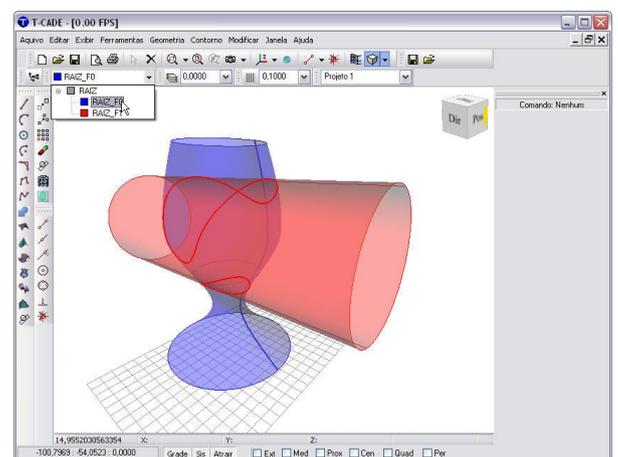


Figura 14: As cores dos objetos evidenciam os diferentes materiais.

Com a nova interface, cada vez que um objeto é criado, é associado a um material, e este material está ligado a uma biblioteca de materiais, com o potencial de aplicação de texturas, inclusive. A aplicação de texturas nos objetos não faz parte dos objetivos desta pesquisa, portanto, não foram criadas as ferramentas necessárias para isto. No entanto, a tecnologia e a estrutura de dados necessários para a criação destas ferramentas já foram implementadas neste trabalho e permitem seu futuro desenvolvimento.

4.2 Usabilidade

A interface de um aplicativo é o mediador entre o aplicativo e seu usuário, assim a comunicação entre eles deve ser clara e eficiente. O usuário deve saber sempre o que o programa espera que ele faça, especialmente no momento de fornecer os dados a um determinado comando. Considerando que o olhar do usuário acompanha sempre o cursor, ao invés de se usar uma barra de status para instruções do usuário, como é usual em programas 3D, foi criado um texto informativo logo abaixo do cursor, indicando então, qual o próximo passo em um comando ou que dados o programa espera que o usuário forneça naquele momento.

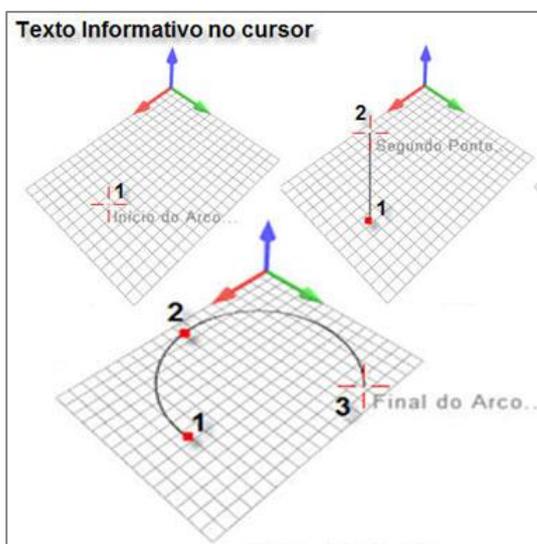


Figura 15: O texto no cursor informa o status da ferramenta ou o que o programa espera que o usuário faça naquele momento.

Esta abordagem evita que o usuário não perceba uma determinada informação relevante para a correta execução do comando, uma vez que esta informação está colocada na área de foco visual do usuário. Este texto informativo mostrou-se muito útil no treinamento de novos usuários. A figura 15 mostra o texto informativo no cursor durante a execução do comando “Arco por 3 pontos”.

Ao invés de fazer um longo estudo de usabilidade com muitos usuários levando em conta a interface terminada, o que seria além de demorado, muito dispendioso, optou-se por usar os recursos disponíveis para fazer, ao longo do desenvolvimento do projeto, inspeções de usabilidade nas diversas ferramentas e técnicas implementadas, desta forma elas puderam ser revisadas e melhoradas à medida que os problemas de usabilidade foram sendo identificados.

5. RESULTADOS

A utilização de uma interface gráfica baseada em OpenGL permite que o software T-CADE tenha um melhor desempenho gráfico quanto a velocidade de atualização da tela, que um maior número de polígonos sejam manipuláveis e visíveis

em tempo real, uma correta representação de superfícies resolvendo problemas complexos de visibilidade, transparência, sombreamento, seleção de objetos, iluminação, etc.

Para uma avaliação comparativa do desempenho gráfico em ambas interfaces, foram criadas rotinas com “Timers” (temporizadores) para a medição da taxa de atualização da Tela, medida em Quadros por segundo (FPS – frames per Second). A Tabela 1 mostra os resultados médios obtidos:

Tabela 1: Desempenho gráfico medido em Quadros por Segundo (FPS)

Objetos	Interface Original	Nova Interface
Mil Linhas	60 – 80 FPS	30 – 40 FPS
10 Mil Linhas	15 – 25 FPS	10 – 20 FPS
400 Faces	10 – 20 FPS	40 – 50 FPS
900 Faces	05 – 15 FPS	40 – 50 FPS
2500 Faces	02 – 10 FPS	40 – 50 FPS
5 Mil faces	00 – 05 FPS	40 – 50 FPS
10 Mil faces	0 FPS (Mais de 1 seg.)	35 – 45 FPS

Considerando a simplicidade da interface original, ela apresenta um melhor desempenho no desenho de linhas, uma vez que as rotinas de otimização da biblioteca gráfica OpenGL na nova interface continuam sendo calculadas, mas tem pouco ou nenhum efeito no desenho de linhas, porque nas linhas não temos o efeito da oclusão de objetos (um objeto ficar “escondido” atrás do outro). Quando se trata da representação de superfícies o desempenho da nova interface é muito superior. Além de manter uma taxa de atualização de tela estável, independente das circunstâncias, mantém-se regularmente acima do padrão estabelecido para o cinema de 30 quadros por segundo. Embora utilizada nos teste, é improvável a situação em que existam apenas linhas, em um projeto 3D os objetos são representados por superfícies e/ou sólidos, sendo as linhas utilizadas para a construção destes.

Quanto à interatividade, a interface implementada permite que, durante a criação do objeto, o resultado das ações do usuário seja imediatamente mostrado, antes mesmo de o comando ser finalizado, permitindo que o usuário corrija suas ações caso necessário.

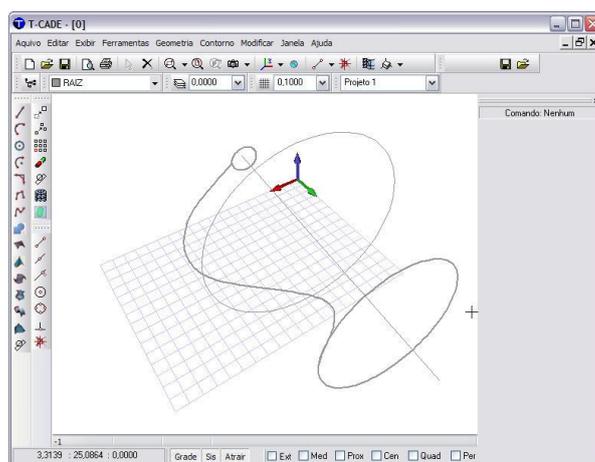


Figura 16: Representação de uma superfície de Revolução na interface original.

Um dos exemplos mais significativos da melhoria apresentada na nova interface se verifica na criação de superfícies. Na interface original, quando uma superfície é criada, são representadas somente as isolinhas da superfície (linhas que mantêm uma coordenada paramétrica constante), mas sua malha de representação não é mostrada automaticamente.

te. A Figura 16 mostra as linhas representantes de uma superfície de revolução, a curva geradora e os paralelos no início, meio e final da curva na interface original.

Assim, para que o usuário pudesse visualizar a malha da superfície era necessário o acionamento de outros comandos. Após três cliques com o cursor, a superfície era mostrada como esperado, primeiro para mostrar um painel de modos de visualização e depois para escolher o modo de visualização e, finalmente, para atualizar a malha de representação, conforme mostra a Figura 17. Isto era feito, justamente, para contornar as deficiências no desempenho, uma vez que a representação de superfícies era o que limitava a interface original. É importante ressaltar, ainda, que os modos de visualização da interface original são modos básicos, não permitem a suavização da superfície nem mesmo a aplicação de transparências.

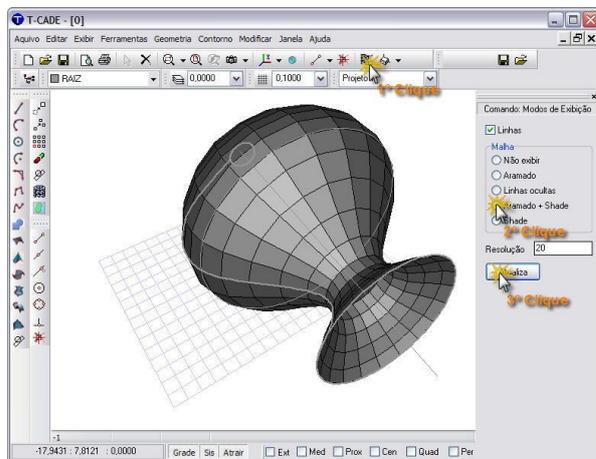


Figura 17: Três passos necessários para a geração da malha de visualização da superfície na interface original.

A criação de superfícies na nova interface já mostra automaticamente a malha da superfície, bem como suas linhas de construção, sem a necessidade do acionamento de novos comandos para sua representação. O modo de visualização padrão é o SmoothShade (modo Suavizado), conforme mostrado na Figura 18.

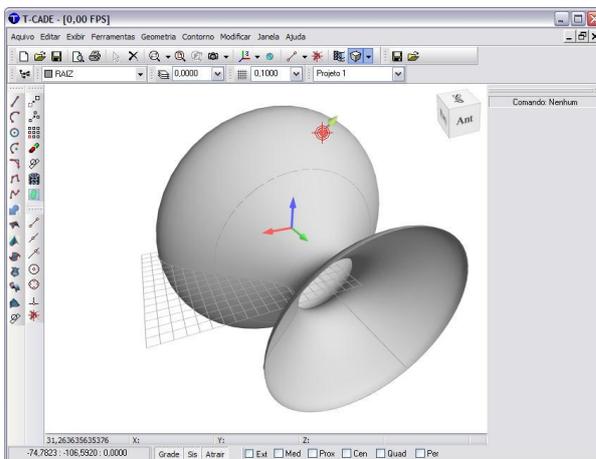


Figura 18: Criação de uma superfície de Revolução na nova interface.

Além do modo de visualização suavizado (SmoothShade), outra inovação foi o modo Ghost (transparente). Este modo permite ao usuário uma melhor compreensão da cena, tanto da geometria dos objetos como, da relação entre elas, uma vez que as superfícies não ocultam completamente outras

superfícies, que de outra forma não estariam visíveis. Um bom exemplo do uso do modo de visualização transparente é a visualização das linhas de interseção entre as superfícies.

A plataforma T-CADE pode determinar as linhas de interseção entre duas superfícies quaisquer e, a partir destas interseções, determinar os domínios da superfície (partes da superfície onde será gerada a malha). A Figura 19 mostra a interseção entre duas superfícies (linha em vermelho) por um ponto de vista que naturalmente ocultaria esta linha se as superfícies não fossem transparentes. Nesta imagem, também é possível ver a deformação em perspectiva da superfície cilíndrica, o que ajuda a perceber a tridimensionalidade da cena.

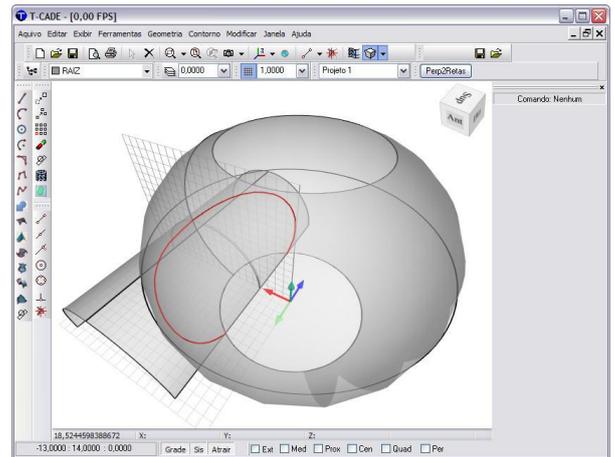


Figura 19: Modo Ghost mostrando a interseção entre superfícies.

Utilizando um exemplo mais realístico, a Figura 20 mostra a sequência de modelagem de um avião, partindo-se dos perfis transversais e linhas de base até a geração das superfícies utilizando os comandos Loft e Revolução. Evidencia-se na imagem maior, a superfície que marca o corte do vidro da cabine e a superfície das asas penetrando no corpo do avião.

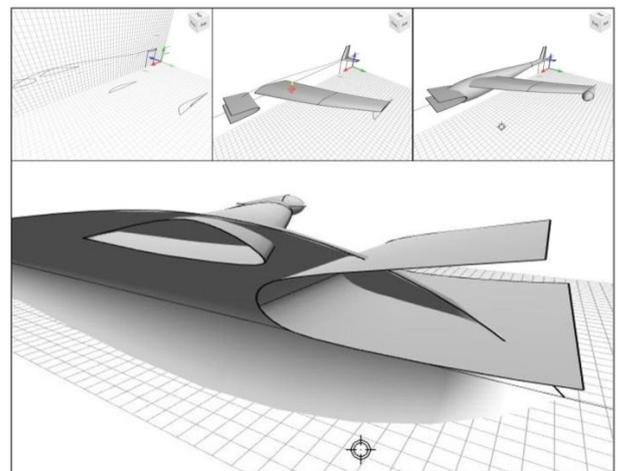


Figura 20: Modelagem com interseção de superfícies.

Uma das principais funcionalidades do T-CADE é a geração de malhas não-estruturadas para análise de elementos finitos. Na criação das superfícies, uma malha simples de visualização é criada automaticamente conforme a resolução corrente e é apresentada conforme o modo de visualização corrente (Hide, SmoothShade, Ghost). A partir das interseções entre as superfícies, é possível gerar uma malha não-

estruturada somente na parte relevante da superfície.

A geração desta malha segue parâmetros específicos e a visualização dos resultados é fundamental neste processo. A Figura 21 mostra a malha não estruturada (malha para análise / laranja) no corpo do avião em contraste com a malha estruturada (de visualização / cinza) na asa do avião. Esta figura mostra, ainda, a parte do corpo correspondente ao vidro da cabine, já cortada.

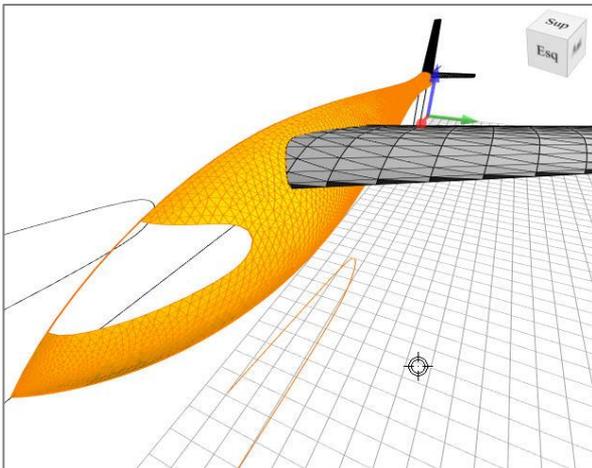


Figura 21: No modo ShadedLines é possível visualizar os elementos da malha.

A Figura 22 mostra uma vista interna do corpo do avião, onde se pode notar a junção do corpo do avião com asa e a parte da asa (dentro do corpo) em que a malha não foi gerada. A figura foi criada utilizando o modo SmoothShade, que valoriza a suavização e o sombreamento da superfície.

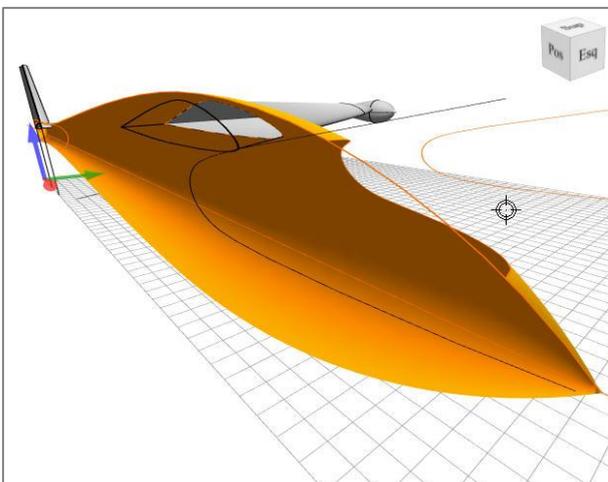


Figura 22: Interseção entre asa e corpo do avião.

Outra melhoria na representação de superfícies é percebida na edição de superfícies. Na interface original, quando uma superfície é movida ou copiada, apenas suas linhas de representação acompanham o cursor e são representadas na nova posição, enquanto que sua malha permanece no mesmo local de origem. Em uma primeira instância, isto causava confusão no usuário inexperiente sobre a eficácia do comando. Além disto, a correta representação da superfície exigia que fossem acionados novamente os comandos de atualização da malha, para só então mostrar a malha de representação da superfície no local correto. A Figura 23 ilustra a representação de uma superfície quando movida na interface original.

Tanto para a ação de mover quanto para a de copiar, era necessário o acionamento do comando via barra de ferramentas ou menu, a seleção dos objetos, o término da seleção, a indicação de um ponto de base, e por fim, a indicação de um novo ponto para onde os objetos selecionados seriam movidos. Na nova interface, opções para a manipulação direta de objetos foram criadas para que se pudesse selecionar, mover ou copiar os objetos diretamente a partir de um único movimento.

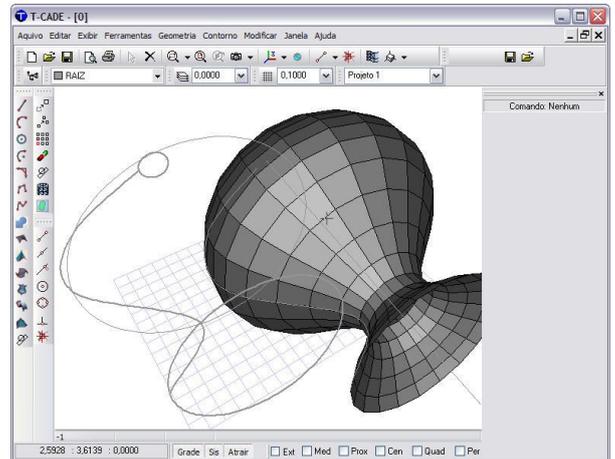


Figura 23: Objeto movido para esquerda, sua malha permanece no local de origem.

Além desta melhoria, a sua representação também é atualizada constantemente ao longo do processo, mostrando não só a nova posição da superfície selecionada como também a posição original, auxiliando o usuário ao criar esta referência visual. Enquanto a superfície é movida, uma linha é mostrada indicando a trajetória entre o ponto de base e o ponto final, sendo que esta linha também é atualizada interativamente. Ao final do comando, a superfície na posição original desaparece, restando apenas a superfície selecionada na nova posição. A Figura 24 mostra a nova posição da superfície selecionada, a linha da trajetória do cursor e a posição original do objeto.

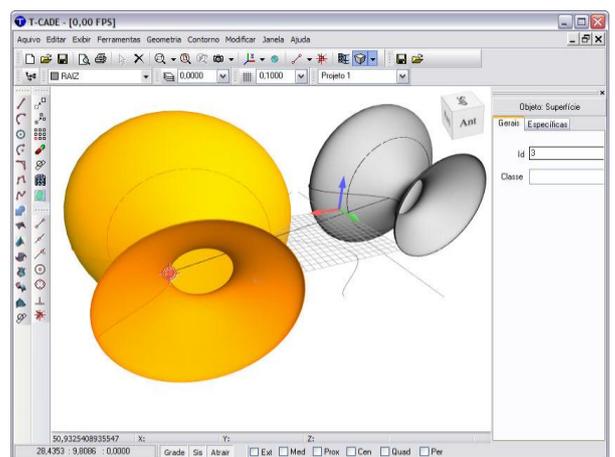


Figura 24: Superfície sendo movida na nova interface.

A partir da combinação de teclas de modificação (Ctrl, Alt e Shift), é possível alternar entre o movimento no modo ortogonal (eixos X, Y e Z) e entre mover ou copiar, tudo isto ainda durante o movimento. Tanto o formato do cursor como a cor da linha da trajetória do cursor se adaptam às modificações do comando, indicando seu estado, conforme mostrado

anteriormente na Figura 29 (pág. 89).

O uso destas combinações de teclas com os comandos de manipulação de objetos aumentou o controle do usuário, tornou mais ágil e facilitou o processo de transformação da cena tridimensional. Neste caso, a redução do número de cliques com o mouse tem reflexo direto no tempo de execução do comando, permitindo que o usuário execute sua tarefa mais rapidamente e com maior precisão.

No que se refere à usabilidade, ao invés de fazer um longo estudo com muitos usuários, levando em conta a interface terminada, o que seria, além de demorado, muito dispendioso, optou-se por usar os recursos disponíveis para fazer, ao longo do desenvolvimento do projeto, inspeções de usabilidade nas diversas ferramentas e técnicas implementadas. Desta forma, estas ferramentas e técnicas puderam ser revisadas e melhoradas na medida em que os problemas de usabilidade foram sendo identificados.

6. CONSIDERAÇÕES FINAIS

Este trabalho mostrou o desenvolvimento de um protótipo de interface para interação tridimensional levando em conta conceitos de usabilidade, interatividade e desempenho gráfico. Para a melhoria do desempenho gráfico, foi implementada a biblioteca gráfica *Glscene* que, nos testes apresentados, manteve uma taxa de atualização de tela acima de 30 quadros por segundo (padrão de cinema) de modo relativamente constante.

Com a melhoria do desempenho gráfico, é possível trabalhar com até um milhão de polígonos (no modo de visualização menos eficiente) sem que a interface seja um limitador do processo de interação em tempo real.

O aumento da interatividade melhora a experiência do usuário e estabelece as condições necessárias para o desenvolvimento de novas ferramentas de criação e edição de objetos, imprescindíveis em programas para modelagem tridimensional.

Melhorias na seleção de objetos também foram implementadas. A seleção de múltiplos objetos através de uma janela de seleção pode ser feita de duas maneiras diferentes: abrindo-se uma janela da direita para a esquerda, a janela fica com um fundo verde transparente e uma borda com linha tracejada, indicando que todos os objetos tocados pela janela (total ou parcialmente) serão selecionados; abrindo-se a janela da esquerda para a direita, a janela fica com um fundo azul transparente e uma borda com linha contínua, indicando que somente os objetos inteiramente dentro da janela serão selecionados.

A seleção de superfícies também foi melhorada, na nova interface quando o cursor é passado sobre uma superfície o ponto 3D na superfície é automaticamente reconhecido, bem como o vetor normal da superfície naquele ponto, este reconhecimento do ponto 3D tem o potencial de criar novas ferramentas de edição destes objetos ou criação de outras superfícies a partir destas seleções.

Para a melhoria da interatividade, foram criadas duas ferramentas de visualização e implementadas técnicas de manipulação direta de objetos. Uma destas ferramentas foi o *Viewcube*, que permite a rápida mudança entre vistas pré-definidas de modo animado e de fácil utilização. A outra ferramenta de visualização, o *QuadMenu*, facilita a seleção de comandos de navegação (controle do movimento da câmera) e permite sua utilização de modo transparente (sem interrupção dos comandos acionados).

Além destas ferramentas interativas, a representação das superfícies também mostra uma melhoria significativa em relação à interface original, tanto nas possibilidades dos

modos de visualização (suavização e transparência), como na representação interativa dos objetos, nas etapas de criação e na edição de linhas e superfícies. Os diferentes modos de visualização possibilitam ao usuário escolher como as superfícies serão representadas. Esta escolha tem influência direta no desempenho e, por consequência, na interatividade.

Em casos em que a visualização dos triângulos da malha não estruturada (para análise de elementos finitos) seja necessária, pode-se usar, por exemplo, o modo *ShadedLines* (arestas visíveis e sombreado). Este modo é o que tem o maior custo computacional em termos de desempenho gráfico. Quando a visualização da superfície for mais importante do que a disposição dos triângulos da malha, pode-se usar o *SmoothShade* (suavizado) ou o modo *Ghost* (transparente), melhorando muito o desempenho, a interatividade e a representação visual da superfície.

Além destas, outras melhorias foram implementadas, como diferentes cursores indicando o estado atual do aplicativo, representação de superfícies e principalmente na estrutura de dados para representação dos objetos.

Os resultados comparativos do desempenho gráfico entre a interface original e nova interface certificam a melhoria. As ferramentas e técnicas criadas para melhoria da interatividade tornaram o trabalho do usuário mais fácil e rápido nesta plataforma.

A criação das ferramentas gráficas em conjunto com as técnicas de manipulação tornou a experiência do usuário mais interativa, indicando que os objetivos foram alcançados. Com uma interface para interação tridimensional adequada às necessidades da plataforma T-CADE em termos de desempenho, interatividade e usabilidade, esta poderá agora continuar a ser desenvolvida, uma vez que as inovações pretendidas para a plataforma dependiam de uma interface robusta, ágil e confiável.

REFERÊNCIAS

- [1]. DOS SANTOS, Sérgio Leandro. Concepção e Desenvolvimento de uma Interface Gráfica para Interação Tridimensional. Porto Alegre, 2009. 108p Dissertação (Mestrado em Design) Programa de Pós-Graduação em Design, UFRGS, 2009.
- [2]. GLSCENE OFFICIAL SITE, acessado em 24 de novembro de 2007. <http://www.glscene.org>
- [3]. HEWETT, T. *Curricula for Human-Computer Interaction*. **ACM SIGCHI Report**. 1992. Disponível em: <<http://sigchi.org/cdg/>>. Acesso em: 25 Agosto de 2008.
- [4]. KETTNER, Lutz. Theoretical Foundations of 3D-Metaphors. In Workshop on the Challenges of 3D Interaction at the CHI'94, february 1994.
- [5]. KHAN, A., MORDATCH I., G. FITZMAURICE, J. MATEJKA, e G. KURTENBACH. "ViewCube: A 3D Orientation Indicator and Controller." **SIGGRAPH**. New York: ACM, 2008. p.17-25.
- [6]. NIELSEN, Jakob. Usability engineering. EUA: Morgan Kaufmann, 1993.
- [7]. NIELSEN, J. e MOLICH, R. *Heuristic evaluation of user interfaces*. **Proc. CHI'90** Conference on Human Factors in Computer Systems., 1990: p249- 256.
- [8]. OPENGL OFFICIAL SITE, acessado em 24 de novembro de 2007. <http://www.opengl.org>.
- [9]. PREECE, J.; ROGERS, Y.; SHARP, H.; BENYON, D.;

- HOLLAND, S. e CAREY, T. **Human-Computer Interaction**. London: Addison-Wesley, 1994.
- [10]. SHNEIDERMAN, Ben. Direct manipulation: a step beyond programming languages. *IEEE Computer* 16(8) (August 1983), p57-69.
- [11]. TEIXEIRA, F.G.: Modelamento paramétrico e geração de malha em superfícies para aplicações em engenharia; Porto Alegre: UFRGS, 2004. Originalmente apresentada como tese de Doutorado, Engenharia mecânica da Universidade Federal do Rio Grande Sul, 2003.
- [12]. TEIXEIRA, F.G., T.L.K. SILVA, R.P. SILVA, e J.L.F. AYMONE. *Virtual Design: Concepts*. **SAE Technical Paper Series 2008-36-0332.**, Warrendale: Society of Automotive Engineers, Inc., 2008a.
- [13]. TEIXEIRA, F.G., T.L.K. SILVA, R.P. SILVA, e J.L.F. AYMONE. *Virtual Design: Technologies*. **SAE Technical Paper Series 2008-36-0341.**, Warrendale: Society of Automotive Engineers, Inc., 2008b.
- [14]. ZELENNIK, R., e A. Forsberg. "UniCam—2D gestural camera controls for 3D environments." *SIGGRAPH*. New York: ACM, 1999. p169-173.