

Modelagem de objetos 3D em VRML: Uma implementação multiplataforma orientada ao ensino

André Tavares da Silva[†]
Fernando Santos Osório[‡]

[†]INFORMAÇÃO – Acessoria em Informática Ltda.
Caixa Postal 82 – 95.840-000 – Triunfo – RS – Brazil

[‡]UNISINOS – Universidade do Vale do Rio dos Sinos
Curso de Informática / Mestrado em Computação Aplicada
Av. Unisinos, 950 – 93.022-000 – São Leopoldo – RS – Brazil
atavares@cpovo.net, osorio@exatas.unisinos.br

Abstract. *VRML (Virtual Reality Modeling Language) is a language oriented to the Internet and it is a powerful tool for the creation of new applications on Virtual Reality. The objective of this work is the integration of modeling techniques and the VRML through a software we developed. In order to do that, the sweep modeling technique was chosen. The modeling system developed in this work, written in Java programming language, can be used either as an executable program or as an applet in an Internet browser together with a VRML viewer. This software is distributed freely for educational use.*

This modeling system, with simple user interface and straightforward operation, does not require an advanced knowledge of modeling techniques as the commercial systems usually do. Also, it is a platform independent program and can be used through the Internet, since it was written in Java. Thus, it is also a tool with a didactic appeal that allows the students to examine the sweep modeling technique in a simple system, and it can be used to introduce some concepts about Virtual Reality.

1. Introdução

Um dos sentidos humanos mais ricos e diversificados é a visão, cuja capacidade de percepção nos faz interagir com o mundo real, onde podemos gerar novas ações que irão modificá-lo. Atualmente, quando a proliferação da computação atinge praticamente todos os setores da sociedade, a tendência metafórica é justamente atingir o máximo de proximidade e interação com a “máquina humana”. Temos exemplos da área de inteligência artificial, que procura imitar o cérebro, e por outro lado, ainda mais atraente, a computação gráfica que procura desenvolver a habilidade de representação visual das máquinas. O forte apelo das imagens seduz o ser humano mais do que as palavras, e quanto mais próximo da realidade for o apelo visual, mais confortável e próximo da sua própria realidade o homem se sentirá.

Cada vez mais, a computação gráfica tem sido utilizada para a criação de imagens que representam modelos do mundo real [Foley 1996, Hermida 1994]. Neste sentido, através da realidade virtual [Adams 1994, Cadoz 1997], tenta-se criar um ambiente no qual a pessoa possa sentir-se imersa total ou parcialmente em um mundo virtual, onde objetos imaginários podem ser sentidos e manipulados.

A realidade virtual é uma tecnologia que permite uma melhor interface homem-máquina, e, junto com a Internet através da VRML [Web 1999], se torna uma ferramenta muito poderosa para permitir a criação de novas formas de resolução de problemas. É portanto, um novo meio de comunicação, na qual pode-se manipular a informação através de uma “*experiência em 1ª pessoa*”.

Este trabalho tem por objetivo apresentar a integração de técnicas de modelagem interativa, juntamente com a VRML, através da implementação prática de um software.

A técnica de modelagem escolhida para fazer esta integração foi a técnica de modelagem por *sweep*, por ser uma maneira prática e fácil de se modelar uma grande variedade de objetos do mundo real [Casacurta 1991]. A interface do sistema de modelagem desenvolvido junto a este trabalho é bastante simples de usar, podendo o software ser usado não só como aplicativo para permitir a gravação dos objetos gerados (descrição geométrica poligonal), mas também ser executado através da Internet, juntamente com um visualizador VRML, já que a VRML está voltada principalmente para a Internet. Para isto ser possível, a linguagem usada para o desenvolvimento do software foi a linguagem Java [Javasoft 1999].

Vamos portanto descrever neste trabalho o sistema “*ATSWorlds*”, inteiramente desenvolvido na Unisinos [Silva 1999], com a finalidade de criar objetos e mundos em VRML, modelados através de diferentes tipos de *sweep*. Este sistema já se encontra operacional, sendo usado em estudos sobre a modelagem de objetos sólidos de revolução (*sweep*) e em estudos práticos sobre a linguagem VRML. Este software é distribuído livremente, podendo também ser acessado através da Internet no seguinte endereço: <http://www.inf.unisinos.br/~osorio/ATSWorlds.html>

2. Modelagem Geométrica

Um sistema de realidade virtual deve ter a capacidade de criar um “mundo virtual”, inserindo objetos neste mundo, e assim determinando: forma, posição, transparências, reflexões e texturas dos objetos, entre outros aspectos. Para isto, é necessário inicialmente modelar os objetos como sólidos geométricos, e posteriormente atribuir a estes objetos propriedades físicas como: transparência, reflexão, textura, etc.

A necessidade de modelar objetos como sólidos levou ao desenvolvimento de uma grande variedade de representações. A área de modelagem geométrica [Gomes 1998, Foley 1996, Diegues 1989, Mortenson 1985] trata deste problema da criação, manipulação e topologia dos objetos gráficos no computador. Nesta seção daremos uma breve introdução a algumas das principais técnicas usadas para representar objetos em sistemas de computação gráfica.

2.1. Instanciamento de Primitivas

Em geral, as primitivas geométricas são objetos simples de descrever e representar, constituindo os blocos básicos da construção de modelos (e.g. esferas, cubos, pirâmides). Estas primitivas podem sofrer algumas transformações: translação, rotação, mudança de escala, e deformações. Tais transformações são utilizadas tanto para posicionar as primitivas no espaço quanto modificar a geometria destas. No posicionamento das primitivas são utilizados os movimentos de rotação e translação. Para modificar a geometria, uma transformação bastante utilizada é a mudança de escala, que permite uma mudança das dimensões da primitiva. Usando uma mudança de escala linear em apenas um dos eixos, pode-se transformar um cubo em um paralelepípedo qualquer. Usando transformações projetivas, pode-se obter figuras das mais variadas através de um simples cubo. O uso de primitivas básicas adicionadas as transformações para modificar a geometria destas permite reduzir a complexidade final do modelo resultante deste tipo de processo de modelagem.

2.2. Representação de Limites

O modelo por representação de limites define um sólido indiretamente através da representação das suas superfícies limitantes. Computacionalmente é conveniente dividir uma superfície do modelo em faces, sendo que cada face é limitada por um conjunto de vértices e arestas. Os limites de um objeto podem ser divididos em faces, vértices e arestas de forma ilimitada, ou seja, não existe uma forma única de representar os limites de um objeto.

2.3 Representação por decomposição volumétrica

Este é um modelo por decomposição, ou seja, os sólidos são descritos através da união de blocos básicos. Primitivas podem variar no tipo, tamanho, posição, parametrização e orientação, e a maneira como os blocos são combinados é que distingue cada variação deste esquema de representação.

São modelos de representação por decomposição volumétrica:

- Enumeração Exaustiva
- Representação por “Octree”
- Decomposição de Células
- Árvore Binária de Decomposição Volumétrica

2.4 Geometria Sólida Construtiva – CSG

O método mais conhecido de representação por construção é o método CSG, que vem do inglês “*Constructive Solid Geometric*” (Geometria Sólida Construtiva) no qual sólidos complexos são compostos de primitivas geométricas através de transformações no espaço e operações booleanas.

As transformações são utilizadas na representação CSG com a finalidade de posicionar as primitivas no espaço ou modificar a geometria de uma primitiva. No posicionamento são utilizados movimentos de rotação e translação. Para modificar a geometria das primitivas é bastante usada a mudança de escala, permitindo uma mudança das dimensões das primitivas. Usando essas transformações a primitiva dada por um cubo pode ser transformada para obter paralelepípedos das mais variadas dimensões. Também pode-se obter um prisma qualquer usando transformações projetivas. O uso de transformações para modificar primitivas permite reduzir o número de primitivas.

Após transformar e posicionar devidamente as primitivas no espaço, o sistema CSG se utiliza das operações booleanas para combinar as diversas primitivas e criar o modelo final. As operações booleanas são a união (\cup), interseção (\cap) e diferença ($-$) de conjuntos.

Os objetos são armazenados como uma árvore em que os operadores são nodos internos e as primitivas são as folhas. Os nodos podem representar operadores booleanos ou transformações (translação, rotação ou escala).

2.5 Sweep

O sweep [Diegues 1989, Casacurta 1991] permite criar sólidos de revolução, onde a partir de um contorno descrito em 2D realiza-se um deslocamento e a medida que este contorno descreve uma trajetória no espaço 3D são gerados os polígonos que irão compor o objeto final. Por exemplo, uma simples linha rotacionada 360 graus ao redor de um eixo permite que se modele um tubo. ***O sweep é uma maneira natural e intuitiva de se construir uma grande variedade de objetos.*** Por esta razão, muitos sistemas de modelagem permitem ao usuário construir objetos por *sweep*. Devido as suas características, esta foi a técnica escolhida para realizar a integração com o VRML [Web 1999], portanto, ela será detalhada na próxima seção.

2.6 Considerações Finais Sobre Modelagem Geométrica

Nesta seção foram apresentadas algumas das principais técnicas de modelagem geométrica existentes usadas para representar objetos, dando uma breve introdução das mesmas. Como o objetivo deste trabalho era o de integrar técnicas de modelagem com a VRML, era necessário que se optasse por uma técnica dentre essas descritas acima para realizar essa integração.

A técnica escolhida para a realização do trabalho foi a técnica de modelagem por *sweep*. A principal razão de nossa escolha foi o fato de que a definição dos objetos pode ser feita de forma bastante simples, através de um esboço de um contorno em 2D, e de alguns poucos parâmetros, dando origem à sólidos de revolução bastante interessantes.

Do ponto de vista prático, muitos destes sólidos gerados por *sweep* exigiriam um grande número de operações para serem modelados por outras técnicas, e além disto, muitos processos de fabricação de peças e objetos na indústria utilizam-se de conceitos similares aos empregados na modelagem por *sweep* (e.g. tornos e extrusores). Por isto, esta técnica será o tema do próximo seção, onde também serão descritos seus principais tipos. Também serão descritos alguns dos algoritmos implementados e a descrição da construção do sistema de modelagem desenvolvido junto a este trabalho.

3. Implementação

Este trabalho tem por objetivo a integração de técnicas de modelagem com a VRML, através da implementação da técnica de *sweep*. Optamos pela criação de um sistema de modelagem com uma interface simples de usar, onde o usuário desenha seus próprios polígonos para gerar os objetos pela técnica de modelagem por *sweep*, com a possibilidade de ler e gravar arquivos contendo as descrições de polígonos.

A linguagem de programação escolhida para realizar a implementação de um sistema de modelagem por *sweep* que gere objetos VRML 2.0 [Web 1999, Duarte 1998], foi a linguagem Java da Sun Microsystems [Javasoft 1999]. Como a VRML está voltada principalmente para a Internet, o ideal para realizar esta integração seria um sistema que funcionasse também pela Internet, podendo ser transmitido junto com as páginas Web e sendo independente de plataforma. Nossa opção pela linguagem Java permite que este sistema seja aplicado em diferentes ambientes e facilitando o seu uso no junto ao ensino de conceitos de modelagem de peças (e.g. mecânica), computação gráfica e realidade virtual.

Existem atualmente vários visualizadores para VRML [Web 1999]. Grande parte destas ferramentas são de uso livre e estão disponíveis através da Internet, como os *plug-ins* usados para testar os modelos gerados pelo sistema de modelagem desenvolvido junto a este trabalho (Cosmo Player da SGI, Microsoft VRML Viewer, WorldView da Intervista e Viscape VRML da Superscape).

Resolveu-se então que o ideal para este sistema era que ele funcionasse no *browser* (Netscape / Microsoft Internet Explorer) juntamente com um visualizador VRML. Assim, não seria necessário ao usuário utilizar dois sistemas independentes: um para gerar os modelos e outro para visualizá-los, pois ambos poderiam ser integrados em uma mesma ferramenta. Mas, para que fosse possível gravar os objetos gerados, dadas as características da linguagem Java, também era necessário que se pudesse executar o sistema de modelagem como aplicação independente, já que pelo *browser* não é possível, por questões de segurança, realizar o acesso ao disco para a escrita de dados.

Assim, decidiu-se realizar a criação de uma ferramenta que seja tanto uma aplicação independente quanto uma aplicação que possa ser executada a partir do *browser*. Com a linguagem Java, isto é possível de se fazer, sendo também por este motivo a melhor linguagem para implementação do sistema de modelagem. Quando o sistema é executado pelo *browser*, é denominado de *applet*. Quando é executado por um interpretador Java fora do *browser*, é chamado de aplicação. Após ser decidida a forma de como o sistema iria gerar os modelos, o ambiente e a linguagem, o passo seguinte foi a definição da interface, que será objeto de estudo dos próximos itens.

O sistema de modelagem implementado junto a este trabalho recebeu o nome de ATSWorlds [Silva 1999]. A seguir são descritas e mostradas as duas interfaces deste sistema: executado independentemente (aplicação) e em um *browser* (*applet*). Depois, é feito uma descrição da integração do sistema de modelagem com a VRML. E, por último, são detalhados os tipos de modelagem por *sweep* implementados neste trabalho.

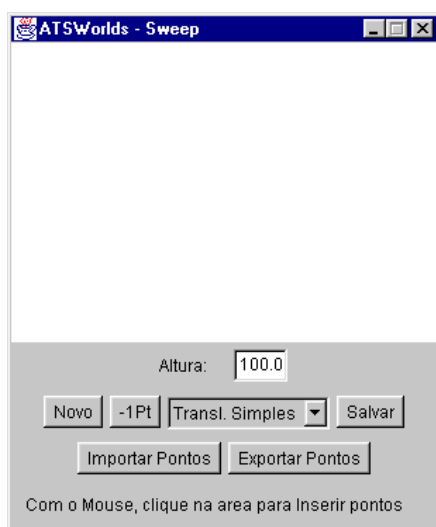
3.1 As Interfaces

O sistema implementado junto a este trabalho gera objetos VRML a partir da técnica de modelagem por *sweep*. Como foi visto anteriormente, este sistema de modelagem pode ser usado tanto através de uma aplicação independente quanto via um *applet*.

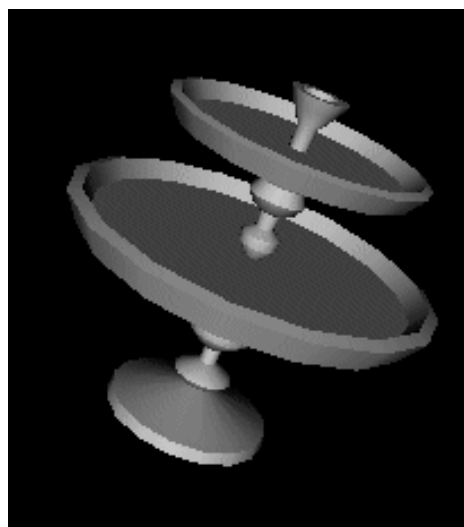
3.1.1 Aplicação

Para que se possa salvar os objetos que são gerados pelo sistema de modelagem, é necessário utilizar o ATSWorlds como aplicação. Neste modo, também é possível importar e exportar em arquivos os polígonos geradores do *sweep* em um formato próprio. A figura 1 mostra a interface do programa sendo utilizado como aplicação.

É interessante salientar que os arquivos com a descrição dos objetos poligonais e dos objetos em VRML, criados pela aplicação ATSWorlds, estão em formato texto, o que possibilita a sua análise posterior. Esta facilidade de se analisar o conteúdo dos arquivos gerados tem um papel bastante importante no ensino de linguagens como a VRML, e por consequência auxiliando a realização de estudos sobre a construção de mundos virtuais.



(a) Interface da aplicação



(b) Exemplo de um objeto gerado pela aplicação

Figura 1. ATSWorlds como aplicação.

A área em branco é o local onde são desenhados os polígonos (2D) que gerarão os objetos pela técnica de *sweep*. Para desenhar um polígono, basta clicar com o mouse em pontos desta área, marcando os pontos que definem os vértices deste. A função de *Undo* também foi implementada permitindo desfazer o último ponto, caso isto seja desejado.

Se for escolhido o tipo de *sweep* translacional cônico, será apresentado na área de desenho um ponto representado por uma pequena cruz vermelha, indicando as coordenadas do ponto de fuga. Este ponto pode ser movido da mesma maneira que os outros pontos, clicando nele e arrastando-o até a posição desejada.

Se forem escolhidos *sweeps* do tipo rotacional completo, rotacional parcial ou helicoidal, na parte direita da área de desenho será exibida uma linha vermelha que representa o eixo ao redor do qual será feita a rotação destes tipos de *sweep*.

A parte inferior da aplicação é denominada barra de status. Também é usada para dar mensagens de erro ou para mostrar as coordenadas do ponto que está sendo movido. As opções logo abaixo da área de desenho, dependem do tipo de *sweep* que se quer realizar, portanto são variáveis. Estas opções serão estudadas no item em que serão definidos os algoritmos de cada tipo de modelagem por *sweep* implementados neste trabalho.

Para iniciar um novo polígono, descartando o polígono existente na área de desenho, deve ser pressionado o botão **Novo**. Pressionando-se o botão **Exportar Pontos** será gravado o estado desta área de desenho, ou seja, os pontos formadores do polígono e a localização do ponto de fuga. Para ler um estado gravado anteriormente, ou ler pontos gerados por outra aplicação, deve ser usado o botão **Importar Pontos**,

permitindo através deste recurso de Exportar/Importar uma melhor integração com outras aplicações externas ao sistema.

Depois de desenhado o polígono, escolhido o tipo de *sweep* e ajustados os parâmetros necessários, basta clicar no botão **Salvar** para que o ATSWorlds grave um objeto VRML (tipo “WRV”) a partir desse polígono pela técnica de modelagem por *sweep* escolhida. Para visualizar o objeto gerado, deve-se ler o arquivo gravado pelo sistema de modelagem em um *browser* com um *plug-in* VRML instalado, ou ler o arquivo em algum outro visualizador capaz de ler arquivos VRML 2.0 ou superior.

3.1.2 Applet

O sistema quando usado como *applet*, permite que se visualize tanto o sistema de modelagem quanto o objeto gerado na mesma tela, sem a necessidade de ficar alternando de aplicativo cada vez que um novo objeto é criado ou alguma modificação é realizada. Neste modo, os objetos não podem ser gravados, pois o *browser* não permite isto. Sendo assim, o applet serve apenas como uma ferramenta de visualização e prototipagem de objetos, de uso didático, mas sem um aproveitamento prático imediato dos objetos modelados.

O applet do ATSWorlds é bastante útil para alunos que queiram examinar na prática a técnica de modelagem por *sweep* em um sistema bastante simples, podendo inclusive, executá-lo através da Internet (como em [Minho 1999]). Assim este sistema torna-se em uma excelente ferramenta didática para o ensino da área de computação gráfica, geometria (sólidos geométricos feitos por superfícies de revolução), entre outras possíveis aplicações.

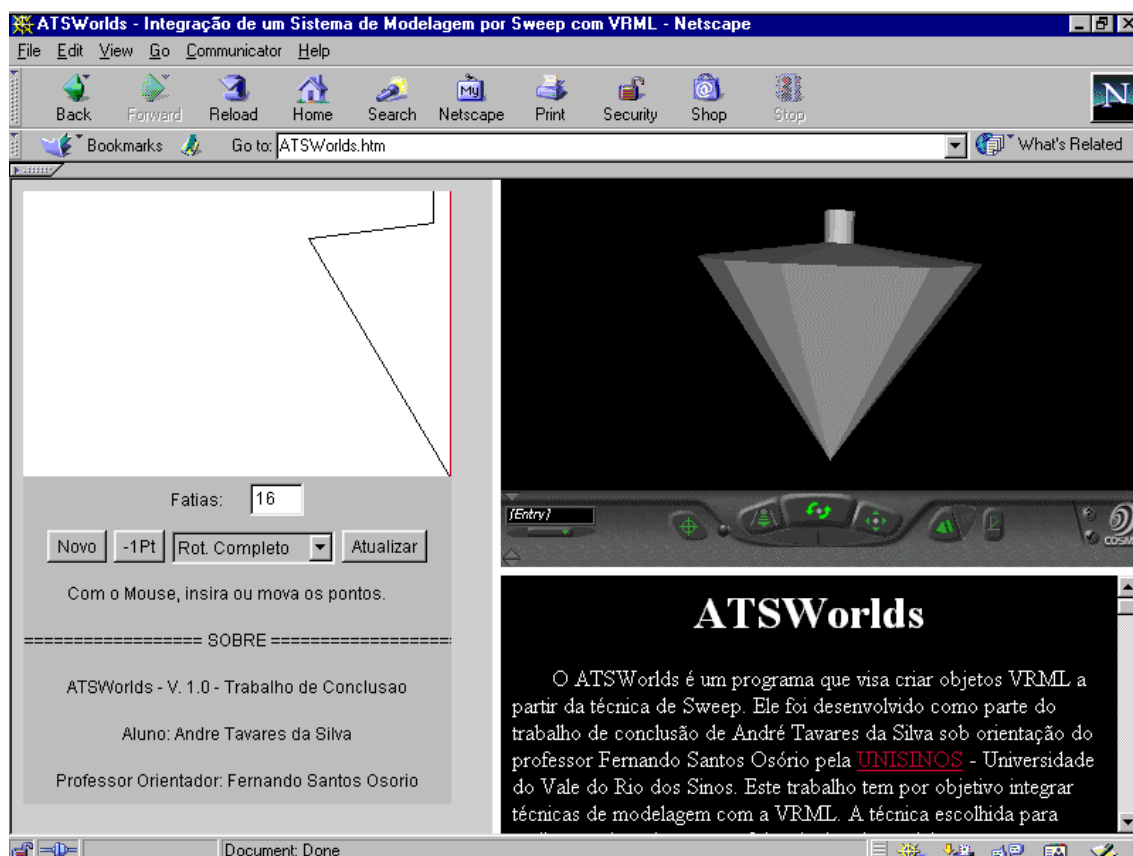


Figura 2. Applet ATSWorlds em uma página HTML.

A interface da *applet* e da aplicação são muito semelhantes, devido ao fato de serem ambas geradas a partir do mesmo sistema. O próprio sistema se encarrega de fazer a modificação da interface dependendo do fato de ser *applet* ou aplicação. Isto faz

com que a construção dos objetos seja feita de forma muito semelhante ao item anterior. A diferença básica é que no lugar do botão “Salvar”, na *applet* existe o botão **Atualizar**. Este botão faz com que o objeto VRML seja gerado e atualizado no visualizador do *browser*, em vez de salvar o objeto em disco. No *applet* não há os botões para importar e exportar o polígono.

Além do sistema de modelagem e do visualizador VRML, a página também contém uma parte contendo um texto explicativo sobre o sistema de modelagem, de como utilizar o ATSWorlds, e sobre alguns requisitos necessários à execução da página. Este texto poderia ser substituído por um roteiro de uma aula prática a ser realizada com o uso deste sistema.

A figura 2 mostra a interface do sistema de modelagem sendo executado como *applet* em uma página HTML. O ATSWorlds ocupa a parte esquerda da página, o visualizador na parte superior direita e na parte inferior direita o texto explicativo (neste exemplo foi usado o visualizador Cosmo Player da SGI e o *browser* Netscape).

Depois de desenhado o polígono, escolhido o tipo de *sweep* e ajustados os parâmetros necessários, basta clicar no botão **Atualizar** para que o ATSWorlds atualize o objeto na parte superior direita da página. Este botão deve ser usado toda vez que for alterado o polígono gerador, o tipo de *sweep*, ou os parâmetros e se queira visualizar o objeto a partir das mudanças feitas. Neste modo não é preciso ficar alternando entre programas, onde a alteração dos dados e a visualização dos resultados é feita de modo automático.

3.2 Técnica de Modelagem por Sweep

Os tipos de modelagem por *sweep* [Casacurta 1991] implementados no sistema ATSWorlds foram: Translacional Simples, Translacional Cônico, Rotacional Completo, Rotacional Parcial e Helicoidal. Com exceção do *sweep* Helicoidal, em que o polígono gerador deve ser fechado, em todos os outros tipos de *sweep* os polígonos geradores podem ser abertos ou fechados, permitindo uma maior flexibilidade em termos do objeto final obtido.

3.2.1 Sweep Translacional Simples

O tipo mais simples é o *sweep* translacional, no qual cada aresta desenhada é deslocada por uma trajetória normal ao plano em que estão as arestas (similar ao processo de extrusão). O sistema de modelagem foi elaborado de forma que, neste tipo de *sweep*, a origem do objeto gerado fique exatamente no centro da área de desenho. Esta área representa o plano XZ do sistema de coordenadas da VRML. Portanto, o deslocamento é feito no eixo Y (na vertical). Assim, se o desenho do objeto for realizado de forma centralizada na área de desenho, a origem ficará no centro geométrico do objeto formado. Tanto a base quanto o topo do objeto formado serão definidos pelo próprio polígono desenhado pelo usuário. A altura deste objeto é atribuída através do parâmetro “Altura” do ATSWorlds, informado no respectivo campo de entrada de dados da interface.

O algoritmo usado para a geração deste tipo de *sweep* é bastante simples. Para formar a lista de pontos do polígono, converte-se cada ponto do polígono desenhado pelo usuário para dois pontos no espaço da VRML. Ou seja, cada ponto formará um vértice da base e um vértice do topo do objeto. O eixo X do objeto VRML é representado pelo eixo horizontal da tela, o eixo Y é dado pelo parâmetro Altura, e o eixo Z do objeto é representado pelo eixo vertical da tela. As faces são feitas pela união destes pontos. As faces laterais são formadas pela junção dos pontos que formam cada aresta do polígono gerador. Como as arestas da base e do topo são paralelas, cada face lateral terá um formato retangular. As faces superior (topo) e inferior (base) são formadas pela união dos próprios pontos que as formam. O ATSWorlds também permite que seja feito o *sweep* translacional de um polígono aberto, para dar maior liberdade ao usuário, onde neste caso, serão gerados apenas as faces laterais, não fechando a poligonal definidora do topo e da base do objeto.

3.2.2 Sweep Translacional Cônico

Este tipo de *sweep* é uma extensão do translacional simples. Neste caso, somente uma das faces é igual ao polígono gerador desenhado pelo usuário. Os vértices da outra face, ao serem transladados, convergem para um ponto denominado de “ponto de fuga”. Esta face, portanto, sofre uma alteração de escala enquanto é deslocada. O ponto de fuga não necessita ficar no centro do polígono gerador. Isto faz com que este deslocamento possa também ser não perpendicular ao plano da base do objeto gerado.

A base do objeto formado será o próprio polígono desenhado pelo usuário. A altura do objeto é atribuída através do parâmetro “Altura”. O ponto de fuga é representado por uma pequena cruz vermelha na área de desenho e pelo parâmetro “Altura PF”. Isto facilita ao usuário visualizar a forma que terá o objeto. Se a altura do ponto de fuga for negativa, o *sweep* será do tipo translacional cônico divergente. E a parte menor, neste caso, será voltada para baixo. O sistema de modelagem também permite a criação de um objeto pela técnica de *sweep* translacional cônico de polígono aberto, gerando somente as faces laterais.

3.2.3 Sweep Rotacional Completo de Polígono Fechado

O *sweep* rotacional, faz uma superfície ser rotacionada sobre um eixo do mesmo plano desta superfície. Este eixo, no sistema de modelagem ATSWorlds, é representado por uma linha vermelha localizada na parte direita da área de desenho (ver figuras 2 e 3). O parâmetro “Fatias” representa o número de setores que terá o objeto. Quanto maior o número de fatias, mais suave será o contorno do objeto e maior o tamanho do arquivo gerado.

O eixo Y do objeto é o eixo vertical da tela. Os valores de X e Z dos pontos do objeto gerado são calculados pela distância entre o ponto e o eixo de rotação. As faces são formadas pela união de pontos vizinhos e os pontos vizinhos da próxima fatia. Por isso, as faces serão sempre formadas por quadriláteros.

3.2.4 Sweep Rotacional Completo de Polígono Aberto

Como no *sweep* rotacional completo de polígono fechado, o *sweep* é realizado pela rotação do polígono sobre um eixo representado pela linha vermelha da área de desenho. O sistema de modelagem foi desenvolvido de forma que o início e o fim do polígono gerador não necessite estar sobre o eixo de rotação. O próprio sistema se encarrega de fechar o objeto.

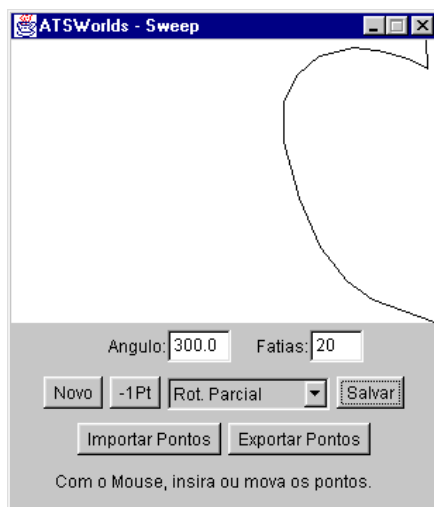
3.2.5 Sweep Rotacional Parcial de Polígono Fechado

Esse tipo de *sweep* é semelhante ao rotacional completo de polígono fechado, mas difere dele pelo fato de não fazer uma volta completa em torno do eixo de rotação. Para realizar este tipo de *sweep*, além do eixo de rotação e do parâmetro “Fatias”, existe também o parâmetro “Angulo”. Este representa qual o ângulo total de revolução que a superfície realizará.

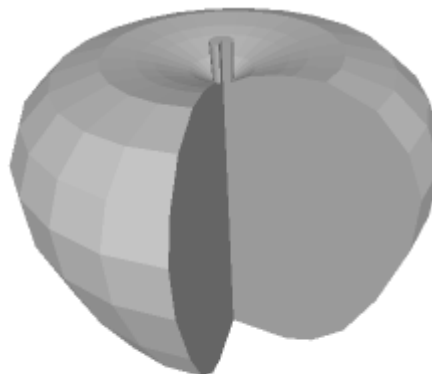
O valor do ângulo deverá estar entre zero e 360 graus, excluindo estes. Para obter um ângulo igual a 360 graus deve-se realizar um *sweep* rotacional completo. Ângulos menores que zero não são necessários, já que todos os ângulos de valores negativos formarão objetos semelhantes aos formados por ângulos positivos.

3.2.6 Sweep Rotacional Parcial de Polígono Aberto

Este tipo de *sweep* é semelhante ao *sweep* rotacional completo de polígono aberto. O início e o final do polígono gerador não necessitam estar sobre o eixo de rotação e o ângulo deve ficar entre zero e 360 graus. A figura a seguir mostra o sistema de modelagens ATSWorlds com um polígono para gerar um objeto pela técnica de *sweep* rotacional parcial de polígono fechado e o objeto gerado a partir deste polígono (vide figura 3).



(a) polígono gerador

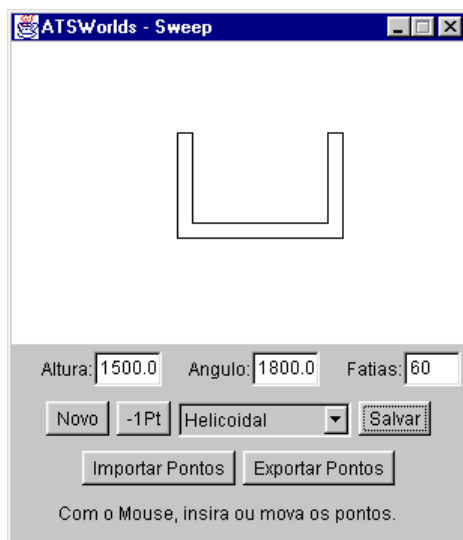


(b) objeto gerado

Figura 3. Sweep Rotacional Parcial de Polígono Aberto.

3.2.7 Sweep Helicoidal

O *sweep* helicoidal é uma combinação dos tipos de *sweep* rotacional parcial de polígono fechado e translacional. Além de incremento de revolução também é feito um incremento de translação no mesmo sentido do eixo de revolução. Neste tipo de *sweep*, o ângulo de rotação pode ser superior a 360 graus, fazendo mais de uma volta em torno do eixo. O ângulo também pode ser negativo, fazendo uma rotação em sentido inverso. A figura 4 mostra o sistema de modelagens ATSWorlds com um polígono para gerar um objeto pela técnica de *sweep* rotacional parcial de polígono fechado e o objeto gerado a partir deste polígono (vide figura 4).



(a) polígono gerador



(b) objeto gerado

Figura 4. Sweep Helicoidal.

4 Conclusão e Perspectivas

Este trabalho que apresentamos demonstra a possibilidade de integração de técnicas de modelagem por sweep com a VRML, através da implementação prática de um software. A interface do sistema de modelagem desenvolvido junto a este trabalho é bastante simples de usar, podendo o software ser usado por pessoas que não necessitem

de uma formação anterior específica sobre o assunto: tanto em relação aos conceitos de modelagem em computação gráfica, quanto em relação ao uso da aplicação em questão, cuja interface é simples e intuitiva.

O software desenvolvido é distribuído livremente e foi implementado em Java, permitindo o seu uso em diferentes plataformas de hardware e software. Portanto, as facilidades de uso da sua interface, a simplicidade com que podemos criar objetos simples ou complexos em VRML, e a possibilidade de seu uso em praticamente qualquer ambiente, configuram esta aplicação como uma excelente ferramenta didática para o ensino de computação gráfica, geometria (sólidos de revolução) e modelagem de sólidos (torneamento e extrusão), e para a construção de mundos virtuais com o uso de ferramentas de realidade virtual. A possibilidade de analisar os modelos gerados (arquivos texto com código VRML – WRL) torna ainda mais interessantes as possibilidades de aplicação do ATSWorlds no ensino.

Apesar deste sistema ser capaz de construir uma grande variedade de objetos, existem ainda algumas melhorias que deverão ser incorporadas ao sistema futuramente, visando aperfeiçoar os trabalhos desenvolvidos até o presente. Algumas destas melhorias são:

- Construção de objetos por outros tipos de sweep, como o Sweep Translacional com Torção e o Sweep Geral, e até mesmo poderão ser adicionadas outras técnicas de modelagem neste sistema, como CSG, por exemplo;
- Definição e integração de atributos de cor, transparência, reflexão e textura nos objetos gerados, melhorando a qualidade visual dos mesmos;
- Definição de transformações de translação, rotação e escala sobre o objeto final, permitindo que o mesmo seja posicionado e inserido em qualquer posição de um mundo virtual. Isto permitirá que se faça a união de vários objetos para formar um outro objeto qualquer, ou para compor uma mesma cena complexa.
- Fazer o editor gráfico de polígonos 2D com janela virtual, possibilitando zoom e panning;
- Importar e exportar os polígonos geradores para outros formatos de arquivos, como por exemplo o formato DXF do AutoCAD;

A ferramenta descrita neste artigo apresenta-se disponível de forma completa e funcional no site <http://www.inf.unisinos.br/~osorio/ATSWorlds.html>, sendo que estudos visando a implementação dos itens citados acima estão sendo estudados.

Bibliografia

- Adams, L. “Visualização e Realidade Virtual”. São Paulo: Makron, 1994.
- Cadoz, Claude. “Realidade Virtual”. São Paulo: Ática, 1997.
- Casacurta, Alexandre. “SIHMOS, sistema híbrido de modelagem de sólidos”. Porto Alegre: UFRGS, 1991. Dissertação de Mestrado, 169 p.
- Diegues, José P. P. “Modelagem Geométrica para Computação Gráfica”. Rio de Janeiro: Achiamé, 1989.
- Duarte, Lucio M.; Silva, Daniela E.; Zanoni, Cícero. “VRML2.0”. Porto Alegre. <http://tinós.pucrs.br/~grv> (30/09/1998).

- Foley, James D.; Dam, Andries V.; Feiner, Steven K.; Hughes, John F. "Computer Graphics: Principles and Practice". 2a. ed. EUA: Addison-Wesley Publishing Company, 1996.
- Gomes, Jonas; Velho, Luiz. "Computação Gráfica". RJ: IMPA, 1998.
- Hermida, Afonso. Ray Tracing: "Aventuras em Computação gráfica & Animação". Rio de Janeiro: Berkeley, 1994.
- Javasoftware. Javasoftware - Sun Corporation. Linguagem de Programação JAVA – Java 1.2 SDK. <http://www.javasoftware.com/> (01/12/1999)
- Mortenson, M. E. "Geometric Modeling". John Wiley & Sons, NY, 1985.
- Silva, André Tavares. "Integração de Técnicas de Modelagem com a VRML". Trabalho de Conclusão de Curso – Curso de Informática. UNISINOS – Centro de Ciências Exatas e Tecnológicas. São Leopoldo, 1999. <http://www.inf.unisinos.br/~osorio/ATSWorlds.html>
- Minho. Universidade do Minho, Departamento de Informática. *Software, Interaction & Multimedia*. "VRML Interactive Tutorial". Portugal. [http:// sim.di.uminho.pt/vrml/](http://sim.di.uminho.pt/vrml/) (18/05/1999).
- Web 3D Consortium. "Web 3D Consortium". EUA. <http://www.vrml.org/> (01/07/1999).