

UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE ENGENHARIA ELÉTRICA  
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



Técnica de Realidade Virtual para manipulação e deformação de  
superfícies NURBS

JOSÉ FLÁVIO DOS REIS

Outubro  
2012

UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE ENGENHARIA ELÉTRICA  
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Técnica de Realidade Virtual para manipulação e  
deformação de superfícies NURBS

Dissertação apresentada por José Flávio dos Reis  
à Universidade Federal de Uberlândia para  
obtenção do título de Mestre em Ciências

Orientador: Prof. Dr. Edgard Lamounier Júnior

Co-orientador: Prof. Dr. Alexandre Cardoso

Outubro  
2012

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA DA UNIVERSIDADE FEDERAL DE UBERLÂNDIA COMO PARTE DOS REQUISITOS PARA OBTENÇÃO DO TÍTULO DE MESTRE EM CIÊNCIAS.

BANCA EXAMINADORA:

---

Prof. Dr. Edgard Lamounier Júnior  
Orientador (UFU)

---

Prof. Dr. Alexandre Cardoso (Faculdade de  
Engenharia Elétrica - UFU)

---

Prof. Dr. Luciano Coutinho (Faculdade de  
Engenharia Elétrica - UFU)

---

Prof. Dr. Celso Volpe (UNISANTA)

Dedico este trabalho, de maneira especial à  
minha esposa Letícia e ao meu filho Gabriel,  
pelo amor e incentivos dispensados comigo.

## AGRADECIMENTOS

A Deus!!!

À minha esposa, por estar sempre ao meu lado, me incentivando, apoiando em todas as fases deste trabalho.

Ao meu filho, que veio para me dar mais força para concluir este trabalho.

Aos meus pais, agradeço todo o apoio e carinho que me deram e por acreditarem em meu potencial, incentivando sempre meus estudos e aos valores que me transmitiram.

Ao professor e Orientador Edgard Afonso Lamounier Júnior, o meu profundo agradecimento por todo o interesse, paciência e ajuda dispensada ao longo deste processo.

Aos colegas do Instituto que de alguma forma contribuíram para a realização deste trabalho.

Aos colegas do Laboratório de Computação Gráfica.

## RESUMO

A Realidade Virtual (RV) lida principalmente, com a criação de ambientes tridimensionais gerados em tempo real e disponibilizados para interação com diferentes dispositivos multissensoriais. Como consequência, uma alta capacidade de processamento e modelagem dos ambientes é requerida para aplicações desta natureza. Estas aplicações, atualmente, estendem-se a vários setores produtivos como: indústria, medicina e saúde, treinamento (capacitação) e, em particular, na Educação. São várias as áreas do saber que exploram a RV como uma ferramenta didática para complementar as técnicas aplicadas ao binômio ensino/aprendizagem num contexto pedagógico. Nos últimos anos, diferentes assuntos relacionados com o ensino de Matemática tem sido objeto de interesse e pesquisa para o uso de técnicas de Realidade Virtual. Os sistemas desenvolvidos têm-se mostrado cada vez mais avançados no uso desta tecnologia.

Uma área da Matemática de grande interesse para indústria é o estudo das superfícies NURBS (*NonUniform Rational B-Spline*). Estas superfícies têm aplicação em várias áreas da indústria e da engenharia – tais como design de automóveis, aeronaves, sapatos, garrafas etc. Entretanto, sistemas computacionais pesquisados que foram elaborados como proposta de ferramenta de estudo e manipulação de NURBS ainda apresentam pequenos níveis de interação com os modelos criados. Estes critérios associados àqueles de navegação e envolvimento – implícitos nos conceitos trabalhados pela RV – providenciariam, certamente, um ambiente mais natural e amigável para o usuário do sistema. Portanto, esta pesquisa surgiu com uma proposta de investigar como as técnicas de RV podem ser usadas para o ensino e até mesmo o treinamento na área de NURBS.

Dessa forma, um sistema computacional foi elaborado e avaliado com potenciais usuários. Os resultados avaliados demonstraram que técnicas de Realidade Virtual oferecem um interessante campo de investigação para proporcionar alguns métodos computacionais que conduzam para uma forma mais dinâmica e motivadora para o estudo de NURBS.

## ABSTRACT

The Virtual Reality (VR) mainly deals with the creation of three-dimensional environments generated in real time and made available for multisensory interaction with different devices. A high processing capacity and modeling environments is required for applications of this nature. These applications currently extend to various productive sectors such as industry, medicine and health, training (capacity building) and, in particular, in the Education. There are several areas of knowledge that explore VR as a teaching tool to complement the techniques applied to the binomial teaching / learning in a pedagogical context. In recent years, various issues related to the teaching of Mathematics has been a subject of interest and research into the use of Virtual Reality techniques. The developed systems have proved a increase advanced in the use of this technology.

One area of the interest for mathematics industry is the study of surfaces NURBS (nonuniform Rational B-Spline). These surfaces have application in many areas of industry and engineer - such as design of automobiles, aircraft, shoes, bottles etc. However, reseachers computer systems were developed as a proposed tool for study and manipulation of NURBS with small levels of interaction and created models. These itens associated with the navigation and involvement, implicit concepts to worked by RV, certainly provide a more natural and user-friendly system. Therefore, this research has come up with a proposal to investigate how VR techniques can be used for teaching and even training in the area of NURBS.

Thus, a computer system was developed and evaluated with potential users. The evaluated results demonstrate that virtual reality techniques offer an interesting field of research to provide some computational methods, that lead more dynamic and motivational to study NURBS.

## LISTA DE PUBLICAÇÕES

A seguir são apresentadas as publicações deste trabalho:

1. REIS, José Flávio; LAMOUNIER, Edgard; CARDOSO, Alexandre. (2011). **Estudo Comparativo de Modeladores de Sólidos para Construção de Ambientes Virtuais ou Aumentados.** In: VIII WORKSHOP DE REALIDADE VIRTUAL E AUMENTADA, Uberaba-MG.
2. REIS, José Flávio; LAMOUNIER, Edgard; CARDOSO, Alexandre. (2012). **Técnica de Realidade Virtual para Manipulação de Superfícies NURBS.** In: X CEEL – Conferência de Estudos em Engenharia Elétrica, Uberlândia-MG.

## LISTA DE FIGURAS

Figura 1: Relação entre os quatro tipos de representações de curva .....	6
Figura 2: Interface interativa do programa Rhinoceros 3D. ....	9
Figura 3: Interface interativa do programa White_dune, software livre sob a licença GNU General Public License. ....	10
Figura 4: Janela principal do editor Unity 3D composta de várias janelas (Views) .....	12
Figura 5: Ambiente Virtual do programa VR Playground. ....	16
Figura 6: Programa Corte e Visualização de Poliedros Não-Convexos. ....	17
Figura 7: Ambiente Virtual do programa Corte e Visualização de Poliedros não-convexos... ..	18
Figura 8: Ambiente virtual do programa Poly-Pro. ....	19
Figura 9: Arquivo VRML, surf.wrl .....	20
Figura 10: Arquivo VRML, surf.wrl .....	20
Figura 11: Arquivo VRML, surf.wrl .....	21
Figura 12: Interface do VRMLPad como parte de um código de AV. ....	23
Figura 13: Interface do Cortona com objetos criados com o código mostrado na Figura 12... ..	24
Figura 14: Figura mostrando a interação dos Pontos de Controle com o Painel. ....	28
Figura 15: Figura mostrando a interação dos Pontos de Controle com o Painel após a ação do código acima. ....	28
Figura 16: Esquema geral da Arquitetura do Sistema desenvolvido. ....	29
Figura 17: Interface Gráfica com o Usuário. ....	30
Figura 18: Superfície plana gerada no <i>software</i> White_Dune. ....	31
Figura 19: Superfície importada do <i>software</i> White-Dune visualizada no Cortona. ....	31
Figura 20: Ambiente Virtual do protótipo criado. ....	32
Figura 21: Ambiente Virtual desenvolvido e seus respectivos itens de funcionamento. ....	33
Figura 22: Interface Gráfica com o Usuário – GUI. ....	34
Figura 23: ambiente virtual após o uso do <i>plug-in</i> do cortona. ....	35
Figura 24: Representação gráfica da esfera amarela circundada, em destaque, por um círculo vermelho. ....	38
Figura 25: O objeto em destaque, circundado por um círculo vermelho, a esfera branca semitransparente, foi criado para marcar os pontos e as interações. ....	39
Figura 26: Função “position” mostra a posição do marcador .....	41
Figura 27: Função “posUsuario” mostra as coordenadas do ponto de controle. ....	43
Figura 28: Página de Abertura do Software NURBS Edit. ....	44

Figura 29: Apresentação dos itens que compõem o Ambiente Virtual de NURBS Edit. ....	45
Figura 30: Enumera a posição de cada ponto de controle .....	46
Figura 31: Sela gerada no software NURBS Edit com alguns pontos de vista. ....	47

## LISTA DE TABELAS

Tabela 1: Comparação das ferramentas para criação de ambiente virtual para NURBS. ....	13
Tabela 2: Comparação entre os softwares criados para auxiliar o ensino/aprendizagem. ....	22
Tabela 3: Comparação entre os softwares criados para auxiliar o ensino/aprendizagem incluindo o NURBS Edit. ....	58

## SUMÁRIO

CAPÍTULO 1 – Introdução .....	1
1.1 Motivação .....	1
1.2 Objetivos.....	2
1.3 Justificativa.....	3
1.4 Estrutura da Dissertação .....	3
CAPÍTULO 2 – FUNDAMENTOS .....	5
2.1 Introdução.....	5
2.2 NURBS – NonUniform Rational B-Spline .....	5
2.3 Sistemas de apoio à construção de NURBS .....	8
2.3.1 Rhinoceros .....	8
2.3.2 White_Dune.....	10
2.3.3 UNITY 3D.....	11
2.4 Comparação dos softwares para edição de NURBS.....	13
CAPÍTULO 3 – TRABALHOS RELACIONADOS .....	15
3.1 Introdução.....	15
3.2 VR Playground .....	15
3.3 Corte e Visualização de Poliedros não-Convexos.....	16
3.4 Poly Pro .....	18
3.5 Surf .....	19
3.6 Comparação dos Trabalhos Relacionados.....	21
CAPÍTULO 4 – ARQUITETURA DO SISTEMA .....	23
4.1 Introdução.....	23
4.2 – VRML.....	23
4.3 – JavaScript.....	26
4.4 Arquitetura do Sistema .....	29
4.5 Ambiente Virtual .....	30
4.6 Interface – GUI.....	33
4.7 Elementos Virtuais e Painel de Controle.....	35
4.8 Considerações Finais .....	36
CAPÍTULO 5 – DETALHES DE CONSTRUÇÃO DO SISTEMA .....	37
5.1 Introdução.....	37

5.2 Implementação do Ambiente Virtual .....	37
5.3 Implementação dos Deslocamentos da Superfície .....	39
5.4 Considerações Finais .....	43
CAPÍTULO 6 – AVALIAÇÃO DO SISTEMA E RESULTADOS .....	44
6.1 Introdução .....	44
6.2 Utilizações do Sistema.....	44
6.3 Estudo de Caso .....	46
6.4 Avaliação do Sistema .....	48
6.4.1 Questões sobre a Funcionalidade.....	48
6.4.2 Questões sobre a Usabilidade .....	49
6.4.3 Questões sobre a Confiabilidade .....	51
6.4.4 Questões sobre a Eficiência .....	52
6.4.5 Questões sobre o Sistema .....	54
6.4.6 Questões sobre Avaliação de Desempenho .....	55
6.5 Comparação do software desenvolvido, NURBS Edit com outros programas. ....	58
6.6 Limitações do Sistema.....	59
6.7 Considerações Finais .....	59
CAPÍTULO 7 – CONCLUSÕES E TRABALHOS FUTUROS.....	60
7.1 Conclusões.....	60
7.2 Trabalhos Futuros .....	61
REFERÊNCIAS .....	62
ANEXO I.....	66
APÊNDICE I.....	76

# **CAPÍTULO 1 – Introdução**

## **1.1 Motivação**

De maneira geral, a habilidade de comunicar, visualmente, em áreas de interesse sempre fez parte da vida do ser humano permitindo-o expressar-se ao longo do tempo, desde desenhos primitivos, figuras e pinturas até o cinema e outras expressões artísticas. O uso do computador fortaleceu tais formas de expressão. Não demorou muito para que essas tecnologias convergissem e, rompendo a barreira da tela do monitor, passassem a gerar ambientes tridimensionais interativos em tempo real [1].

Assim, a Realidade Virtual permite ao usuário retratar e interagir com situações imaginárias, como os cenários de ficção, envolvendo objetos virtuais estáticos e em movimento. Permite também reproduzir com fidelidade, ambientes da vida real como casas, universidades e bancos, de forma que o usuário possa entrar nesses ambientes e interagir com seus recursos de forma natural. É possível também o uso das mãos com o auxílio de aparatos tecnológicos, como luva, e eventualmente gestos ou comandos de voz [2].

Tori et al. definiram a Realidade Virtual como sendo uma interface avançada para aplicações computacionais, que permite ao usuário a movimentação e interação em tempo real, em um ambiente tridimensional (3D), podendo-se fazer uso de dispositivos multissensoriais, para atuação ou feedback [1].

A Realidade Virtual, de acordo com Cardoso e Lamounier [3], gera experiências de utilização de sistemas que têm sido desenvolvidas e aplicadas nos mais diversos campos, desde aplicações médicas, arquitetura e projetos, aplicações científicas, em artes, na educação, na visualização e no controle da informação, entretenimento e comunicações.

Como uma das aplicações da RV esta relacionada à Educação, é possível usar ferramentas computacionais para melhorar o desempenho do ensino e desenvolver no aluno o senso crítico e a capacidade de observação. Segundo Roussou [4], a Realidade Virtual inclui vários níveis de recursos interativos educacionais, em que o usuário tem a sensação de estar imerso no ambiente, vivenciando e modificando ações num contexto específico de ensino no ambiente virtual.

Assim, acredita-se que se pode utilizar a RV para desenvolver ambientes virtuais para o ensino das NURBS que são modelos matemáticos usados regularmente em programas gráficos para criar e representar curvas e superfícies. Estas estruturas são de grande importância na elaboração de desenhos por computador (*Computer Aided Design, CAD*), no desenvolvimento de um novo produto e objetos para aplicações científicas, na indústria ou na modelagem de peças para automóveis, aeronaves, navios, entre outros. É importante ressaltar que na área educacional, há certa ausência de ambientes virtuais desenvolvidos especificamente para o ensino dessas funções.

## 1.2 Objetivos

O objetivo deste trabalho é investigar técnicas computacionais que suportem a visualização e manipulação de NURBS, possibilitando ao usuário realizar deformações nas superfícies geradas, de uma forma mais natural e intuitiva. Para atingir os objetivos desejados, as seguintes metas foram definidas:

- Estudar sistemas computacionais relacionados à Realidade Virtual, desenvolvidos para o processo de ensino/aprendizagem, afim de, identificar vantagens e desvantagens;
- Estudar algoritmos que possibilitem a modelagem e manipulação de NURBS em um ambiente virtual;
- Projetar e construir um sistema protótipo que suporte os algoritmos estudados;
- Analisar as técnicas de interação no ambiente virtual construído;
- Avaliar o sistema protótipo, junto aos usuários, professores e alunos do ensino superior;
- Avaliar e discutir os resultados das avaliações do sistema protótipo testado para que esse possa nortear trabalhos futuros.

### **1.3 Justificativa**

A Realidade Virtual é uma tecnologia de interface avançada que possibilita ao usuário não somente usar o sistema de software, como também perceber-se dentro do ambiente tridimensional gerado por computador. Assim, o usuário pode explorar e mesmo modificar o ambiente virtual, o que lhe é possibilitado através de técnicas de navegação, interação e imersão, segundo Braga [5].

Assim, de acordo com Tori et al. [6], o uso da Realidade Virtual justifica-se pelo amplo potencial em proporcionar educação e aprendizagem, como resultado do processo de exploração, descoberta, observação e construção de uma nova visão do conhecimento, proporcionando ao aluno a oportunidade de melhorar sua percepção do objeto em estudo. Portanto, essa tecnologia tem condições de contribuir no processo cognitivo do aprendiz, proporcionando não apenas a teoria, mas também a experimentação prática do conteúdo em estudo.

Entretanto, dos ambientes virtuais estudados para ensino e aprendizagem de diferentes tópicos, não foi identificado algum específico que se comportasse como uma ferramenta de introdução e uso de NURBS – até o momento da elaboração desta dissertação. Daí, a motivação deste trabalho em avaliar o potencial de um sistema de RV para este fim.

### **1.4 Estrutura da Dissertação**

O trabalho foi estruturado em sete capítulos. O Capítulo 1, é a Introdução, a qual se apresenta.

O Capítulo 2, trata-se dos Fundamentos, ou seja, programas utilizados para desenvolver os diversos tipos de Ambientes Virtuais utilizados para a criação de objetos relacionados à simulação de ambientes reais.

O Capítulo 3, direciona e relata os trabalhos relacionados ao objeto de estudo deste trabalho, enfatizando-se os principais trabalhos correlatos à criação de Ambientes Virtuais como ferramentas educacionais.

O Capítulo 4 descreve a arquitetura do sistema proposto, no qual se tem a visualização geral da criação do protótipo, geração das NURBS e o ambiente virtual para a manipulação e interação com o objeto criado. A descrição metodológica do sistema e a descrição dos algoritmos matemáticos utilizados no desenvolvimento do protótipo.

O Capítulo 5, refere-se à implementação do sistema, ou seja, experimentação e teste do protótipo com alunos e professores.

O Capítulo 6, mostra os resultados e discussões relevantes sobre o funcionamento e limitações do protótipo em termos de utilização e funcionalidade.

O Capítulo 7 apresenta as conclusões sobre o programa criado, bem como constam sugestões para futuros trabalhos.

Por fim, houve a apresentação e citação das referências utilizadas que serviram para nortear a pesquisa e desenvolvimento do trabalho.

## **CAPÍTULO 2 – FUNDAMENTOS**

### **2.1 Introdução**

Neste capítulo, são apresentados a definição de NURBS e programas utilizados para desenvolver ambientes virtuais baseados nas mesmas. As técnicas utilizadas para a criação de objetos e sua inserção no ambiente virtual são discutidas. Mas, principalmente, enfatizam-se as técnicas abordadas sobre Realidade Virtual (RV) desenvolvidas e aplicadas nos diversos campos do ensino para aproximar e facilitar o cotidiano real da aprendizagem.

### **2.2 NURBS – NonUniform Rational B-Spline**

As NURBS surgiram a partir dos trabalhos pioneiros de Pierre Bézier sobre curvas e superfícies, entre os anos de 1960 a 1970. Segundo, Félix et al. [7], as NURBS são curvas que podem ou não passar pelo seu ponto de controle inicial e final, podendo-se aproximar dos demais pontos de controle, mas sobre tudo, matematicamente, é descrita pela diferença entre um modelo de curva por polinômios, cujo controle é global e de outro descrito pelo quociente entre polinômios, de controle local. Descreve também que, as NURBS são curvas contínuas que têm direções e magnitudes equivalentes, em que os ajustes de controle ou deformações na superfície da curva, em um determinado trecho não se propagam em toda curva o que permite a representação exata de formas quadráticas.

Tal como apontado por Rogers [8], é de extrema relevância a capacidade das NURBS para representar curvas cônicas e superfícies com maior precisão. Num sistema de modelagem NURBS, pode-se usar uma única representação interna para uma variada gama de superfícies e curvas, de linhas retas e superfícies planas em círculos e esferas precisas, bem como complicados cortes para esculpir superfícies. Além disso, permite que elementos NURBS sejam facilmente colocados dentro de uma superfície ampla. Esta característica única de NURBS é a chave para o desenvolvimento de um sistema de modelagem sólida, seja para desenho assistido por computador: de automóveis, aeronaves, embarcações, sapatos, garrafas

de xampus, chuveiro ou de personagens animados nas produções mais recentes de Hollywood ou jogos de computador. Como resultado, as NURBS são usadas por designs e pela comunidade de computação gráfica.

Normalmente, os desenhistas e os usuários de gráficos assistidos por computador, por exemplo, os educadores, não compreendem que no funcionamento de um programa há uma série de ferramentas sendo utilizadas cujo suporte e modelagem matemática é fundamental para que obtenham resultados no uso do programa. Portanto, um sistema que suporte utilização e desenho de curvas como as NURBS, exige um bom equacionamento descritivo. No ANEXO I deste trabalho há uma sucinta descrição matemática a respeito das técnicas e critérios estudados para a geração de curvas ou superfícies NURBS. Entretanto, para melhor compreensão, a Figura 1 apresenta um panorama geral do estudo sobre NURBS.

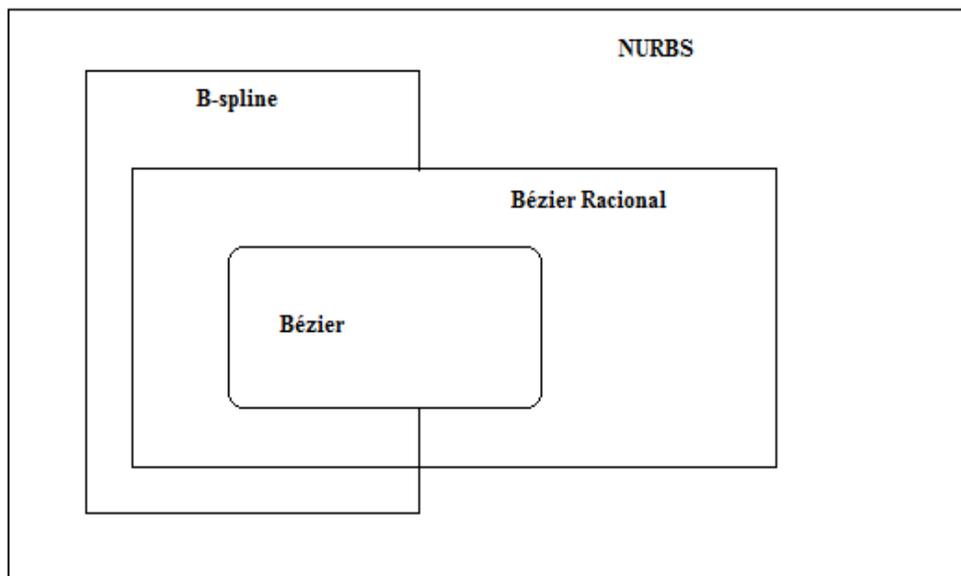


Figura 1: Relação entre os quatro tipos de representações de curva

FONTE: Lavoie, 2001 apud Minetto [9].

De acordo com Piegl e Tiller [10], tem-se 14 propriedades básicas das superfícies NURBS para serem geradas e modificadas que devem atender o seguinte:

- 1<sup>a</sup> Função positiva definida:  $R_{i,j}(u,v) \geq 0, \forall i,j,u,v$  e  $\forall (u,v) \in [0,1] \times [0,1]$ ;
- 2<sup>a</sup> Partição da unidade:  $\sum_{i=0}^n \sum_{j=0}^m R_{i,j}(u,v) = 1 \quad \forall (u,v) \in [0,1] \times [0,1]$ ;
- 3<sup>a</sup> Suporte local:  $R_{i,j}(u,v) = 0$  para  $(u,v) \notin [u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$ ;

- 4<sup>a</sup> Dado o período  $u_{i_0}, u_{i_0+p+1} \times v_{j_0}, v_{j_0+q+1}$ , no máximo  $(p + 1) (q + 1)$  funções de base são não nulas, em particular  $R_{i,j}(u,v)$  para  $(i,j) \in [i_0-p, i_0] \times [j_0-q, j_0]$ ;
- 5<sup>a</sup> Toda função possui no mínimo dois pontos extremos, ou seja, possui ao menos um mínimo e um máximo; A função NURBS  $R_{i,j}(u,v)$  para  $p > 0$  e  $q > 0$  possui um mínimo e um máximo no intervalo  $(u,v) \in [0,1] \times [0,1]$ ;
- 6<sup>a</sup>  $R_{0,0} = R_{n,0} = R_{0,m} = R_{n,m} = 1$ ;
- 7<sup>a</sup> Todas as derivadas parciais de  $R_{i,j}(u,v)$  existem no interior do período, são funções racionais de denominador não nulo. Sobre o nó,  $R_{i,j}(u,v)$  é  $C^{p-k}$  e  $C^{q-k}$  contínuo, na direção de  $u$  e  $v$ , respectivamente, onde  $k$  é a multiplicidade do nó;
- 8<sup>a</sup> Se  $w_{i,j} = a$ ,  $\square(i,j)$ , então  $R_{i,j}(u,v) = N_{i,p}(u)N_{j,q}(v)$  para  $\square(i,j)$
- 9<sup>a</sup> Interpolação dos pontos de controle extremos:  $S(0,0) = P_{0,0}$ ,  $S(1,0) = P_{1,0}$ ,  $S(0,1) = P_{0,1}$  e  $S(1,1) = P_{1,1}$ ;
- 10<sup>a</sup> Transformações lineares: transformações tais como rotação, translação, escala e cisalhamento aplicados à superfície são realizadas aplicando tal transformação ao polígono de controle;
- 11<sup>a</sup> Região convexa: se  $w_{i,j} \geq 0$  para  $\square(i,j)$ , a superfície esta contida na região convexa formada pelo polígono de controle; de fato, se  $(u,v) \in [u_{i_0}, u_{i_0+1}] \times [v_{j_0}, v_{j_0+1}]$ , então  $S(u,v)$  estará contida na região formada pelos pontos de controle  $P_{i,j}, \square i \in [i_0 - p, i_0] \square j \in [j_0 - q, j_0]$ ;
- 12<sup>a</sup> Esquema de modificação local: movendo  $P_{i,j}$  ou  $w_{i,j}$ , a superfície  $S(u,v)$  modifica sua forma somente no intervalo  $[u_i, u_{i+p+1}] \times [v_j, v_{j+q+1}]$ ;
- 13<sup>a</sup> Bézier, B-spline não racionais e Bézier racionais são um caso particular de superfícies NURBS;
- 14<sup>a</sup> A continuidade e a diferenciabilidade da superfície  $S(u,v)$  seguem das funções de base; em particular  $S(u,v)$  é infinitamente diferenciável no interior do intervalo de nós, e é  $C^{p-k}$  e  $C^{q-k}$  contínua na direção  $u$  e  $v$ , respectivamente; onde  $k$  é a multiplicidade do nó.

Com estas 14 propriedades, podem-se controlar os objetos criados e modificar a forma de um objeto, através dos pontos de controle em um intervalo determinado, aplicando-se a 12<sup>o</sup> propriedade das superfícies NURBS.

Neste trabalho a propriedade de maior relevância é a 12ª propriedade, a qual utiliza-se para modificar o objeto através dos parâmetros do ponto de controle, alterando seus valores e portanto, o objeto sofre modificações em sua estrutura.

## 2.3 Sistemas de apoio à construção de NURBS

### 2.3.1 Rhinoceros

O programa Rhinoceros ou Rhino, como é mais conhecido, é um software de modelagem NURBS – *Non Uniform Rational B-Spline*, que normalmente é utilizado por profissionais da área de design, arquitetura e engenharias, para desenho de peças e equipamentos. As superfícies NURBS são muito empregadas para simular formas orgânicas perfeitas, como o corpo humano, incluindo curvas, superfícies: abertas, fechadas e com buracos, além de sólidos, modelos fechados formados por uma ou mais superfícies unidas por suas arestas. O software proporciona a possibilidade de manipular as superfícies diretamente, admitindo que modelos sólidos possam ser divididos, editados e unidos novamente. Além disso, possui recursos básicos de criação, conversão e edição de malhas (*mesh*) para integração com outros softwares do mercado [11].

O sistema ainda suporta as tradicionais tecnologias CAD - *Computer-Aided Design* e CAM - *Computer-Aided Manufacturing* em que associa precisão e flexibilidade na modelagem de curvas e superfícies, para desenvolver objetos de formas simples ou complexos.

Há a versão Rhinoceros 3D que foi desenvolvida pela Robert McNeel & Associates para ser utilizado no sistema operacional Windows. O programa foi gerado como um *plug-in* para o AutoCAD, da Autodesk, de acordo com Leake [12]. Posteriormente, mais desenvolvido, o projeto se tornou um aplicativo independente.

A interface do Rhino pode ser visualizada na Figura 2, a qual apresenta as partes descritas e assinaladas na figura.

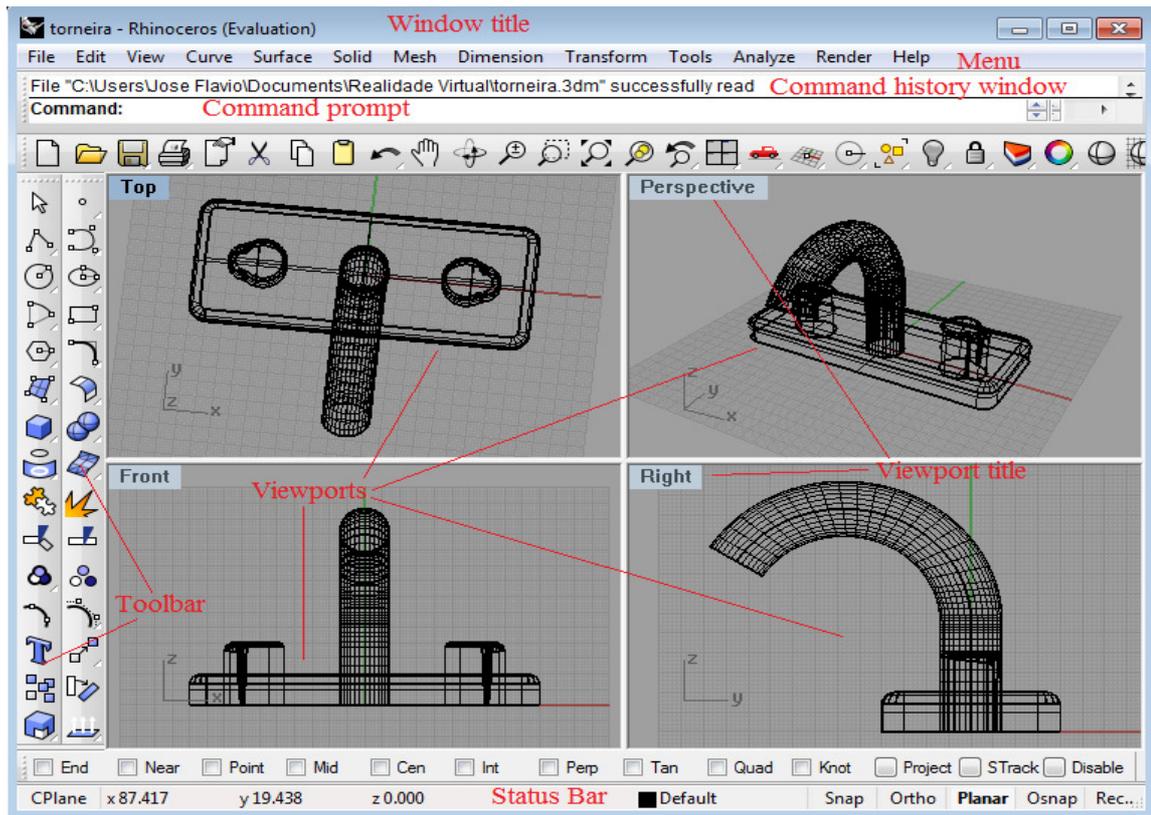


Figura 2: Interface interativa do programa Rhinoceros 3D.

FONTE: Software Rhinoceros - Rhino window parts [11].

- *Menu* – barra comum a todo sistema operacional Windows, onde podemos acessar comandos e opções de configuração do *software*;
- *Command history window* – lista os comandos e informações digitadas;
- *Viewport title* – nomes das vistas;
- *Status bar* – Sistema de coordenadas;
- *Viewports* – Mostra o ambiente de trabalho, incluindo o objeto de exibição, título da janela de exibição, fundo, plano de construção em grade;
- *Toolbar* – opções para criação, edição e análise de curvas, superfícies e polígonos;
- *Window title* - Mostra o nome do modelo atual do arquivo.
- *Command Prompty* - Lista os comandos e informações digitadas;

Para o desenvolvimento deste projeto o Rhino seria de grande utilidade, pois apresenta vantagens em relação aos demais *softwares* devido a sua funcionalidade e diversidade de comandos de fácil assimilação, garantindo ao usuário uma rápida aprendizagem do programa, além de apresentar boa integração com outros softwares, em especial com o VRML (*Virtual Reality Modelling Language*), que é uma linguagem de modelagem muito conhecida para construção de ambientes virtuais. Entretanto, um empecilho verificado na utilização do

Rhinoceros para o desenvolvimento desta pesquisa foi o seu custo de aquisição e também a manutenção de sua licença, por ser um *software* proprietário.

### 2.3.2 White\_Dune

O programa White\_Dune é um software com poucos recursos, que gera NURBS simples. É compatível com o sistema X3D/VRML97, e foi desenvolvido como ferramenta para o Unix/Linux/MacOSX e MS Windows, por Stephen F. White sobre o programa "duna", portanto, White\_Dune. É um software livre sob a licença da GNU, *General Public License* (GPL) [13].

Também possui um alto índice de usabilidade – facilidade de manipulação de suas funcionalidades. Esta é uma ferramenta que pode ler arquivos do tipo VRML97/X3D, além de mostrar e deixar o usuário modificar os objetos na tela, como: forma, tamanho, altura e posição.

Na Figura 3, observa-se a interface de exibição do White\_dune, que apresenta as seguintes partes.

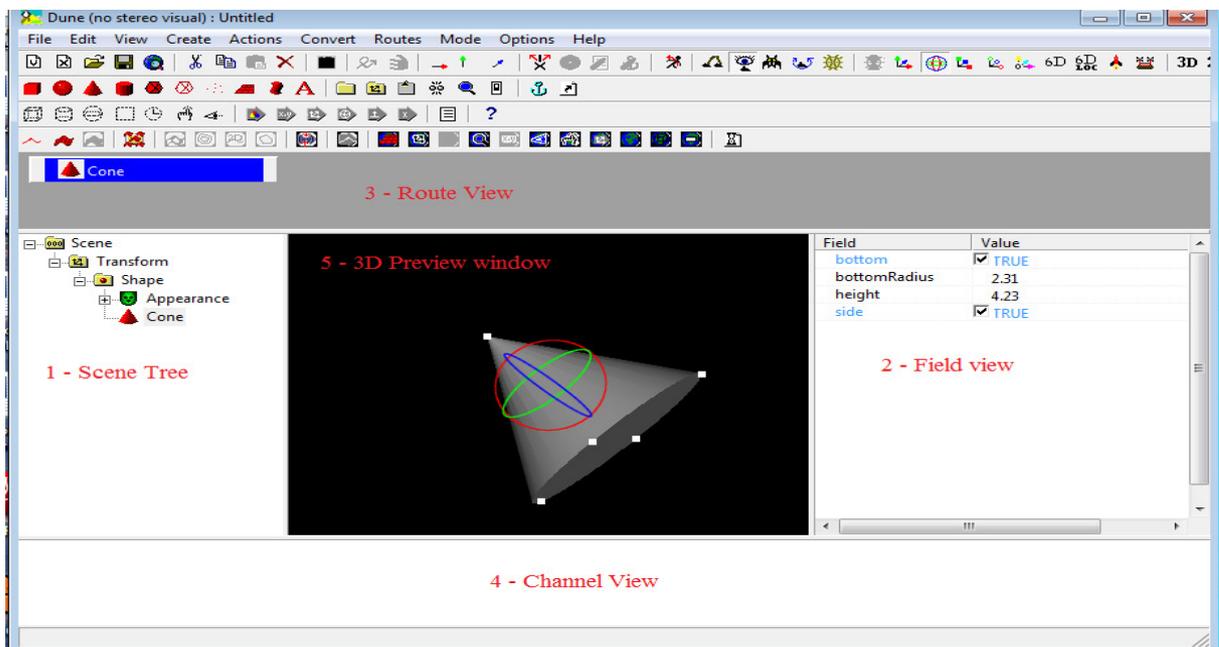


Figura 3: Interface interativa do programa White\_dune, software livre sob a licença GNU General Public License.

Fonte: Software White-Dune

- 1) Scene Tree – esta janela mostra o gráfico de cena, sua estrutura e como os nós são organizados em um arquivo VRML/X3D;
- 2) Field View – esta janela mostra os valores dos campos do nó selecionado, em um arquivo VRML/X3D;
- 3) Route View – esta janela mostra as rotas da troca de informações dos diferentes nós de arquivos VRML/X3D;
- 4) Channel View – esta janela é usada para nós interpolator;
- 5) 3D Preview window – esta janela mostra uma pré visualização do objeto da cena em um arquivo VRML/X3D [14].

O sistema é composto por uma tela do ambiente, onde são mostrados na parte superior alguns ícones. No centro da tela tem-se a visualização dos objetos, onde o usuário tem a opção de interagir com os mesmos, através dos pontos de controle, podendo-se aumentar, diminuir a altura, o raio e visualizar o objeto de vários ângulos.

Entretanto, existem algumas desvantagens, como por exemplo, o *White\_Dune* apenas gera NURBS simples que conseqüentemente se torna um editor de VRML97 e não um navegador VRML97. Além disso, não suporta sons, nem scripts ou integração com os sensores do mouse; torna visível quase todos os nós VRML97, exceto *MovieTexture* e *FontStyle*; e alguns campos, como o *IndexedFaceSet.convex* ainda não estão implementadas.

O *White-Dune* apresenta algumas vantagens em relação a outros sistemas avançados de geração de NURBS, pois é um *software* livre, que exporta objetos NURBS para VRML. Além disso, apresenta facilidades de uso por parte dos usuários, na qual, a maioria de suas funções é apresentada em ícones, suporta animação e interação em tempo real que por consequência viabiliza a aprendizagem.

### 2.3.3 UNITY 3D

O software Unity 3D simplifica o processo de criação de jogos 3D ou outras aplicações através de uma ferramenta de edição fácil de usar, que permite inserir elementos pré-fabricados e a possibilidade de combiná-los com outros recursos de forma a criar ambientes de jogos, segundo Bento [15].

De acordo com Neto [16], o Unity 3D foi desenvolvido pela Unity Technologies, e conta com um conjunto de ferramentas que permitem criar conteúdos interativos em três

dimensões, que podem ser jogos, conteúdos educativos ou simulações em áreas da ciência. O Unity 3D é um motor de jogo (*game engine*) proprietário, cuja versão gratuita é limitada.

Segundo Araujo [17], o Unity 3D suporta as linguagens, C#, JavaScript e Boo (uma variante de Python). Sua versão gratuita está disponível nas seguintes plataformas: Windows, Mac e Web player.

De acordo com Viana [18], o Unity 3D suporta uma série de formatos de arquivos multimídia, entre eles, mp3, jpeg, gif, mov, avi, wav e TTF. Ainda importa modelos 3D nos formatos nativos de alguns programas de modelagem, tais como: Blender, Maya, 3D Studio Max e Cheetah 3D.



Figura 4: Janela principal do editor Unity 3D composta de várias janelas (Views)

fonte: <http://unity3d.com/support/documentation/Manual/Learning%20the%20Interface.html>

O ambiente de desenvolvimento Unity 3D, (Figura 4) é composto por várias janelas que facilitam na elaboração dos jogos e seus aplicativos são:

- *Project* – nesta janela são armazenados todos os ativos que compõem o jogo, como cenas, *scripts*, modelos 3D, texturas, arquivos de áudio e *prefabs* - objetos personalizados que irão compor grande parte do seu jogo;
- *Hierarchy* – a hierarquia contém todos os *GameObject* na cena. Alguns destes são instâncias diretas de arquivos ativos, como modelos 3D, e outros são casos de *Prefabs*.
- *Toolbar* – a barra de ferramentas é composta por cinco controles básicos.
- *Scene* – é a forma principal de manipulação dos objetos da cena. Através do *scene*, pode-se selecionar a posição dos ambientes, do jogador, da câmera, dentre outros

*GameObjects*. Manobrar e manipular objetos dentro da cena são algumas das funções mais importantes do Unity.

- *Inspector* – Jogos em Unity são compostos por vários *GameObjects* que contêm malhas, scripts, sons ou outros elementos gráficos como *Lights*. O Inspector exhibe informações detalhadas sobre os *GameObject* selecionados, incluindo todos os componentes ligados e suas propriedades. No inspector pode-se, modificar a funcionalidade de *GameObjects* em sua cena, como se pode observar na Figura 4.

## 2.4 Comparação dos softwares para edição de NURBS

Pode-se verificar que os *softwares* Rhinoceros, White-dune e Unity 3D possuem mecanismos diferentes de funcionamento e, portanto, possuem vantagens e desvantagens no que se refere à criação de ambientes virtuais para manipulação de NURBS.

Após o estudo dos sistemas, foi possível construir a Tabela 1, que apresenta uma comparação destas ferramentas.

Tabela 1: Comparação das ferramentas para criação de ambiente virtual para NURBS.

<b>Crítérios Avaliados</b>	<b>Rhinoceros</b>	<b>White-Dune</b>	<b>Unity 3D</b>
Tecnologia NURBS	Sim	Sim	Não
Interação com outros <i>Softwares</i> (Exporta para VRML)	Sim	Sim	Não
Comandos de fácil assimilação	Sim	Sim	Sim
<i>Software</i> livre / custo	Não	Sim	Não
Interação com os sólidos	Bom	Bom	Bom
Plataformas	Microsoft Windows	Unix/Linux/MacOS X e MS Windows	Microsoft Windows, Mac OS X, Android SDK
Línguas disponíveis no <i>Software</i>	Inglês	Inglês, Alemão e Italiano	Inglês
Utilização	<i>Software</i> de modelagem NURBS	Editor gráfico VRML97, (NURBS) modelador 3D e ferramenta de animação em desenvolvimento	Animação, criação de jogos de vídeo 3D
Documentação (Tutoriais)	Bom	Bom	Bom

Verificou-se no decorrer do estudo que tanto o Rhinoceros quanto o software White\_Dune utilizam-se de tecnologia NURBS para modelar objetos, sendo que o software

White\_Dune dispõe de poucos recursos, com geração de NURBS mais simples. Já o software Unity 3D não possui em sua tecnologia possibilidade de geração de superfícies NURBS. Entretanto, é muito utilizado para a criação de ambientes virtuais e conta com a facilidade de importar objetos de outros softwares, para compor a cena.

Tanto o software Rhinoceros e Unity 3D são proprietários e podem ser testados por um determinado período, gratuitamente. Já o White\_Dune é um programa livre. Finalmente, verificou-se que para a construção de ambientes virtuais, os objetos de interesse não demandam de uma grande riqueza de manipulação e deformação de suas formas. Entretanto, quando os requisitos são a criação de objetos, com grande riqueza de detalhes, para o desenvolvimento dos ambientes virtuais, é recomendável que o sistema Rhinoceros seja adotado.

Deste modo, como este projeto se propõe a desenvolver um protótipo de um ambiente virtual em que os usuários, possam interagir com uma superfície NURBS, realizando deformações na superfície, alterando seus pontos de controle, de forma livre, utilizando-se do mouse para esta interação. Utilizou-se do software White-Dune, para gerar as superfícies NURBS. Outro requisito considerado foi o fato do software ser *open source*. O próximo capítulo apresenta alguns sistemas que foram construídos para suportar o ensino de Matemática em ambientes de Realidade Virtual, como intuito de avaliar sua potencialidade para suportar NURBS.

## **CAPÍTULO 3 – TRABALHOS RELACIONADOS**

### **3.1 Introdução**

O objetivo deste capítulo é apresentar uma revisão de vários programas computacionais, relacionados à ambientes virtuais na área educacional. Apresenta-se na revisão de trabalhos relacionados os conteúdos desenvolvidos nos ambientes, as formas de interação, atentando-se para as vantagens, desvantagens no campo da Realidade Virtual.

### **3.2 VR Playground**

O VR Playground é um sistema desenvolvido pela University College London (UCL), que projetou e implementou, através da criação de um Parque Virtual (AV) para ensinar crianças do ensino fundamental alguns conceitos matemáticos, na qual estas completam um conjunto de tarefas construtivistas abordando problemas de frações matemáticas.

De acordo com Roussou [4], o sistema foi projetado para incorporar a tarefa de aprendizagem, incluindo vários níveis de recursos interativos. E ainda, optou-se por frações matemáticas, por estas gerarem exercícios abstratos, de difícil concepção e aprendizagem, que justifica a representação através de uma simulação de uma situação real de mundo, através da realidade virtual.

No sistema desenvolvido, consta-se de seis elementos principais (Balanços, barras de macaco, um escorrega, um carrossel, um túnel de rastreamento e uma caixa de areia), codificados por cores e representados por blocos.

O sistema apresenta uma área, em que, o conjunto de blocos, representando cada elemento é inicialmente incorreto (Ou muito grande ou muito pequeno) e deve ser modificado por regras que exigem cálculos de frações. Por exemplo, as oscilações inicialmente cobrem uma área de  $3 \times 4$  isto é, 12 blocos de três linhas e quatro colunas. O cenário requer o aumento da área, comparando duas frações ( $1/3$  e  $1/4$ ) e escolher o número representando a

maior quantidade. Neste caso, o jogador deve escolher  $1/3$ , o que resulta em quatro blocos, que são adicionados sobre as peças que devem ser cobertas.

O sistema fornece retorno visual e auditivo, direto e indiretamente para que às crianças respondam as atividades. Uma coruja virtual apresenta o cenário e os objetivos do ambiente para o usuário (ver Figura 5). Os usuários são orientados por pássaros coloridos que flutuam sobre o ambiente virtual e comunicam com o usuário ao ser clicado. O software apresenta pouca interação que se dá através de um dispositivo (uma "varinha mágica") que inclui um joystick para se navegar no ambiente e ainda três botões coloridos para escolher e colocar os blocos, em determinadas áreas do ambiente virtual.

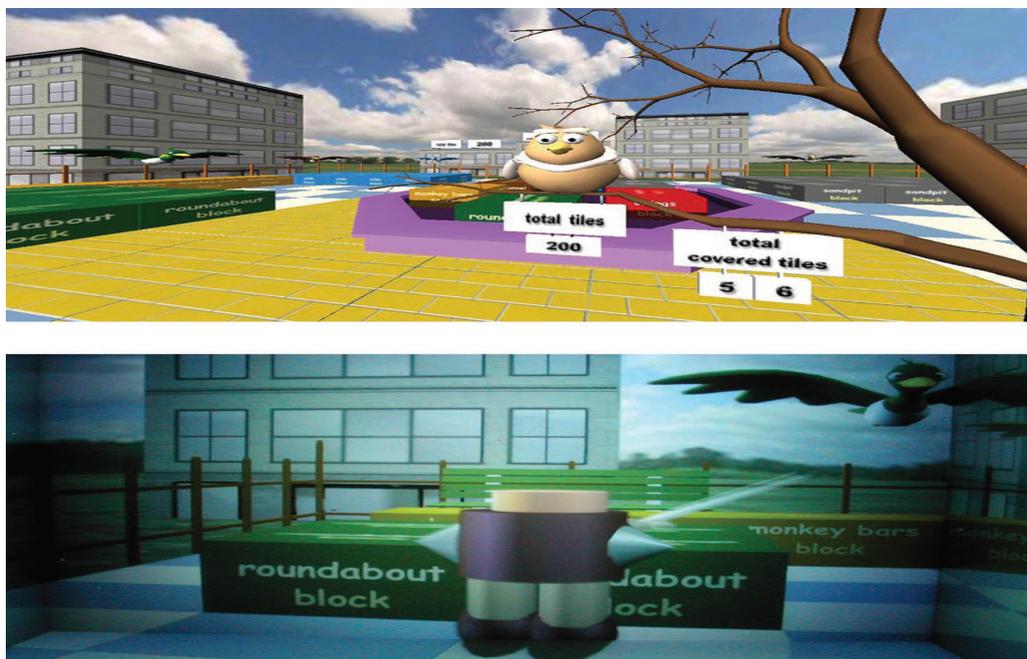


Figura 5: Ambiente Virtual do programa VR Playground.

Fonte: *A VR Playground for Learning Abstract Mathematics Concepts*

O sistema apresenta limitações como: os ambientes virtuais são previamente definidos, portanto os passos do aprendizado já estão previamente determinados, desestimulando a criatividade e a falta de interação com os objetos do ambiente virtual.

### 3.3 Corte e Visualização de Poliedros não-Convexos

O programa Corte e Visualização de Poliedros não-Convexos, Figura 6, trabalha com links, que direciona o usuário para determinado objeto, localizado em um ambiente virtual,

neste caso o *plug-in* Cosmo Player, em que o usuário visualiza o poliedro e controla o sistema pelo painel de controle mostrado na Figura 7.



Figura 6: Programa Corte e Visualização de Poliedros Não-Convexos.

Fonte: Dissertação de Mestrado de Maria Emília da Silva

No ambiente virtual, o usuário, pode interagir com os poliedros, sendo que o usuário, primeiramente, escolhe aleatoriamente três pontos não-colineares, em seguida clica no botão plano, para gerar um plano. Em seguida clica-se no botão corte, onde estava o plano é gerado um corte no poliedro, separando as partes cortadas. E assim é, produzido novos poliedros no ambiente virtual, em que o usuário terá a opção de interagir com os mesmos, podendo, arrastá-los pelo ambiente.

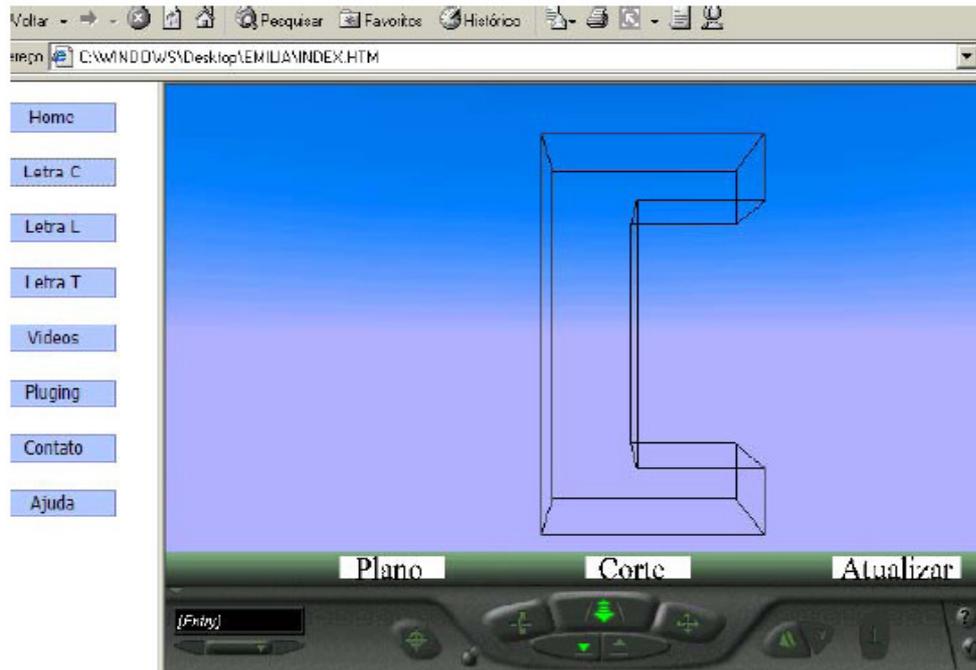


Figura 7: Ambiente Virtual do programa Corte e Visualização de Poliedros não-convexos.

Fonte: Dissertação de Mestrado de Maria Emília da Silva

O sistema mostra figuras geométricas no ambiente virtual, mas há poucas opções de interação, pois o usuário fica limitado em apenas realizar cortes na superfície. Sem opções de interagir mais com o objeto.

### 3.4 Poly Pro

O Poly Pro é um programa de investigação de formas poliédricas, como mostra a figura 8. Esta exhibe formas poliédricas de três maneiras principais: como uma imagem em três dimensões, uma figura achatada em duas dimensões e uma incorporação topológica no plano.

As imagens em três dimensões podem ser estáticas ou giradas de forma intermitente ou ainda sendo dobradas e desdobradas. Os modelos podem ser visualizados em uma janela, em que podemos girar tais objetos, e ainda é visualizado através da janela o objeto se desdobrando, ficando na forma de superfície plana.

O sistema apresenta limitações, pois se tem poucas interações com o ambiente, não permitindo a interação direta do usuário com os objetos criados, podendo apenas girar as figuras ou o próprio sistema que ao clique do mouse desdobra a figura, no ambiente.

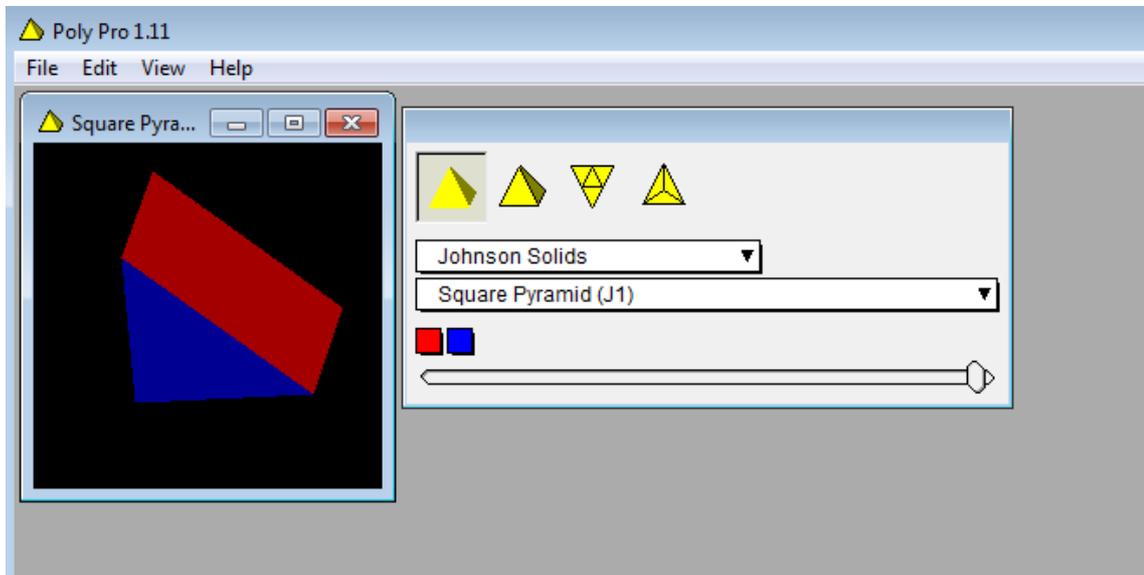


Figura 8: Ambiente virtual do programa Poly-Pro.

Fonte: Poly Pro 1.11.

### 3.5 Surf

A superfície apresentada neste trabalho apresenta algumas características interessantes, pois mostra uma superfície NURBS em um ambiente virtual, e neste ambiente pode-se movimentar/deslocar os pontos de controle gerando deformações na superfície.

A interação com o sistema tem o seguinte funcionamento: ao selecionar um ponto de controle, que no caso, estão marcados por esferas, amarelas, vermelhas e azuis, este se torna visível por apresentar uma esfera semi-transparente branca. Em seguida deve-se arrastar o mouse em um dos planos XY, YZ e XZ, Figura 9, para mover o ponto de controle selecionado.

O software apresenta interação direta do usuário com o objeto, podendo arrastar os pontos de controle de forma aleatória.

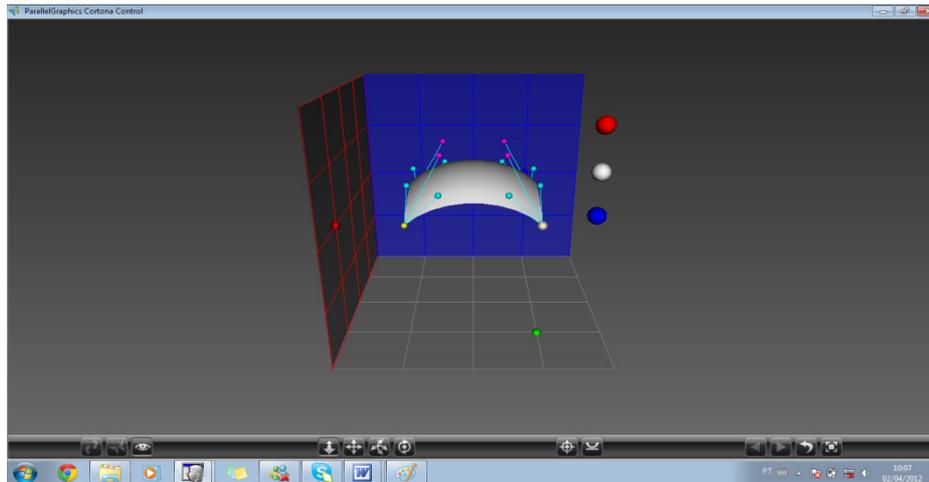


Figura 9: Arquivo VRML, surf.wrl

Fonte: <http://www.ibiblio.org/e-notes/Splines/Inter.htm>

O sistema ainda fornece outras interações, como: a opção de ocultar a superfície, inserir grades e imprimir os valores dos pontos de controle, através do console do VRML. Ao clicar na esfera vermelha que funciona como uma chave de liga/desliga, a superfície é ocultada ou não. O mesmo acontece com a chave branca, deixando a superfície em forma de grade ou não (Figura 10).

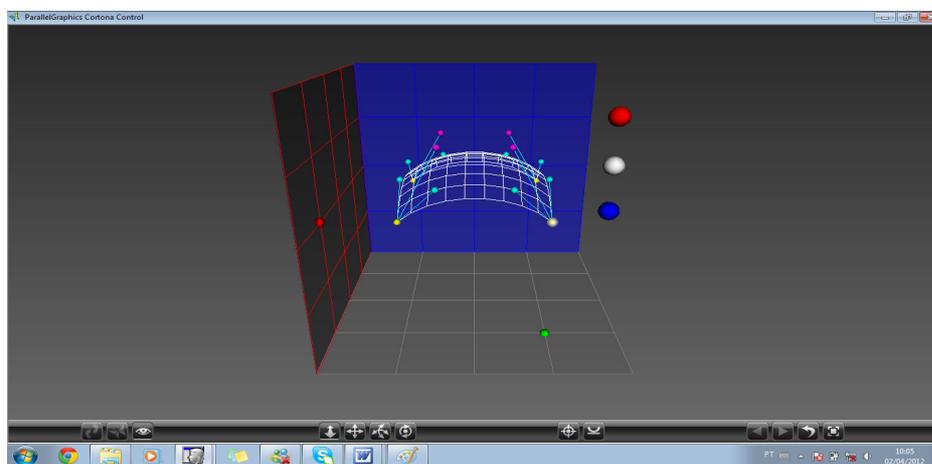


Figura 10: Arquivo VRML, surf.wrl

Fonte: <http://www.ibiblio.org/e-notes/Splines/Inter.htm>

Este projeto também tem a opção de imprimir os pontos de controle, a Cada interação com a superfície, bastando-se clicar sobre a esfera azul, como ilustrado na Figura 11.



Tabela 2: Comparação entre os softwares criados para auxiliar o ensino/aprendizagem.

Sistemas		VR Playground	Corte e Visualização	Poly Pro	Surf
Propriedades					
AV	Sem interação	X	X	✓	X
AV	com Pouca Interação	✓	✓	X	X
AV	com Muita Interação	X	X	X	✓
AV	nenhum Domínio de Objetos Virtuais	X	X	X	X
AV	Pouco Domínio de Objetos Virtuais	✓	X	✓	X
AV	Muito Domínio de Objetos Virtuais	X	✓	X	✓
Visualização	Natural e Intuitiva	X	X	X	X

Analisando a tabela acima, pode-se perceber que não foi detectado nenhum ambiente virtual que suportasse visualização intuitiva e natural dos pontos de controle da superfície NURBS, apenas fazendo a interação sem que o usuário possa ver o que estava acontecendo com os pontos no espaço, por exemplo, a propriedade 12 que garante a deformação local do ponto. Estas propriedades são importantes para providenciar um ambiente mais amigável para o ensino de NURBS.

Assim, este trabalho de pesquisa propõe-se a desenvolver um sistema capaz de absorver estas propriedades. A arquitetura deste sistema é apresentada no capítulo seguinte.

## CAPÍTULO 4 – ARQUITETURA DO SISTEMA

### 4.1 Introdução

Neste capítulo, pretende-se descrever as ferramentas utilizadas para criação do protótipo, utilização das NURBS e o Ambiente Virtual para manipulação e interação com o objeto desenvolvido.

### 4.2 – VRML

O VRML (*Virtual Reality Modeling Language*) é um formato de arquivo que simula ambientes de Realidade Virtual utilizando-se de textos na linguagem ASCII, como mostrado na Figura 12.

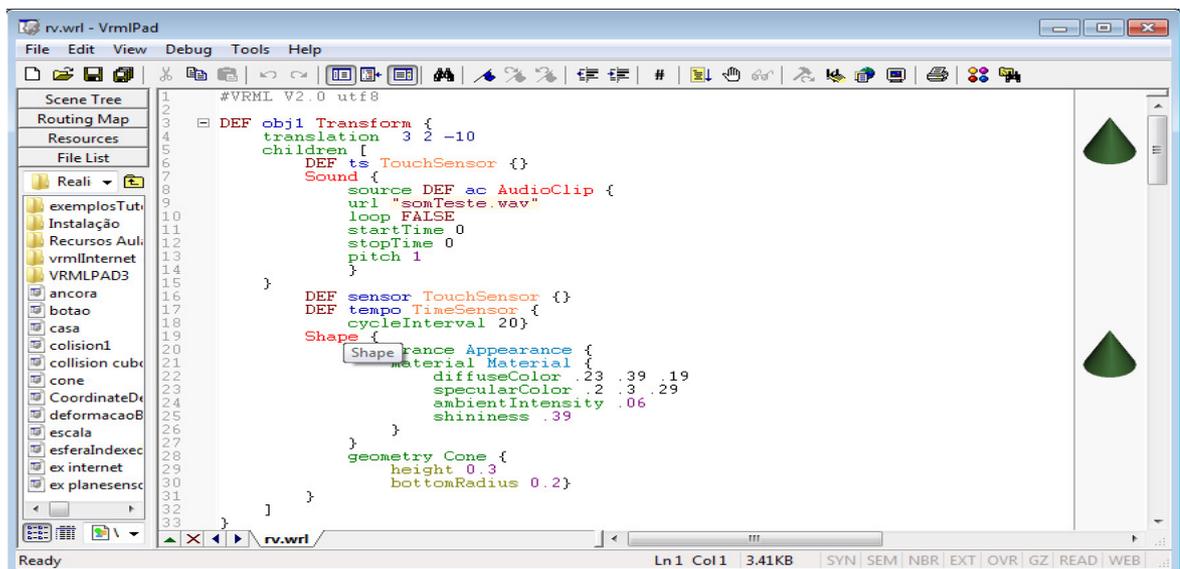


Figura 12: Interface do VRMLPad como parte de um código de AV.

Fonte: VRMLPad

Por meio desta linguagem, escrita em modo texto, é possível criar objetos tridimensionais (3D) podendo-se definir cores, transparência, brilho e textura. Os objetos podem ter formas básicas, como esferas, cubos, ovóides, hexaedros, cones, cilindros, ou formas criadas pelo próprio programador, como as extrusões, ou seja, possibilita a criação de uma infinidade de formas em 3D. Estes podem ser visualizados através de um *plug-in*, como o programa Cortona, na qual sua interface pode ser verificada na Figura 13, segundo Cardoso e Lamounier [19].

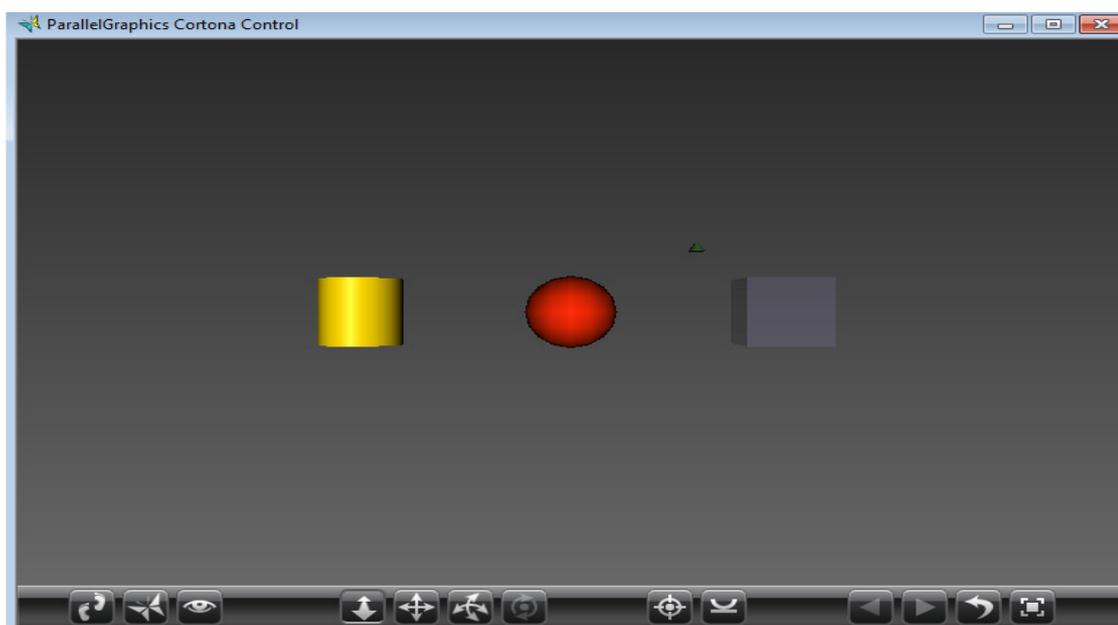


Figura 13: Interface do Cortona com objetos criados com o código mostrado na Figura 12.

Além dos objetos mostrados na figura 13, pode-se acrescentar interatividade a estes por meio de sensores e dessa forma, deslocá-los de posição, acrescentar luz, produzir um som quando o objeto é clicado ou quando o avatar simplesmente se aproxima dele. Além do mais, pode-se abrir um arquivo ou página da Web, ou ainda outra página em VRML, quando o objeto é acionado.

Para desenvolver trabalhos com o VRML, não se necessita de um *software* específico para a criação de arquivos VRML. Apesar de existirem, os objetos podem ser todos criados em modo texto. Usualmente as extensões para esta linguagem é em WRL. Em outros casos, pode-se utilizar o VRMLPad e um visualizador VRML, o Cortona.

Neste trabalho utilizou-se o VRMLPad para escrever o código com um implemento em JavaScript para funcionamento do ambiente virtual e o Cortona como o mecanismo visualizador de todos os programas utilizados.

A maioria dos arquivos VRML contém elementos, como: comentários, valores, campos e nós (objetos), definindo-se nomes para os nós e uso dos nós. Simplificadamente, o VRML caracteriza-se por quatro elementos básicos: Cabeçalho – obrigatório; Protótipos – descrevem as classes de objetos que serão usados na descrição do cenário; Formas, interpoladores, sensores, *scripts* – descrevem o cenário construído; e Rotas – indispensáveis na definição dos comportamentos dos objetos.

De acordo com Cardoso e Lamounier [20], para compreensão, a leitura do arquivo VRML pelo *plug-in* do navegador Cortona, deve conter em seu cabeçalho a instrução “#VRML V2.0 utf8”. Já nos elementos de protótipos (PROTO), deve-se colocar a definição de novos nós (objetos) que possam ser usados no arquivo em questão. As seções de descrição das formas descrevem o cenário construído, onde são exibidas as formas, o navegador e a seção de rotas, as quais são indispensáveis, pois definem as trocas de informações entre os nós (objetos) de descrição de formas, interpoladores, sensores e *scripts*.

A elaboração de cenários tridimensionais, usando VRML, se baseia no desenvolvimento de um grafo de cena, contendo diferentes ramos, ou seja, nós que associados de forma correta, podem ser agrupados ou estarem independentes uns dos outros. A grande diversidade destes nós, incluindo figuras geométricas, propriedades de aparência, sons e vários tipos de nós de agrupamentos, são uma das principais qualidades da linguagem VRML. E ainda, permiti-se a utilização de código através dos protótipos, com base na definição de novos nós (protos) que podem ser aproveitados por outros arquivos e ativados dentro de um arquivo como um nó externo, sem a necessidade de duplicar códigos.

As formas são elaboradas através da associação de elementos 3D geométricos pré-definidos, tais como: cones, cilindros, esferas, caixas, dentre outros, que possuem características próprias e que podem ser combinados a texturas.

A mudança do fundo do ambiente é possibilitada pelo uso de nós específicos, os *backgrounds*, que permitem simular ambientes virtuais diferenciados que se assemelham às condições naturais que variam de um dia nublado à noite.

O controle da aparência dos elementos do cenário é possível, através da inserção de diferentes formas de fontes de luz, com o objetivo de dar mais realismo ao cenário. Pode-se inserir sons e filmes através da utilização de nós específicos, compatíveis com os principais formatos de áudio e vídeo dos tipos: .mpeg, .mpg, .mid, .wav.

Como cita Rosso e Rojas [21], as animações podem ser facilitadas através de *scripts*, que tem a função de dar mais vida ao ambiente virtual, auxiliando na troca de informações entre os elementos do mundo virtual. Esta ferramenta possibilita animações e dinamos às

formas, quando se necessita tomar decisões, realizar repetições ou guardar informações, podendo-se aproveitar as características do nó *script*.

### 4.3 – JavaScript

A linguagem JavaScript foi originalmente desenvolvida por Brendan Eich da Netscape com o objetivo de obter uma linguagem intermediária entre a linguagem Java e o HTML. Assim, o JavaScript é um programa livre ou gratuito, de fácil utilização e capaz de vincular objetos e recursos da linguagem Java e HTML. O JavaScript pode ser usado por autores de páginas HTML para controlar a interação e comportamento de suas páginas, segundo Silva [22].

Uma das vantagens do JavaScript se dá pelo fato do código, JavaScript rodar localmente no navegador do usuário e não em um servidor remoto, possibilitando ao navegador responder a tais ações mais rapidamente. A linguagem JavaScript é interpretada pelo *browser* Web que possui os recursos citados (como o Netscape e o Internet Explorer).

Com o uso de códigos JavaScript pode-se implementar respostas a eventos do usuário, como cliques do mouse, movimentos do mouse sobre um *link* e a entrada de dados em um formulário. Possibilita criar páginas dinâmicas, cujo conteúdo muda de acordo com as requisições do usuário, ou mesmo disparar sons ou executar *applets* quando o usuário entra ou sai de uma página, permitindo uma maior interatividade com o usuário. Assim, é de grande utilidade, o JavaScript, na elaboração de interfaces mais simples para o usuário.

No presente trabalho, para controlar a interatividade das cenas, Ambientes Virtuais do Cortona, com códigos na linguagem VRMLPad, foi adicionada a linguagem JavaScript, uma vez que esta pode controlar a interatividade de um ambiente por *scripts* de forma mais eficiente. Como se pode observar no código mostrado a seguir que controla a posição dos pontos de controle.

```

489  DEF Script Script {
490    eventIn SFVec3f posUsuario
491    eventIn SFBool recebe
...
503    eventOut MFString posString

```

```

504  eventOut    SFFloat    value_changed
505  eventOut SFVec3f pos_inicial
...
518  field SFFloat min2 -10
519  field SFFloat max2 10
520  field SFFloat cur 0
521  field SFFloat cur1 0
...
528  url ["javascript:
529  function posUsuario(value) {
530  x = Math.round(value[0] * 10000) / 10000;
531  y = Math.round(value[1] * 10000) / 10000;
532  z = Math.round(value[2] * 10000) / 10000;
533  posString[0] ="+ x + ";
534  posString[1] ="+ y + ";
535  posString[2] ="+ z + "};
...
633  debug [0] = ' ';
634      string_changed1[0] = ' ';
635      string_changed2[2] = ' ';
636      string_changed3[0] = ' ';
637  }
638      ",]}

640  ROUTE sensor.isActive          TO Script.recebe
641  ROUTE Marcador.translation     TO Script.posUsuario
643  ROUTE Proximity.position_changed TO UsePos.translation
644  ROUTE Proximity.orientation_changed TO UsePos.rotation
645  ROUTE Script.posString        TO Text.string

```

A figura 14 mostra a interação dos pontos de controle e o painel de controle do Cortona antes da execução do código acima. E, a figura 15 mostra a interação dos pontos de controle e o painel de controle do Cortona após a execução do código descrito.

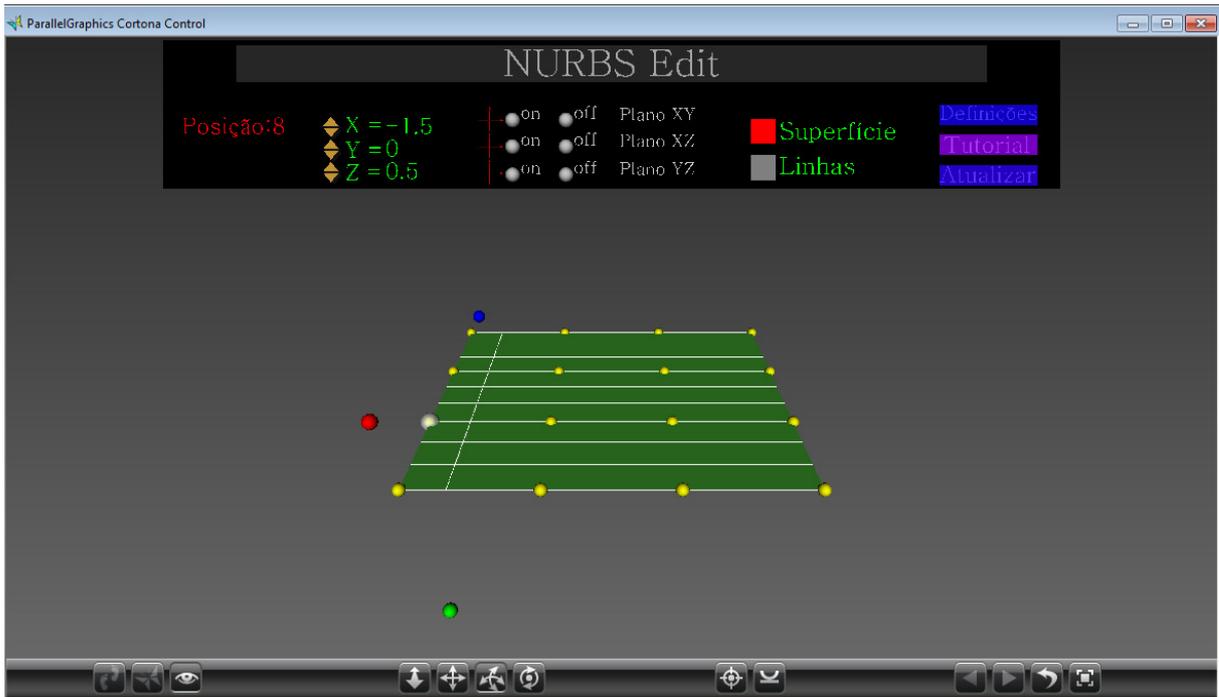


Figura 14: Figura mostrando a interação dos Pontos de Controle com o Painel.

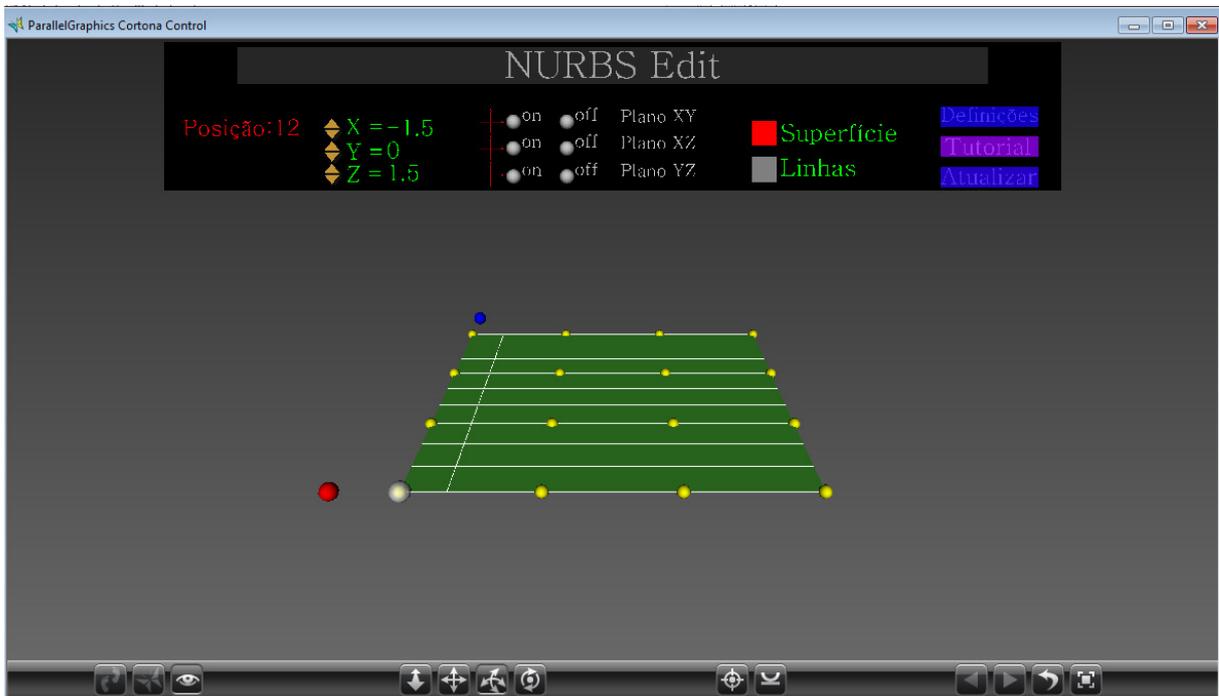


Figura 15: Figura mostrando a interação dos Pontos de Controle com o Painel após a ação do código acima.

## 4.4 Arquitetura do Sistema

A interface do sistema foi desenvolvida para o estudo de Geometria NURBS, com a possibilidade de visualização e deformação de tais superfícies. Permite que alunos e professores tenham acesso ao *software* sem a necessidade de conhecimentos específicos na área de informática. Na Figura 16 tem-se a arquitetura do sistema protótipo.

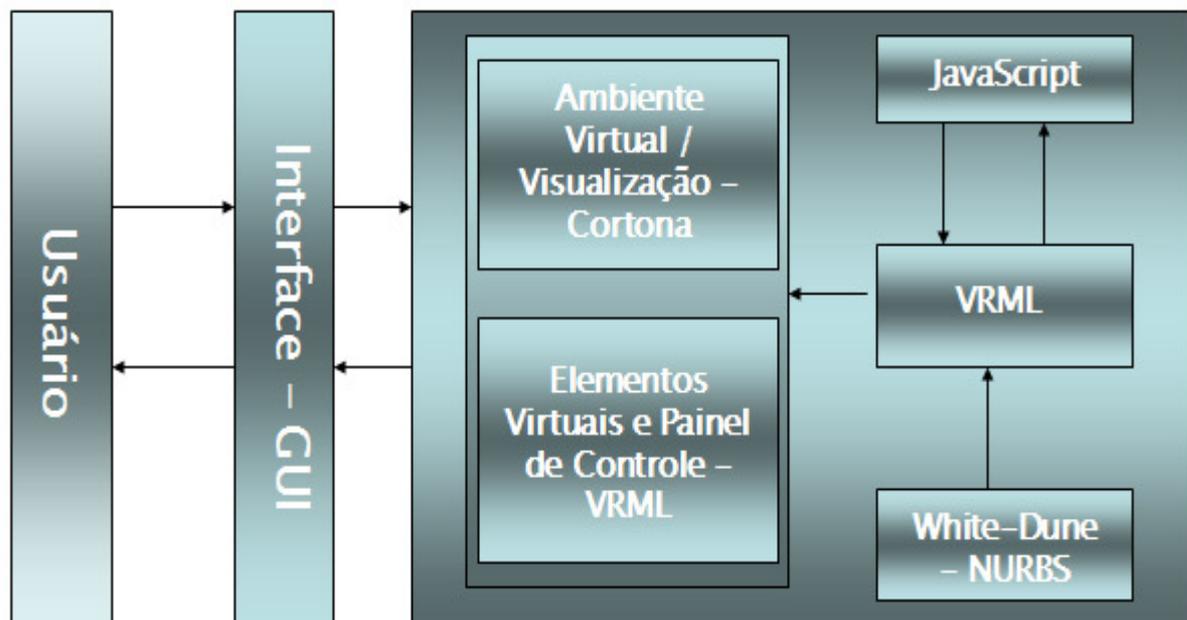


Figura 16: Esquema geral da Arquitetura do Sistema desenvolvido.

O sistema proposto tem basicamente a seguinte arquitetura:

- Ambiente Virtual, que é composto pela Visualização do Cortona;
- Núcleo Gerenciador de deformação onde está os elementos virtuais de controle do VRML, o módulo da Internet;
- GUI – Interface Gráfica que comunica com o Usuário.

Para a elaboração do sistema, se fez necessário ter em mãos os programas, Cortona, VRMLPad, White\_dune e JavaScript. O Cortona é o *plug-in* responsável por exibir as aplicações desenvolvidas no VRML, já o VRMLPad, foi o compilador do código VRML e do JavaScript, utilizado durante todo o desenvolvimento do protótipo. O *software* White\_dune foi utilizado para gerar a superfície NURBS que foi exportada para o VRML.

O download gratuito do *plug-in* Cortona pôde ser feito através do *link*: <http://www.parallelgraphics.com/bin/cortvrml.exe>. O download do VRMLPad pôde ser feito gratuitamente do seguinte *link*: <http://www.parallelgraphics.com/bin/vpad21.exe>.

A interface gráfica GUI foi importante em disponibilizar o Ambiente Virtual, no qual o cenário é composto por objetos previamente definidos para interação, visualização e navegação no ambiente, Figura 17.

Nos sub-tópicos seguintes, há a descrição de cada parte da arquitetura desenvolvida.

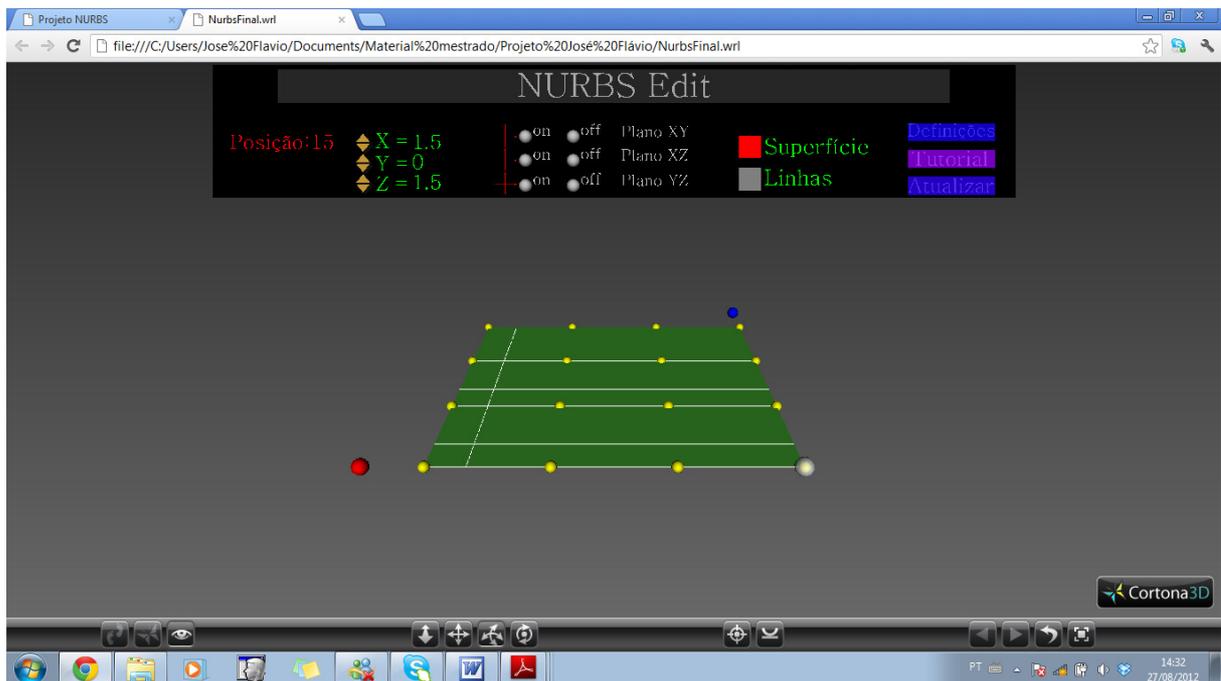


Figura 17: Interface Gráfica com o Usuário.

## 4.5 Ambiente Virtual

O ambiente virtual é composto pelos elementos virtuais e o painel de controle que foram implementados utilizando-se da linguagem VRML. A composição do ambiente virtual se dá pela superfície, desenvolvida no *software* White\_Dune, no qual foi criada uma superfície NURBS plana com dezesseis pontos de controle e exportada para o VRML, como mostrado na Figura 18.

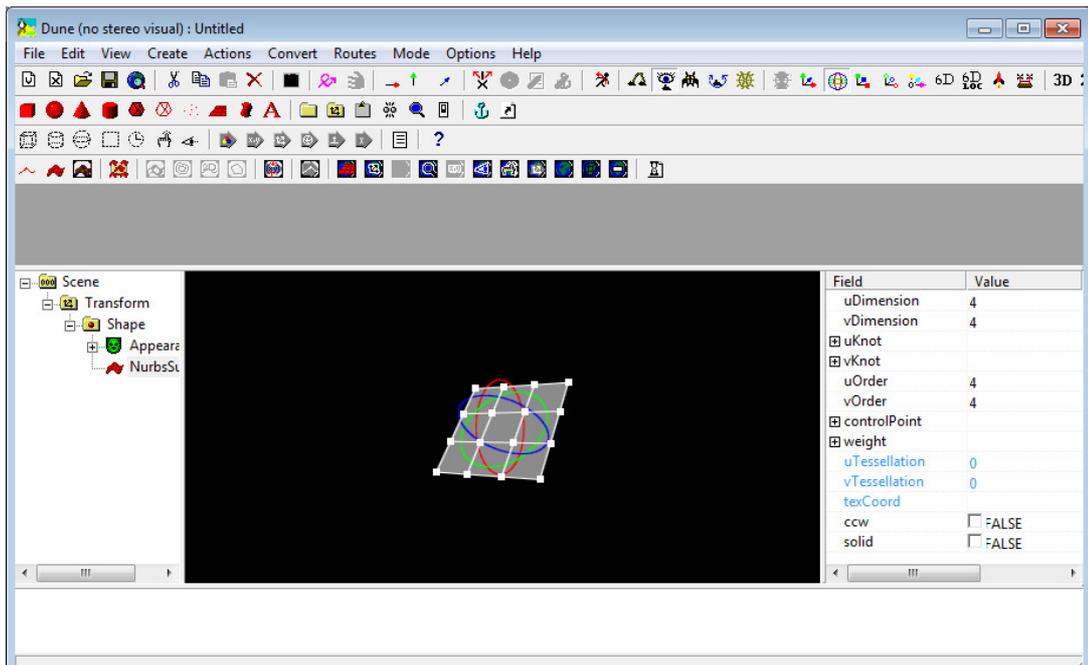


Figura 18: Superfície plana gerada no *software* White\_Dune.

Na figura 19 verifica-se a mesma imagem mostrada na Figura 18, entretanto exportada para o VRML visualizada no Cortona.

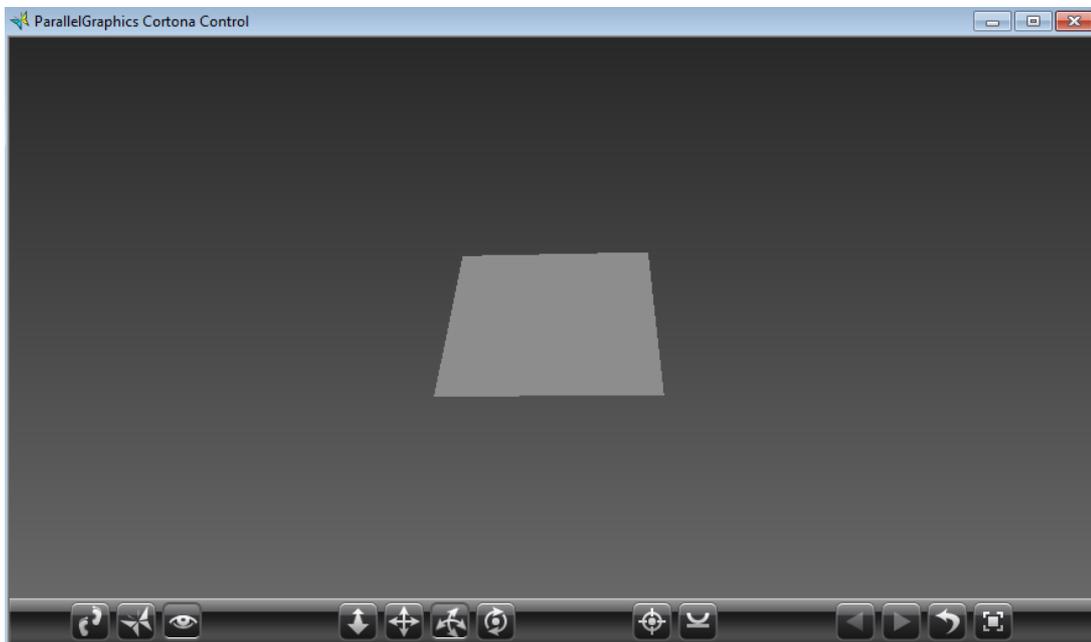


Figura 19: Superfície importada do *software* White-Dune visualizada no Cortona.

A superfície utilizada estava estática, sem interações, dessa forma, procedeu-se à marcação dos pontos de controle (esferas amarelas) para facilitar a visualização dos mesmos e em cada ponto, foi inserido um sensor, que ao clique do mouse, uma segunda esfera (esfera semitransparente e denominada de marcador) é transladada para a posição selecionada. Outros

três marcadores (esfera vermelha, azul e verde) são usados para visualizar a posição do ponto nos planos e para que o usuário possa acompanhar o deslocamento dos pontos de controle no ambiente virtual e ainda podem ser visualizadas através do painel as coordenadas X, Y, Z, com os respectivos valores no espaço tridimensional, como mostrado na Figura 20.

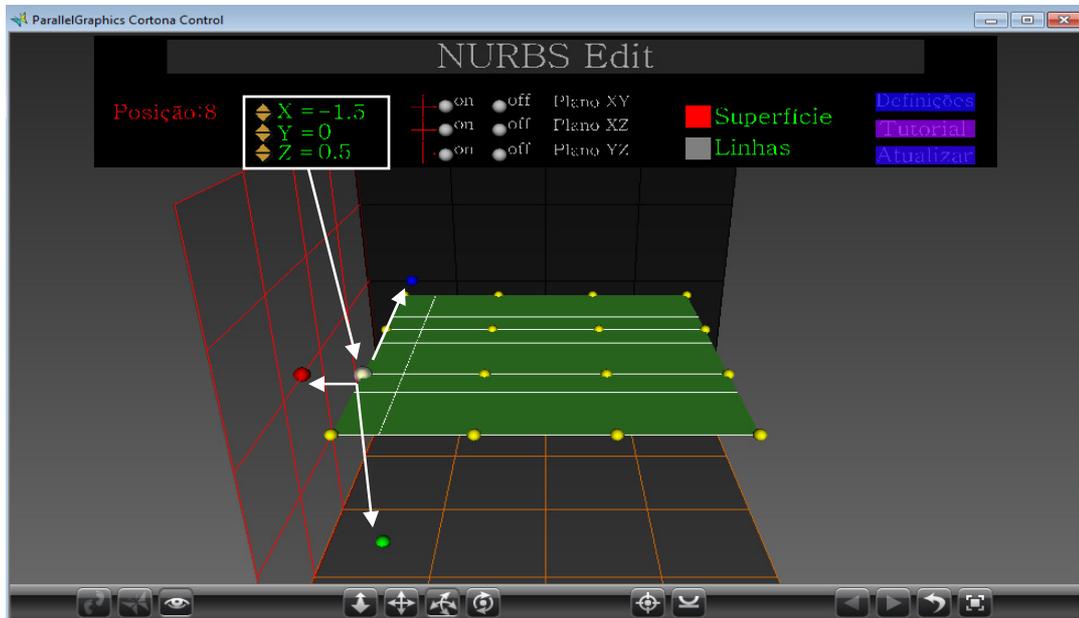


Figura 20: Ambiente Virtual do protótipo criado.

São itens de comandos enumerados e constantes na barra superior do Ambiente Virtual, como mostrados na Figura 21.

- 1) Posição - onde é indicado o valor de cada ponto de controle, da posição 0 ate a posição 15, indicando ao usuário que a interação está ocorrendo na posição selecionada;
- 2) Coordenadas - indicação dos valores do ponto da posição selecionada, no plano XYZ;
- 3) Planos XY, XZ e YZ - tem-se a opção on/off para visualizar ou não o plano no ambiente, com o objetivo de melhorar a visualização e compreensão das interações realizadas por parte do usuário;
- 4) Superfícies e linhas - são ativados por sensores de toque, em que o vermelho tem a função de ocultar ou não a superfície, já a caixa cinza tem a função de ocultar as linhas de grade da superfície;
- 5) Definições - tem-se a visualização de um organizador gráfico em forma de árvore com *links* para páginas HTML com as respectivas definições.
- 6) Tutorial - direciona o usuário para um ambiente em que são apresentadas algumas explicações de como se utiliza o ambiente virtual;
- 7) Atualizar - redireciona o ambiente, retornando para o início, para novas interações.

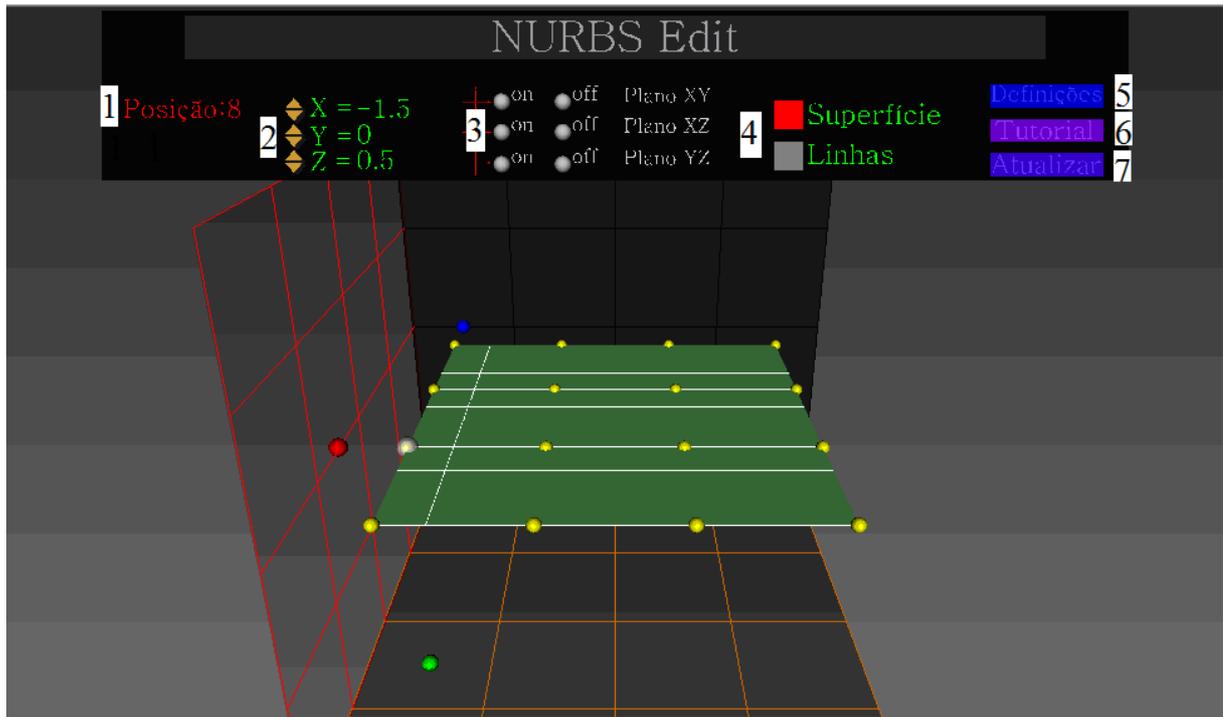


Figura 21: Ambiente Virtual desenvolvido e seus respectivos itens de funcionamento.

A arquitetura proposta neste capítulo procurou estimular o usuário para a aprendizagem das NURBS, dando ao usuário a opção de visualizar a posição do marcador nos 16 pontos de controle distribuídos (esferas amarelas) sobre a superfície e ainda as coordenadas com a possibilidade de acompanhar o deslocamento do ponto selecionado, tendo a escolha de interagir com o AV tanto clicando e arrastando o ponto quanto usando as setas, ao lado esquerdo de cada coordenada X, Y, Z.

Assim a linguagem VRML e JavaScript suportaram o desenvolvimento das interações com os pontos de controle da superfície NURBS em que o objeto pode ser manipulado de forma interativa e ainda mostrando no painel os deslocamentos realizados.

## 4.6 Interface – GUI

A GUI possibilita o acesso ao ambiente virtual, onde se podem realizar as interações e o acesso às definições dos conteúdos do mapa conceitual das NURBS e ao tutorial do sistema. A proposta de trabalho dos Mapas Conceituais está baseada na ideia fundamental da Psicologia Cognitiva de Ausubel que estabelece que a aprendizagem ocorre por assimilação

de novos conceitos e proposições na estrutura cognitiva do aluno. Novas ideias e informações são aprendidas, na medida em que existam pontos de ancoragem.

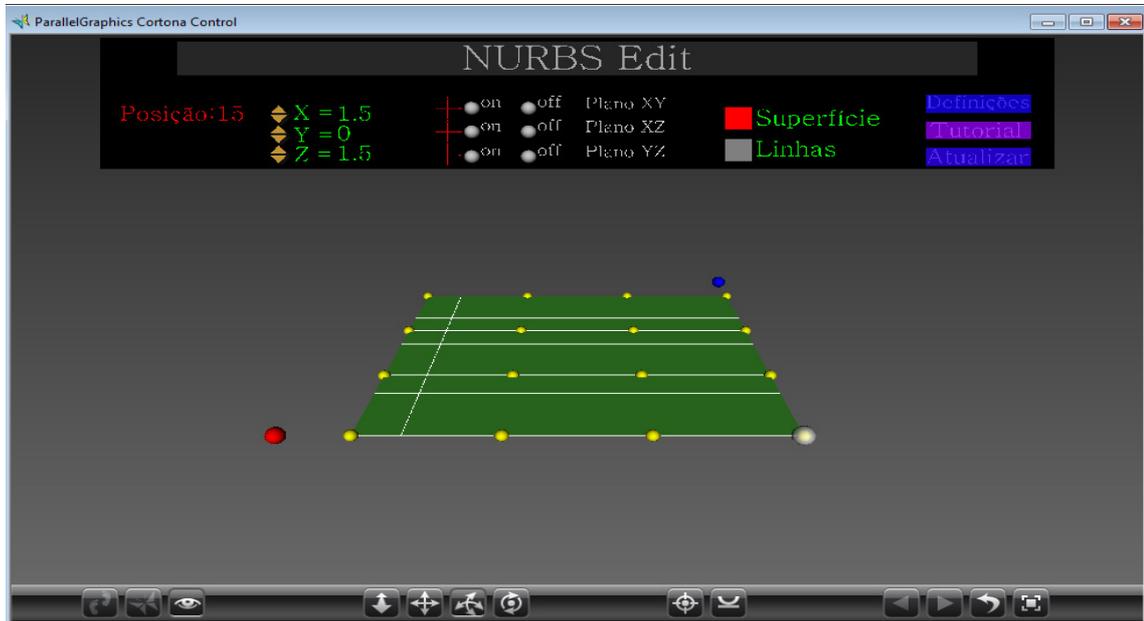


Figura 22: Interface Gráfica com o Usuário – GUI

São componentes da interface gráfica GUI, Figura 22, o Ambiente Virtual formado pelo painel de controle e uma superfície plana gerada no White-Dune e exportada para o VRML, no qual permite ao usuário visualizar graficamente e de modo interativo o resultado de suas ações no sistema. Podendo realizar as deformações e gerando novas superfícies. Pode-se, realizar as interações na superfície, por meio da alteração dos parâmetros dos pontos de controle, e visualizar a superfície e as novas posições dos pontos de controle, no próprio ambiente virtual. O usuário terá a opção de utilizar o *plug-in* do Cortona para aumentar ou diminuir o zoom, transladar e rodar a superfície, como mostra a Figura 23.

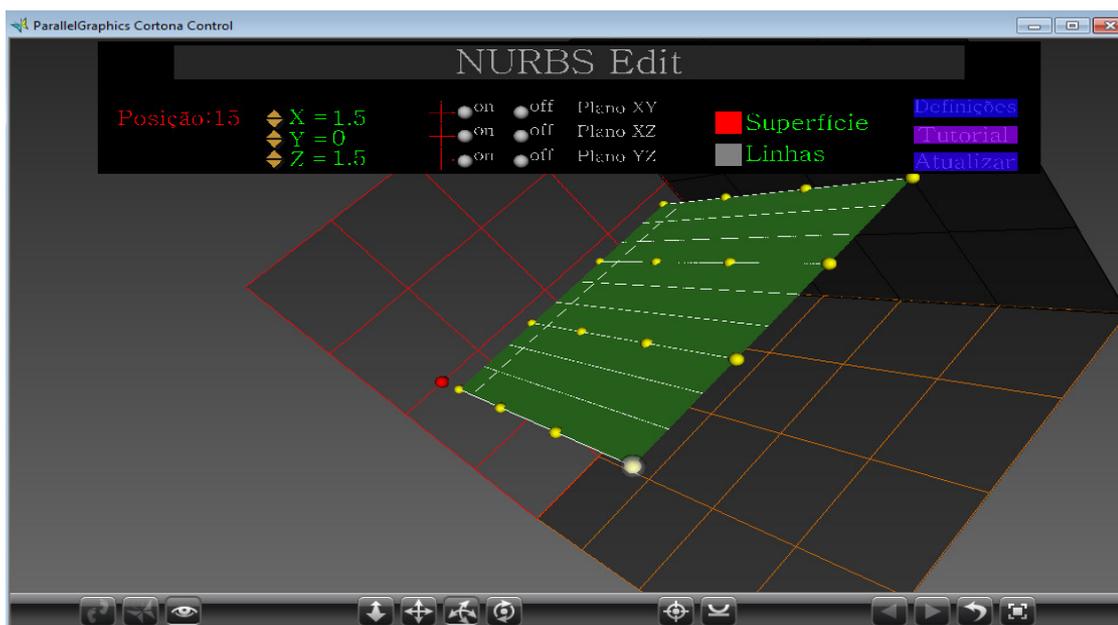


Figura 23: ambiente virtual após o uso do *plug-in* do cortona.

#### 4.7 Elementos Virtuais e Painel de Controle

No conjunto Elementos Virtuais e Painel de Controle, englobam as funções implementadas em JavaScript que controlam as interações do usuário com o ambiente virtual, realizando as deformações na superfície. No ambiente virtual o usuário seleciona um ponto com o mouse e a partir do segundo clique realiza as interações. O conjunto de elementos virtuais e painel de controle controla todas as ações do usuário, tanto na deformação da superfície quanto no retorno dos dados no painel de controle.

No ambiente o usuário tem-se à sua disposição a opção de escolher qualquer ponto de controle da superfície para suas interações e assim gerar a superfície que desejar, deslocando os pontos um de cada vez, até a posição requerida.

O sistema foi elaborado, para que o usuário pudesse manuseá-lo, sem a necessidade de conhecimentos técnicos de informática. A utilização deste sistema será por meio simples de selecionar e deformar os pontos da superfície, por meio do clique e arrasto do mouse ou seleção dos pontos e com o mouse clicar sobre as setas no painel de controle.

## **4.8 Considerações Finais**

A arquitetura proposta neste capítulo procurou estimular o usuário a aprender alguns conceitos das NURBS. A linguagem VRML e JavaScript deram o suporte necessário para a criação do ambiente e as interações propostas. O próximo capítulo apresenta detalhes utilizados para a criação do ambiente virtual, baseado na arquitetura proposta.

## CAPÍTULO 5 – DETALHES DE CONSTRUÇÃO DO SISTEMA

### 5.1 Introdução

Neste capítulo são apresentados as particularidades da implementação do Ambiente Virtual. Aqui, são ressaltadas as linguagens utilizadas, VRML e JavaScript, e são abordados trechos do código e explicações sobre suas respectivas funcionalidades no ambiente virtual.

### 5.2 Implementação do Ambiente Virtual

O Ambiente Virtual apresenta-se através de uma interface, onde está presente o painel de controle e a superfície NURBS implementados em VRML. A linguagem de modelagem VRML descreve as geometrias e aparências diretamente no código fonte, por meio do editor VRMLPad 2.0 e assim, diminui o tamanho dos arquivos, facilitando o compartilhamento na rede de internet.

Utilizou-se de uma superfície gerada no *software* White-Dune que foi exportada para o VRML, e em seguida procedeu-se a marcação dos pontos de controle, com formas geométricas primitivas. Optou-se pela esfera amarela, para proporcionar uma melhor visualização dos pontos de controle da superfície. O trecho de código a seguir mostra a codificação da geometria, esfera amarela.

```
399 Transform {translation IS translation children [  
400 Shape {appearance Appearance {material Material {diffuseColor IS col }}  
401 geometry Sphere {radius0.05}}
```

Na Figura 24, pode-se verificar a representação gráfica do código acima implementado, o qual gerou a esfera amarela.

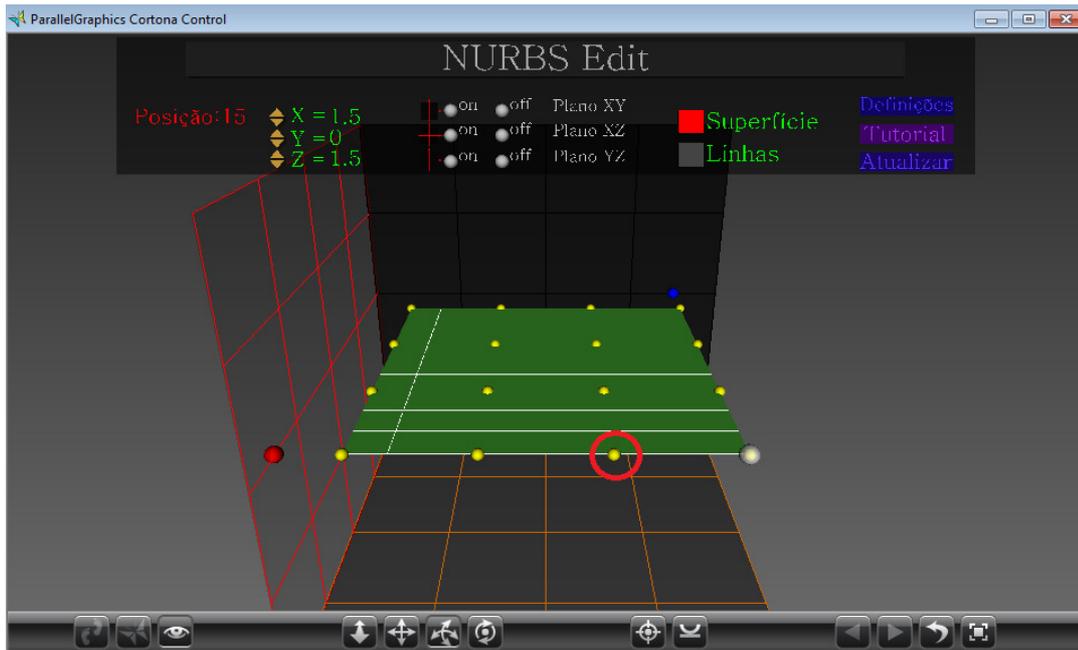


Figura 24: Representação gráfica da esfera amarela circundada, em destaque, por um círculo vermelho.

Após a marcação dos pontos de controle pela esfera amarela, foi desenvolvida uma nova esfera (branca semitransparente apresentada na Figura 25). Esta esfera serve como marcador dos pontos, para a realização das interações propostas. Na esfera branca foram colocados alguns sensores que servem para realizar as interações. Por exemplo, o sensor PlaneSensor, que converte as ações do usuário em movimentos num plano 2D. O sensor TouchSensor que serve para a “captura de toque”. Neste caso irá selecionar o ponto, marcando a sua posição (índice ou valor numérico) no ambiente virtual e com o cursor sobre o ponto, o botão do mouse pode ser pressionado e assim, pode-se arrastar o ponto no ambiente fazendo com que a superfície se movimente, sofrendo deslocamento. Abaixo se tem a descrição do código sobre a implementação da esfera branca com seus respectivos sensores.

```

475 DEF Marcador Transform {children [Shape {
476 appearance Appearance {material Material {diffuseColor 1 1 1 transparency
0.3}}
477 geometry Sphere {radius0.08}}]}
479 DEF sensor TimeSensor {cycleInterval 0.1}
480 DEF sensor1 TimeSensor {cycleInterval 0.1}
481 DEF sensor2 TimeSensor {cycleInterval 0.1}
482 DEF sensor_plano PlaneSensor {
483 minPosition 0 50 maxPosition 0 50}

```

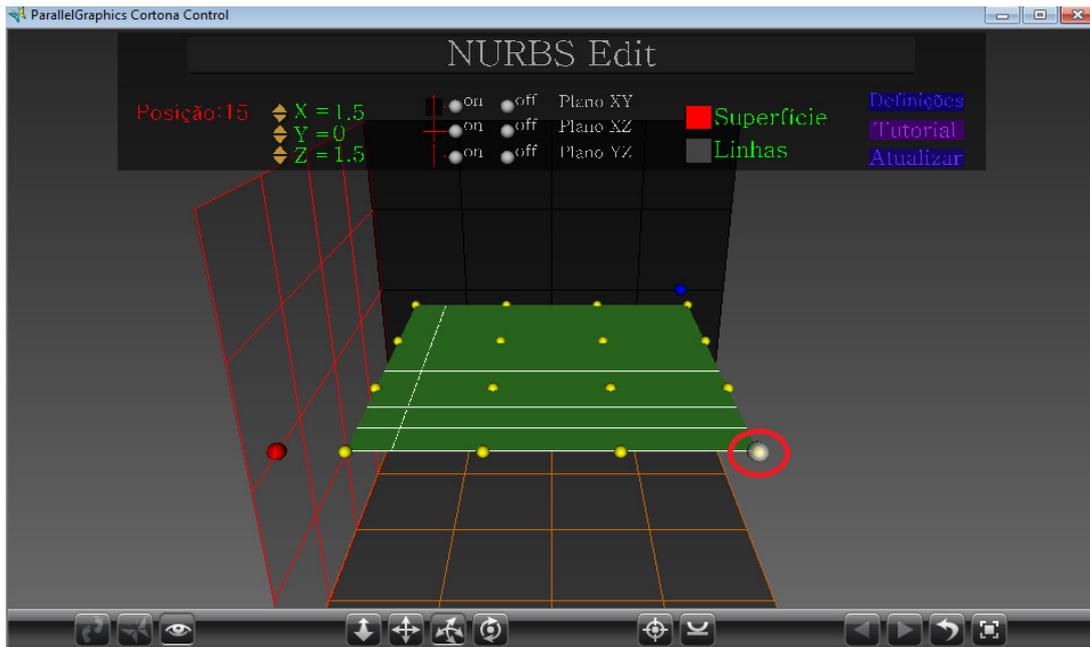


Figura 25: O objeto em destaque, circundado por um círculo vermelho, a esfera branca semitransparente, foi criado para marcar os pontos e as interações.

### 5.3 Implementação dos Deslocamentos da Superfície

Após a modelagem da superfície, foram atribuídos à mesma, movimentos e deslocamentos. O movimento dos pontos de controle altera a forma da superfície, de acordo com as funções NURBS. Nesta secção foram abordadas algumas técnicas de programação em VRML e JavaScript que contribuíram para a criação do Ambiente Virtual e movimentos da superfície.

No Ambiente Virtual desenvolvido, o usuário pode selecionar um ponto na superfície e através do painel de controle terá a visualização da posição do ponto selecionado. O código VRML, abaixo, mostra como esta ação interativa foi criada através da linguagem JavaScript.

```

360 #string índices
362 DEF texIndice Transform {translation 0 0 0
363     children [Transform {
364         translation -0.57 0.285 -1
365         #scale 0.01 0.01 0.01

```

```

366         children Shape {appearance Appearance {material Material
{diffuseColor 1 0 0}}
367             geometry DEF debug Text {
368                 fontStyle FontStyle {size 0.03
369                     #spacing 0.3 style "bold" family "serif"
370                 }
371             }]]]]}
...
770 DEF scr Script {
772 eventIn SFVec3f setXY
773 eventIn SFVec3f setYZ
774 eventIn SFVec3f setXZ
775 eventIn SFVec3f setXYZ
776 eventIn SFInt32 getId
777 eventIn SFBool setIFS
778 eventIn SFBool setGrid
780 eventIn SFBool get
781 eventIn SFVec3f position
...
918 function position(value) {
919     debug = new MFString('Índice:' + iN +");}
920     function get(value){if(value)debug=position;}
921 "]}
...
950 ROUTE Proximity.position_changed TO texIndice.translation
951 ROUTE Proximity.orientation_changed TO texIndice.rotation
952 ROUTE Proximity.position_changed TO scr.position
953 ROUTE scr.debug TO debug.string
...

```

A função “position”, código 781 e 918, é ativada por meio do sensor de toque e pelas rotas, como mostra a Figura 26.

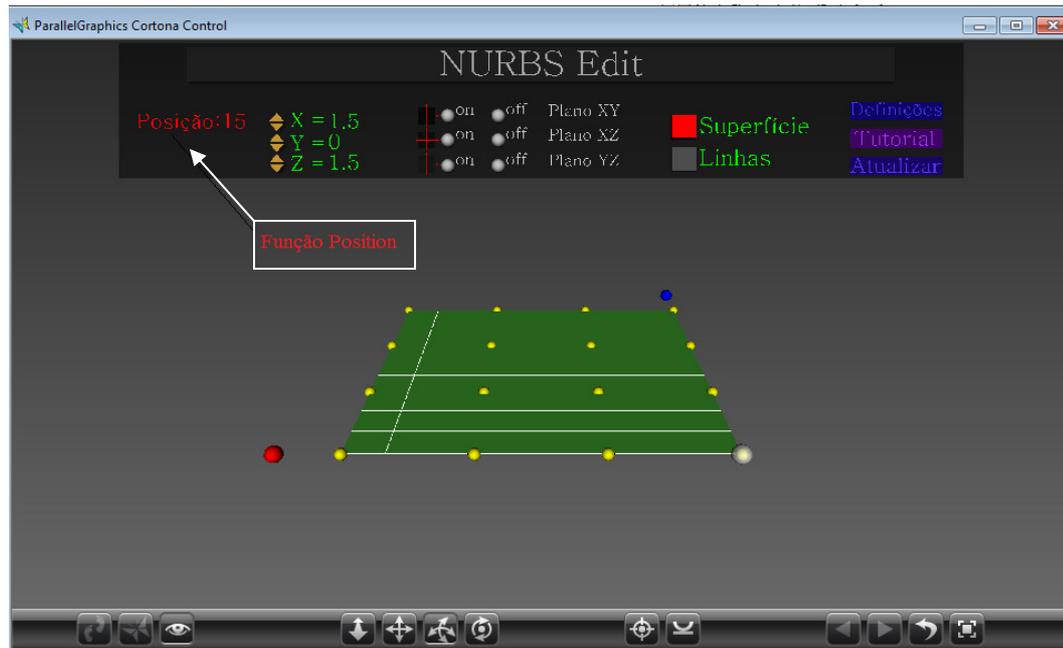


Figura 26: Função “position” mostra a posição do marcador

Outro processo interativo criado no Ambiente Virtual é a posição dos pontos no espaço XYZ em que se mostram as coordenadas dos pontos de controle no painel, de acordo com a interação do usuário com o ambiente. O código VRML que permite esta interação pode ser verificado:

```

4     DEF Proximity ProximitySensor { size 1000 1000 1000 }
...
346    DEF UsePos Transform {
347      translation 0 0 0
348      children [
349        # Text position
350        Transform {
351          translation -0.3 0.285 -1
352          children [
353            Shape {appearance Appearance {material Material {diffuseColor 0 1
0}}
354            geometry DEF Text Text {
355              string "Position"
356              fontStyle FontStyle {
357                size      0.03

```

```

358     } } } } } }
...
489     DEF Script Script {
490         eventIn SFVec3f posUsuario
491         eventIn SFBool recebe
...
503         eventOut MFString posString
...
528         url ["javascript:
529             function posUsuario(value) {
530                 x = Math.round(value[0] * 10000) / 10000;
531                 y = Math.round(value[1] * 10000) / 10000;
532                 z = Math.round(value[2] * 10000) / 10000;
533                 posString[0] ="+ x + ";
534                 posString[1] ="+ y + ";
535                 posString[2] ="+ z + ";}    }
538         function recebe(value){if(value)posString=posUsuario;},]}
...
640     ROUTE sensor.isActive                TO Script.recebe
641     ROUTE Marcador.translation            TO Script.posUsuario
643     ROUTE Proximity.position_changed     TO UsePos.translation
644     ROUTE Proximity.orientation_changed  TO UsePos.rotation
645     ROUTE Script.posString                TO Text.string

```

Como se pode notar no código acima, utilizou-se do nó “ProximitySensor”, código 4, para fixar as coordenadas no painel de controle. Para tornar o ambiente mais intuitivo, optou-se por utilizar as coordenadas para que o usuário possa visualizar a sua interação com a superfície, como mostra a Figura 27.

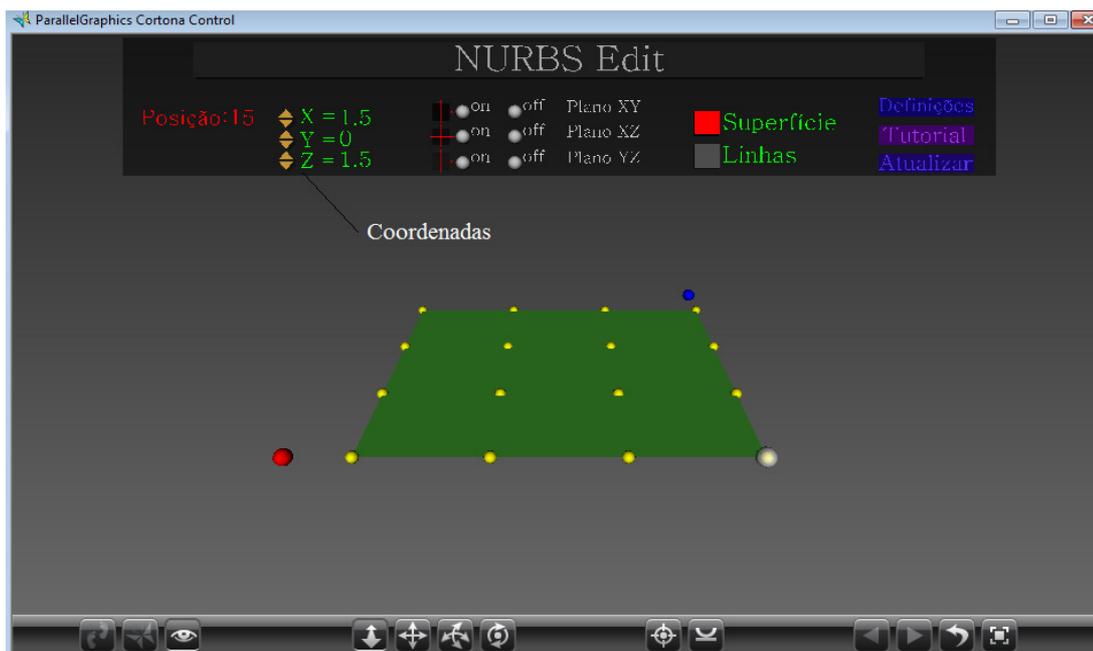


Figura 27: Função “posUsuario” mostra as coordenadas do ponto de controle.

Os fragmentos de códigos mencionados anteriormente são alguns exemplos de técnicas utilizadas na implementação da superfície NURBS. Necessitou-se de diferentes técnicas para desenvolver o ambiente, sendo que a maioria delas foi feita para melhorar a interação do usuário com a superfície.

## 5.4 Considerações Finais

Para o desenvolvimento dos códigos do Ambiente Virtual apresentado, exigiram-se maiores estudos sobre programação. Contudo, foram implementadas técnicas simples como, por exemplo, a criação do organizador gráfico tridimensional para as definições. Para a troca de tela (navegação) do organizador utilizou-se o nó “Anchor” que conecta um arquivo VRML a outro com extensão, .wrl ou .html, e esta conexão se dá na forma de *link*.

Concluiu-se que as técnicas empregadas para o desenvolvimento do sistema, independente de seu grau de dificuldade, atingiram os objetivos propostos, proporcionando a criação de um Ambiente Virtual com suporte para a navegação, interação e a manipulação do objeto.

## CAPÍTULO 6 – AVALIAÇÃO DO SISTEMA E RESULTADOS

### 6.1 Introdução

Este capítulo apresenta uma descrição sobre o funcionamento do protótipo NURBS Edit, denominação do Ambiente Virtual desenvolvido e disponível em HTML. Há a descrição de tópicos importantes sobre a avaliação deste sistema. Para a avaliação, foram considerados aspectos sobre a base teórica, o desenvolvimento ou facilidade de manipulação e a aplicação do AV ao ensino. A avaliação do sistema foi feita por um grupo de professores e de alunos de graduação em Engenharia Elétrica, através de um questionário. Os resultados obtidos foram discutidos e as limitações ressaltadas para que possam ser sanadas em trabalhos futuros.

### 6.2 Utilizações do Sistema

O primeiro contato do usuário com o protótipo NURBS Edit se dá através de uma página HTML que contém cinco *links*, como mostra a figura 28.



Figura 28: Página de Abertura do Software NURBS Edit.

Observam-se na figura 29 os seguintes caracteres:

- 1) Software NURBS Edit – abre o ambiente virtual;
- 2) Mapa Conceitual – acessa o Mapa Conceitual das Superfícies NURBS, Figura 29 A.
- 3) Simulações – abre uma página em HTML com 4 opções, para que o usuário escolha uma e realize os passos indicados, formando uma nova superfície, Figura 29 B.
- 4) Manual do Usuário – permite o acesso a uma página HTML com informações sobre funções dos ícones e explicações de como realizar ações não intuitivas no sistema, Figura 29 C.
- 5) Equipe – Apresenta uma página HTML contendo os nomes dos responsáveis pela criação do Ambiente Virtual, Figura 29 D.

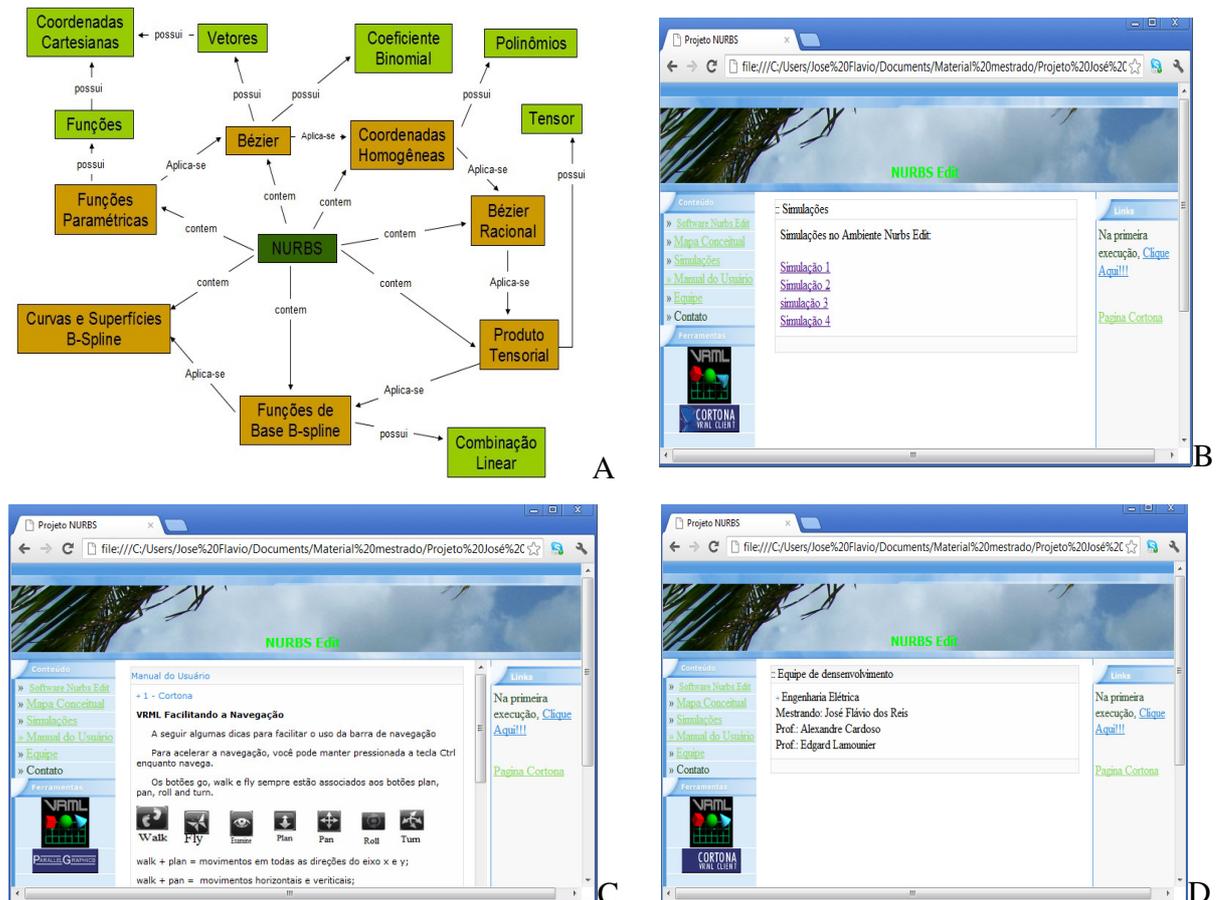


Figura 29: Apresentação dos itens que compõem o Ambiente Virtual de NURBS Edit.

Resumidamente o funcionamento do protótipo se dá após acessar o *link* “Software NURBS Edit” localizado na barra vertical à esquerda, é apresentado ao usuário o Ambiente Virtual, figura 22. Nesse ambiente, o usuário pode realizar as interações, utilizando-se do

mouse, o qual altera as posições dos pontos de controle diretamente na superfície ou através das setas; inclui ou retira do ambiente a superfície, as linhas e os planos cartesianos; através do botão Definições, pode-se acessar um ambiente disposto em forma de árvore, no qual apresenta o conteúdo das NURBS com *links* para páginas HTML com suas respectivas definições; no botão Tutorial abre-se uma página HTML com informações sobre o sistema; e no ícone “Atualizar” atualiza-se a página, reiniciando o sistema.

### 6.3 Estudo de Caso

Como estudo de caso optou-se por desenvolver uma superfície bem conhecida no meio acadêmico, no caso a “sela”. Para construir a superfície os passos abaixo precisam ser seguidos:

1 – Para construir a figura na forma de uma sela, clique no botão vermelho para deixar a superfície em forma de linhas e assim deixar os pontos visíveis para a seleção ou a cada interação use os ícones do cortona (plug-in utilizado para visualização dos objetos virtuais) para girar a superfície e assim ter acesso aos pontos.

2 – Na Figura 30 nota-se a disposição dos pontos de controle, onde se pode visualizar através da marcação a posição de cada ponto sobre a superfície. Assim, seleciona-se com o mouse a esfera de posição 1, clique nas setas do painel, até as coordenadas (x, y, z) ficarem na posição (-0.5, 0, -1.0).

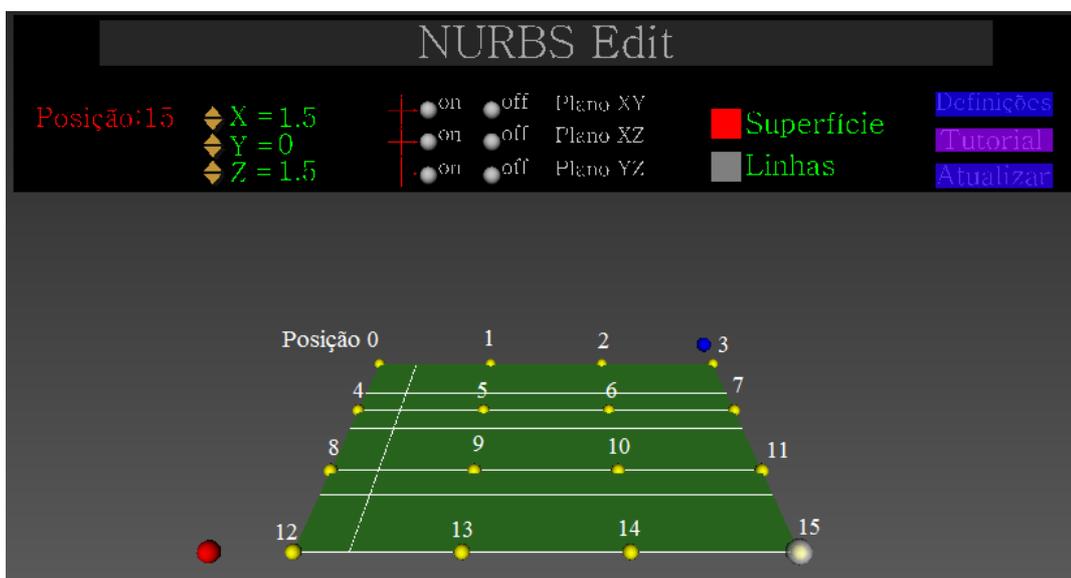


Figura 30: Enumera a posição de cada ponto de controle

3 – Seleciona-se a esfera amarela de posição 2 e com o mouse, clique nas setas do painel, até as coordenadas (x, y, z), ficarem na posição (0.5, 0, -1.0).

4 – Repita o procedimento para que a posição 4 se desloque para (-1.5, 1.5, -0.5).

5 – Desloque a posição 5 para (-0.5, 0.5, -0.5).

6 – Desloque a posição 6 para (0.5, 0.5, -0.5).

7 – Desloque a posição 7 para (1.5, 1.5, -0.5).

8 – Desloque a posição 8 para (-1.5, 1.5, 0.5).

9 – Desloque a posição 9 para (-0.5, 0.5, 0.5).

10 – Desloque a posição 10 para (0.5, 0.5, 0.5).

11 – Desloque a posição 11 para (1.5, 1.5, 0.5).

12 – Desloque a posição 13 para (-0.5, 0, 1.0).

13 – Desloque a posição 14 para (0.5, 0, 1.0).

14 – Após essas interações gire a figura, usando os ícones da parte de baixo do *Plug-in* Cortona, para visualizar o objeto ou utilize o painel de controle para tirar as linhas ou a superfície, para ter uma melhor visualização da figura. Na Figura 31 tem-se a visualização da superfície gerada, com alguns pontos de vista.

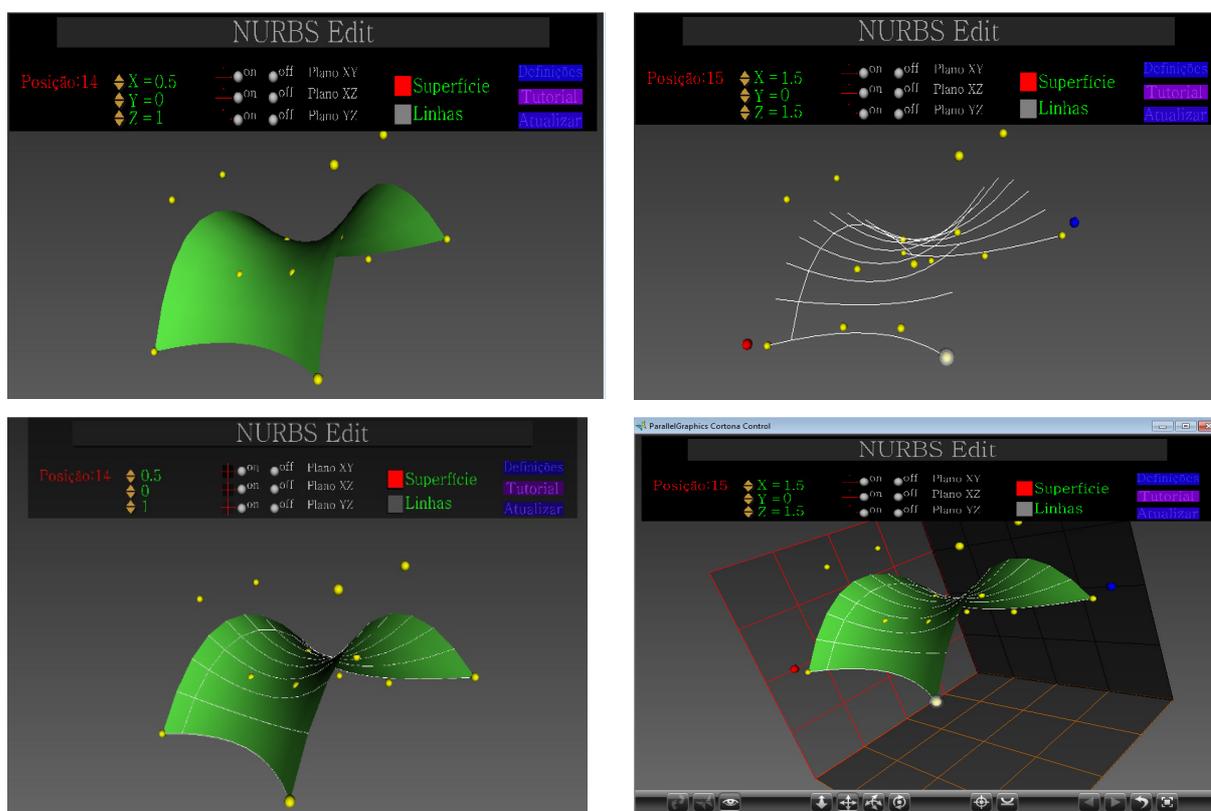


Figura 31: Sela gerada no software NURBS Edit com alguns pontos de vista.

Assim com interações simples com a utilização do mouse pode-se gerar uma infinidade de superfícies com o software NURBS Edit.

## 6.4 Avaliação do Sistema

O sistema desenvolvido foi avaliado através da aplicação de um questionário elaborado com base na ISONORM 9126-1 [23] que também considerou alguns aspectos de *software* educacional descritos por Campo e Campos (2001). O questionário aplicado foi respondido por professores e por alunos de graduação ao utilizarem o protótipo e se encontra no presente trabalho em Apêndice I.

O NURBS Edit foi apresentado a seis alunos da graduação do curso de Engenharia Elétrica da UFU – Universidade Federal de Uberlândia e a sete professores da UFU e do IFTM – Instituto Federal do Triângulo Mineiro, campus Uberlândia, considerados como usuários em potencial. Primeiramente foram fornecidas informações sobre o objetivo do projeto e, em seguida, de forma individual, eles iniciaram o uso do sistema. Após a execução do sistema, responderam o questionário de avaliação. Os resultados obtidos foram descritos a seguir e analisados por itens questionados.

### 6.4.1 Questões sobre a Funcionalidade

Após apresentação do objetivo proposto pelo trabalho e desenvolvimento do sistema protótipo foi-se questionado sobre a funcionalidade do software em relação à utilização no processo educacional. O resultado pode ser verificado no Gráfico 1.

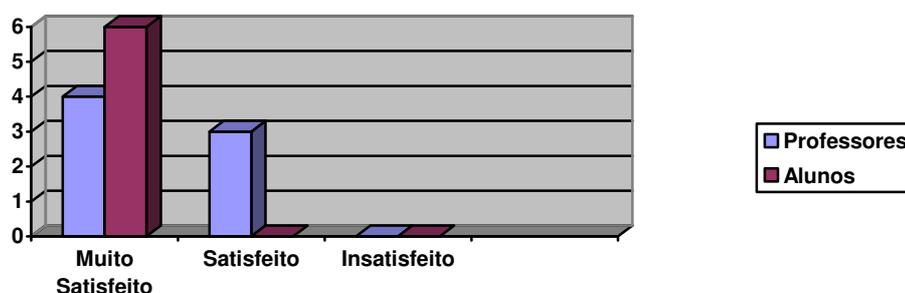


Gráfico 1: Resultado sobre a pergunta: O software auxilia no processo de aprendizagem na finalidade proposta?

Observa-se no gráfico 1, que tanto professores quanto os alunos concordaram que o conjunto de funções atende às necessidades explícitas e implícitas que se destina o *software* em auxiliar no processo de aprendizagem das superfícies NURBS. Todos os alunos se mostraram “muito satisfeitos” e os professores mostraram-se “muito satisfeitos” e “satisfeitos”.

Outro item avaliado foi sobre a funcionalidade global do *software* em relação à utilização no processo educacional. O resultado pode ser verificado no Gráfico 2.

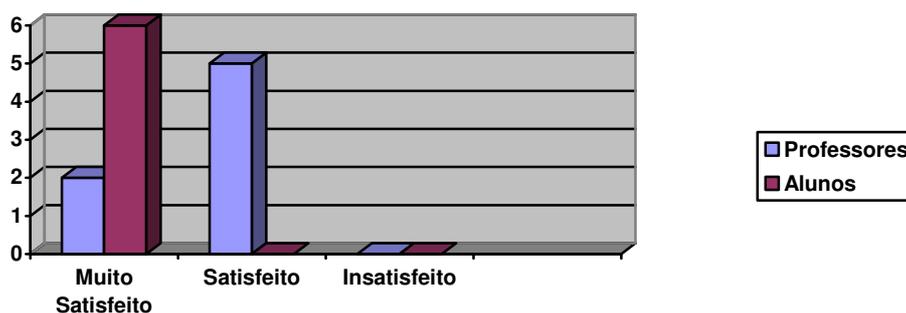


Gráfico 2: Resultado obtido sobre a pergunta: Como avalia globalmente a funcionalidade do software?

Nota-se no gráfico 2 que no geral o software foi avaliado satisfatoriamente quanto a sua funcionalidade pois tanto alunos quanto professores o avaliaram como “satisfeito” e “muito satisfeito”.

#### 6.4.2 Questões sobre a Usabilidade

Após apresentação do objetivo proposto pelo trabalho e desenvolvimento do sistema protótipo foi-se questionado sobre a usabilidade do software em relação à utilização no processo educacional. O resultado pode ser verificado no Gráfico 3.

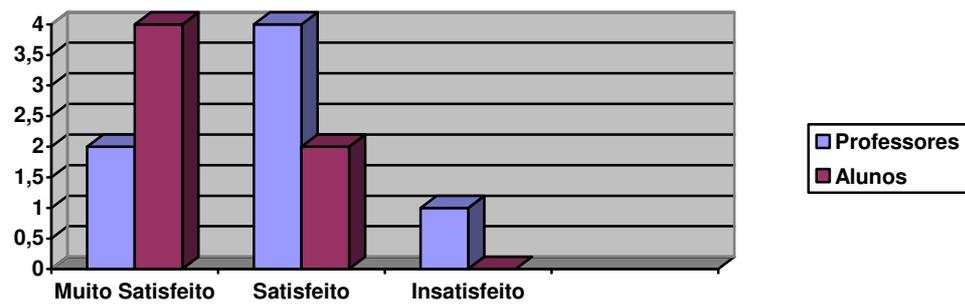


Gráfico 3: Resultado obtido sobre a pergunta: Em relação à facilidade de entender, como funciona o programa?

Quanto à facilidade de entender como funciona o programa, notou-se no gráfico 3 que a maioria ficou, “satisfeito” e “muito satisfeito”, tanto entre alunos quanto entre os professores. Entretanto, entre os professores, houve alguns que classificaram a facilidade de entender o programa como “insatisfeito”. Os professores que responderam “insatisfeito”, não justificaram suas opções quando ao entendimento à facilidade de utilização do *software*.

Outro item avaliado foi sobre a manipulação inicial das superfícies NURBS em relação à utilização no processo educacional. O resultado pode ser verificado no Gráfico 4.

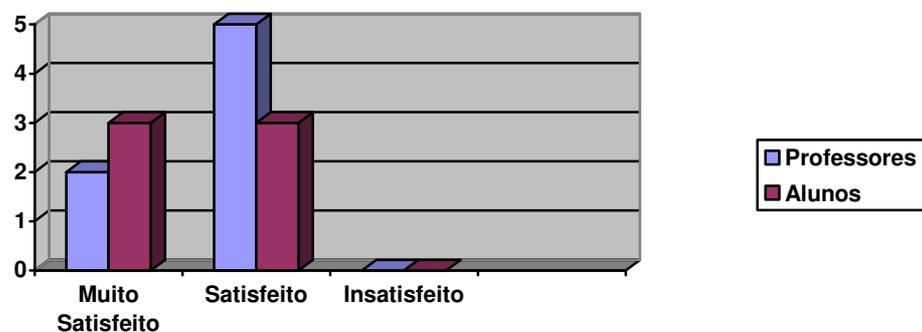


Gráfico 4: Resultado obtido sobre a pergunta: Quanto à manipulação da superfície NURBS inicial?

De acordo com os usuários a manipulação da superfície NURBS inicial é “satisfeito” e “muito satisfeito”, destacando-se a avaliação dos professores como “satisfeito”. Sobre os avaliadores, alguns fizeram observações quanto a opção de selecionar os pontos, pois os mesmos necessitam de um clique para selecionar a posição e um segundo clique para realizar as interações, portanto, alegaram que seria mais fácil, se com um clique já fossem realizadas as interações. Estes avaliaram a manipulação como “insatisfeito”.

Mais um item foi avaliado sobre a usabilidade global do software em relação à utilização no processo educacional. O resultado pode ser verificado no Gráfico 5.

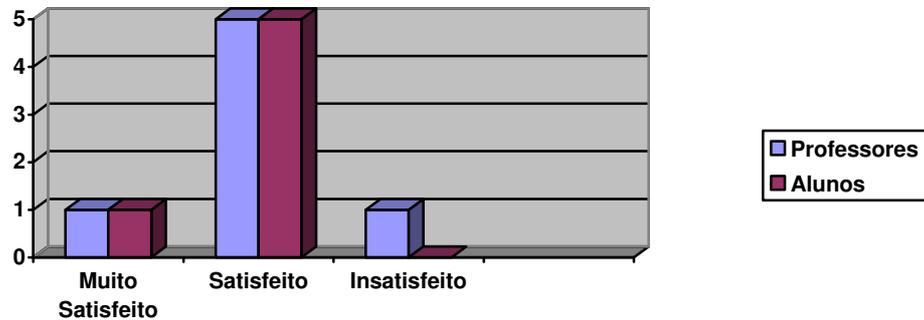


Gráfico 5: Resultado obtido sobre a pergunta: Como avalia globalmente a usabilidade do software?

Observa-se no gráfico 5 que a maioria dos usuários não encontrou dificuldades em usar o software no todo, classificando a usabilidade global com “satisfeito”. Porém, um dos usuários marcou como “insatisfeito” a usabilidade do software e justificou que o sistema precisa ser melhorado sobre a integração dos ícones do Cortona com os do ambiente, fazendo com que os ícones do Cortona se sobreponham aos comandos do Ambiente Virtual. Poucos foram os avaliadores que marcaram “muito satisfeito” com a usabilidade do software.

#### 6.4.3 Questões sobre a Confiabilidade

Após apresentação do objetivo proposto pelo trabalho e desenvolvimento do sistema protótipo foi-se questionado sobre a confiabilidade do software em relação à utilização no processo educacional. O resultado pode ser verificado no Gráfico 6.

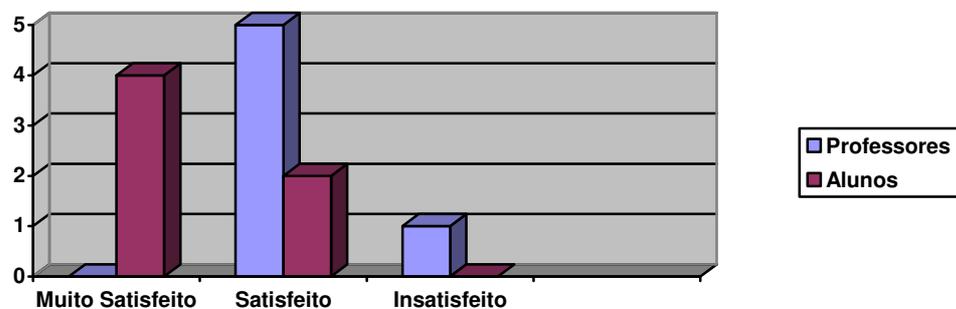


Gráfico 6: Resultado obtido sobre a pergunta: Capacidade de continuar a funcionar corretamente, após erros do próprio software ou erros de manipulação de dados?

Quanto à capacidade do software de continuar a funcionar corretamente, após erros do próprio software ou erros de manipulação de dados, observa-se no gráfico 6 que a maioria dos professores marcaram “satisfeito” com o software e a maioria dos alunos marcaram “muito satisfeito”. Parte dos Professores avaliaram o software como “insatisfeito” e não justificaram esta avaliação.

Outro item avaliado foi sobre a confiabilidade global do software desenvolvido. Os resultados podem ser verificados no gráfico 7.

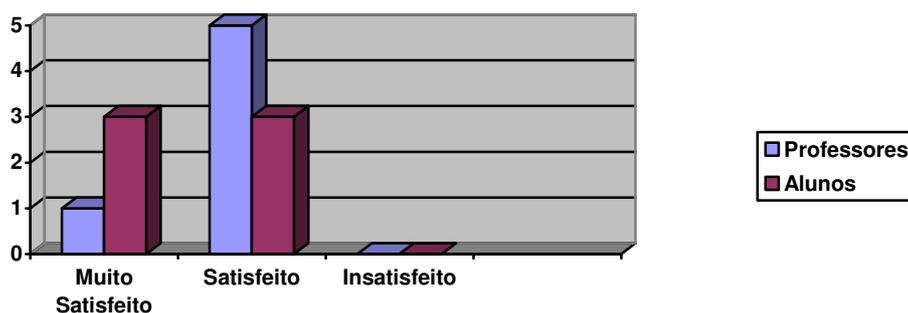


Gráfico 7: Resultado obtido sobre a pergunta: Como avalia globalmente a confiabilidade do software.

Tanto professores quanto os alunos avaliaram a confiabilidade do software como “satisfeito”. Dentre os avaliadores, foram os professores que mais avaliaram o software como “satisfeito” e dentre os alunos foram os que mais avaliaram o programa como “muito satisfeito”, como mostrou o gráfico 7. Tal resultado corresponde ao desempenho do sistema que se mantém ao longo do tempo em condições estabelecidas como satisfatório.

#### 6.4.4 Questões sobre a Eficiência

Após apresentação do objetivo proposto pelo trabalho e desenvolvimento do sistema protótipo foi-se questionado sobre a eficiência do software em relação à utilização no processo educacional. O resultado pode ser verificado no Gráfico 8.

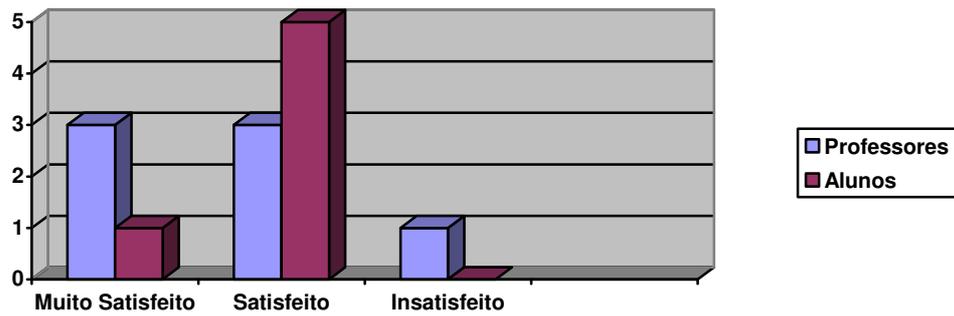


Gráfico 8: Resultado obtido sobre a pergunta: Tempo de geração de uma superfície NURBS

Como visualizado no gráfico 8, o tempo de geração de uma superfície NURBS no ambiente foi avaliado pela maioria dos alunos como sendo “satisfeito”. Na avaliação pelos professores, o tempo de geração das superfícies NURBS foi “satisfeito” e “muito satisfeito”, entretanto, houve uma avaliação como “insatisfeito”, o qual não justificou a sua opção.

Outra avaliação sobre a eficiência do software foi em termos globais como pode-se verificar no gráfico 9, o resultado.

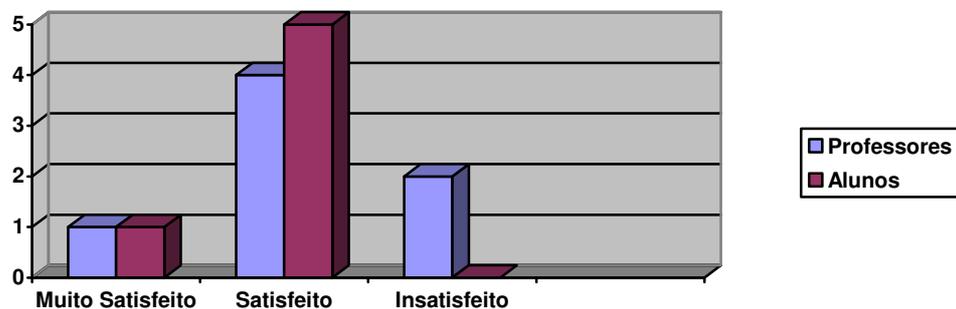


Gráfico 9: Resultado sobre a pergunta: Como avalia em termos globais a Eficiência do software.

Em termos globais a eficiência do software foi apresentada pelo gráfico 9 como “satisfeito” tanto para alunos quanto para professores. Foi obtido um mesmo resultado entre os alunos e professores como “muito satisfeito” em relação á eficiência global do programa. Porém, houve avaliações significativas por parte dos professores sobre a eficiência global relacionada ao ensino, que classificaram como “insatisfeito”, os quais apresentaram as justificativas: que fosse apresentado e demonstrado no ambiente, aplicações práticas do mundo real e a possibilidade de o usuário desfazer e refazer interações no próprio ambiente virtual.

### 6.4.5 Questões sobre o Sistema

Após apresentação do objetivo proposto pelo trabalho e desenvolvimento do sistema protótipo foi-se questionado sobre o sistema em relação à utilização no processo educacional. O resultado pode ser verificado no Gráfico 10.

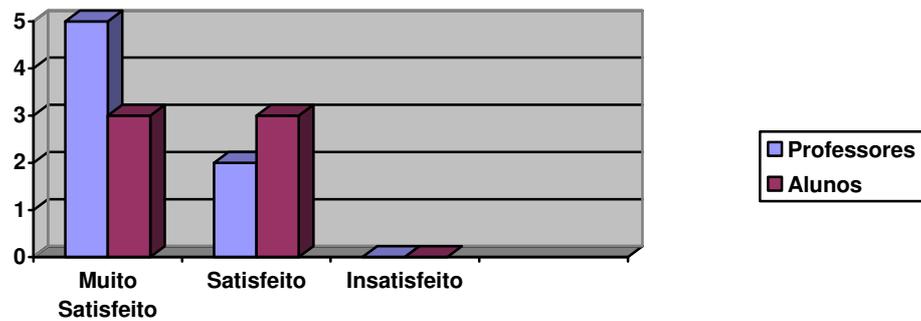


Gráfico 10: Resultado sobre a pergunta: A importância que atribui ao software para o aprendizado em geral.

De acordo com o gráfico 10, percebe-se que tanto professores quanto alunos avaliaram o software como “muito importante” e “importante” para a aprendizagem em geral. Observa-se como um excelente resultado obtido pela grande maioria dos professores ter avaliado o programa como “muito importante” em contribuição para o aprendizado.

Outro item avaliado sobre o sistema foi a importância em manipular as superfícies NURBS em um ambiente Virtual após a apresentação do objetivo do presente trabalho. O resultado dessa avaliação pode ser verificado no gráfico 11.



Gráfico 11: Resultado sobre a pergunta: A importância em manipular NURBS em um ambiente virtual.

A grande maioria dos professores avaliadores classificou como “muito importante” a manipulação de NURBS nos ambientes virtuais. De forma equilibrada os alunos classificaram

a manipulação de NURBS como sendo “muito importante” e “importante”. É possível afirmar com tal resultado entre professores e alunos, se deve a um conhecimento prévio e mais abrangente adquirido em estudos anteriores ou experiências de vida na educação que dispõem o professor em detrimento aos alunos.

Foi avaliado ainda, a importância geral do sistema em disciplinas da área de conhecimento dos avaliadores em que poderiam ser utilizadas as superfícies NURBS. O resultado pode ser verificado no gráfico 12.

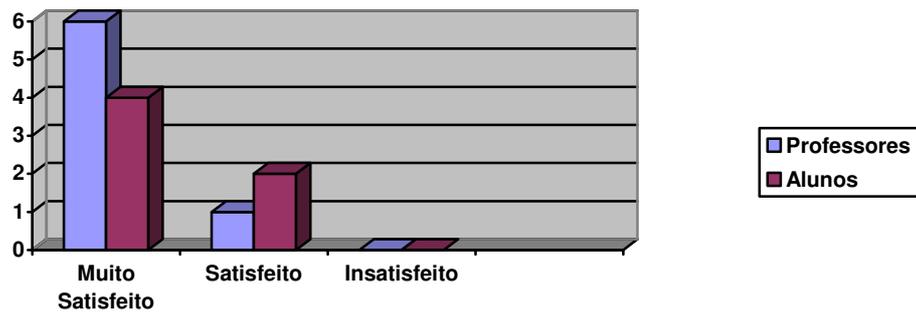


Gráfico 12: Resultado sobre a pergunta: A importância geral do sistema em disciplinas correlatas.

Quanto à importância do sistema em disciplinas correlatas, no gráfico 12 mostrou-se que entre os alunos avaliadores houve uma correspondência entre os resultados obtidos, os quais consideraram o sistema como “muito importante” e “importante”. Verificou-se que a maioria dos professores entenderam ser “muito importante” o uso do sistema em disciplinas que ministram. Tal fato pode ser atribuído à experiência profissional e entendimento maior sobre o assunto objeto desse trabalho.

#### 6.4.6 Questões sobre Avaliação de Desempenho

Após apresentação do objetivo proposto pelo trabalho e desenvolvimento do sistema protótipo foi-se questionado sobre o desempenho do sistema em relação à utilização no processo educacional. O resultado pode ser verificado no Gráfico 13.

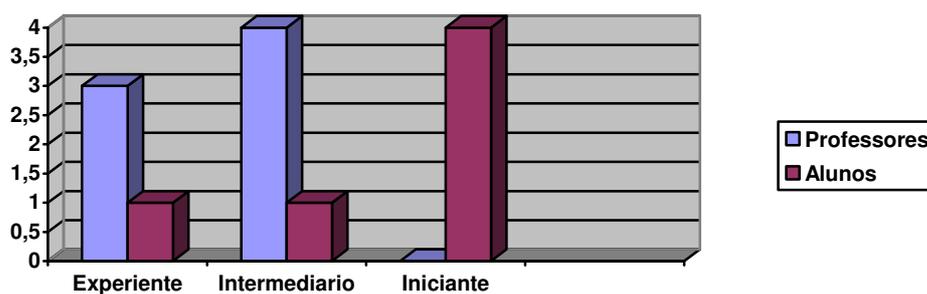


Gráfico 13: Resultado sobre a pergunta: Qual seu nível de conhecimento de NURBS antes da utilização do sistema?

Nota-se no gráfico 13 que a maioria dos professores disse ter pouco conhecimento em relação à edição de superfícies NURBS antes de conhecerem o sistema e se auto-classificaram como “iniciantes” e “intermediários”. Entre os alunos que responderam ao questionamento, muitos disseram serem “iniciantes”, e poucos auto-intitularam como terem um conhecimento “intermediário” e “experiente”.

Outro fator levado em consideração sobre a avaliação de desempenho do sistema que complementa o questionamento anterior foi em relação à experiência em edição de superfícies NURBS. O resultado pode ser verificado no gráfico 14.

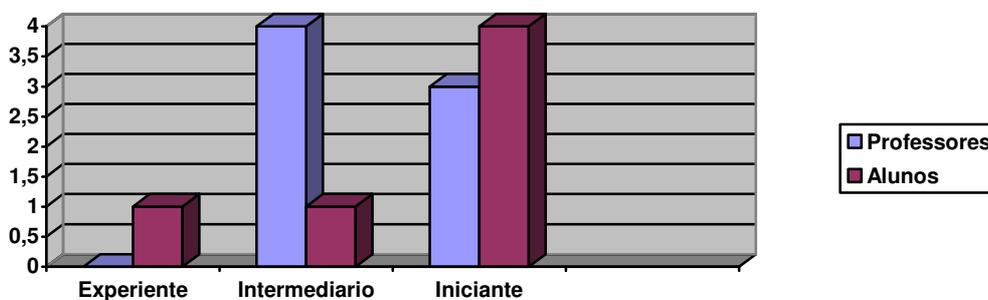


Gráfico 14: Resultado sobre a pergunta: Qual a sua experiência em edição de NURBS?

De forma coerente, obteve-se a correspondência entre o resultado obtido no gráfico 14 com os resultados do gráfico 13. Notou-se que os mesmos que responderam ter pouca, média ou elevada experiência anterior ao uso do sistema com superfícies NURBS, mantiveram seus pareceres em relação à experiência com NURBS.

Foi avaliado outro item em relação a avaliação de desempenho referente ao nível de conhecimento depois de utilizar o sistema desenvolvido com superfície NURBS. O resultado pode ser verificado no gráfico 15.

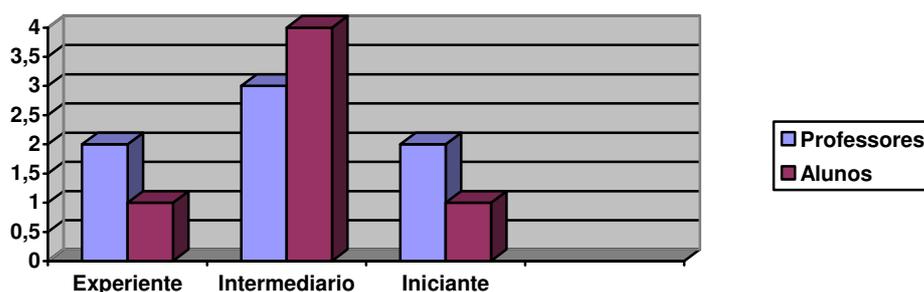


Gráfico 15: Resultado sobre a pergunta: Qual seu nível de conhecimento depois de usar o software?

Observou-se no gráfico 15 que ao responderem a pergunta sobre seu nível de conhecimento depois de usar o software, alguns usuários conseguiram abstrair algum conhecimento em relação às NURBS. Pois, houve um decréscimo na quantidade de avaliadores que se intitularam “iniciante” e um aumento significativo na classificação em conhecimento “intermediário”. Entretanto, o resultado verificado no gráfico 15 em relação aos gráficos 13 e 14, está na proporção de professores que após utilizarem o sistema se auto-classificaram como “experientes”, ou seja, obtiveram algum conhecimento com a utilização do *software*.

A avaliação do item sobre o desempenho na experiência em editar superfícies NURBS depois de ser explicado o objetivo do presente trabalho e de se detalhar o funcionamento do programa, obteve-se o resultado mostrado no gráfico 16.

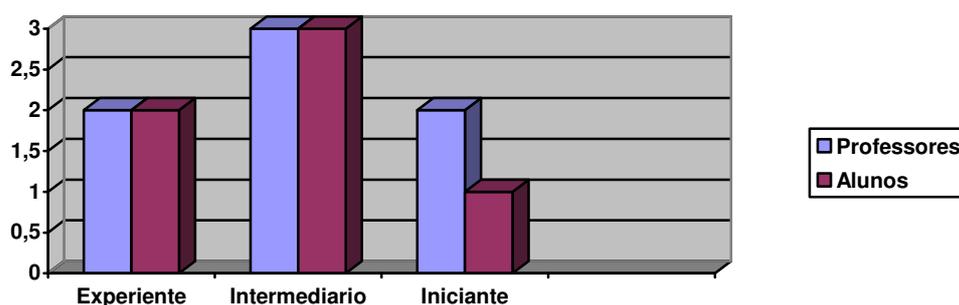


Gráfico 16: Resultados sobre a pergunta: Qual sua experiência em editar NURBS depois da explicação do sistema (disciplina).

Verificou-se no gráfico 16 a obtenção dos mesmos resultados do gráfico 15. Ou seja, que a experiência acompanha o conhecimento, ou que, o conhecimento gerado posteriormente à utilização do *software* gerou experiência em se editar superfícies NURBS tanto para alunos

quanto para professores, sendo mais expressivo e significativo o resultado obtido com a avaliação dos professores.

De acordo com os resultados obtidos nos gráficos gerados de 1 a 16, conclui-se que o software mostrou-se eficiente na elaboração de um protótipo que utilizou tecnologias computacionais para a visualização e manipulação de superfícies NURBS e na interação do usuário com o ambiente virtual desenvolvido para o ensino.

## 6.5 Comparação do software desenvolvido, NURBS Edit com outros programas.

Para se verificar a eficiência obtida nos resultados e tornar o NURBS Edit atrativo ao usuário, elaborou-se a Tabela 3 para compará-lo a outros softwares já desenvolvidos e testados para o ensino.

Tabela 3: Comparação entre os softwares criados para auxiliar o ensino/aprendizagem incluindo o NURBS Edit.

Sistemas Propriedades	VR Playground	Corte Visualização <sup>e</sup>	Poly Pro	Surf	NURBS Edit
AV Sem interação	X	X	✓	X	X
AV com Pouca Interação	✓	✓	X	X	X
AV com Muita Interação	X	X	X	✓	✓
AV nenhum Domínio de Objetos Virtuais	X	X	X	X	X
AV Pouco Domínio de Objetos Virtuais	✓	X	✓	X	X
AV Muito Domínio de Objetos Virtuais	X	✓	X	✓	✓
Visualização Intuitiva	X	X	X	X	✓

Analisando a tabela acima, pode-se perceber que o software NURBS Edit suporta os itens de Muita Interação e Muito Domínio de Objetos Virtuais e Visualização Intuitiva dos pontos de controle da superfície NURBS.

## **6.6 Limitações do Sistema**

Além das limitações apontadas pelos usuários nas sugestões descritas no sub-tópico 6.4, o sistema apresenta outras limitações. Como, por exemplo, o ambiente apresenta uma única superfície, ou seja, o ambiente poderia apresentar outras opções de superfícies NURBS como: esfera, cone, cubo, cilindro, etc. E, a partir dessas outras superfícies geométricas realizarem deformações.

Outra limitação observada foi a quantidade de pontos de controle sobre a superfície, o que limita uma aproximação mais detalhada dos objetos, durante a construção de alguns objetos mais elaborados.

## **6.7 Considerações Finais**

Este capítulo apresentou a avaliação do sistema NURBS Edit, elaborado para o ensino de superfícies NURBS, feito por educadores e alunos. Foram avaliados itens sobre a confiabilidade, usabilidade, funcionalidade, eficiência, o sistema propriamente dito e o desempenho do mesmo ao ser utilizado. Realizou-se a comparação do software NURBS Edit com outros programas desenvolvidos para o ensino e verificou-se que o sistema apresentou algumas características adicionais em relação aos sistemas analisados, o que melhora a sua eficiência em relação a aspectos de ensino/aprendizagem. Por fim, foram levantadas algumas limitações apresentadas pelo software após sua utilização e elaboração.

## CAPÍTULO 7 – CONCLUSÕES E TRABALHOS FUTUROS

### 7.1 Conclusões

Este trabalho teve como objetivo, investigar técnicas de Realidade Virtual para possibilitar ao usuário interagir com um ambiente que suporte manipulação de NURBS. Como resultado, o sistema NURBS Edit foi elaborado usando tecnologias computacionais acessíveis (gratuitas e disponíveis no domínio público) como: linguagens VRML, Java Script, software White\_dune e Cortona. Estas possibilitaram a elaboração de um ambiente para edição de superfície NURBS com visualização clara e possível de ser manipulada. As principais conclusões deste trabalho são:

- Uma vez identificado uma ferramenta com recursos para modelagem de NURBS, como o White-Dune, é possível criar um ambiente virtual que suporte o ensino das mesmas de forma interativa e natural;
- O sistema de edição de NURBS criado, por contemplar técnicas de Realidade Virtual tais como imersão e navegação, proporcionou um ambiente de aprendizagem satisfatório uma vez que deu ao usuário a condição de aprendiz ativo (e não passivo). Esta característica proporciona um melhor condicionamento para o ensino;
- A fim de suportar interação e navegação de qualidade num ambiente virtual como o NURBS Edit, é necessário providenciar a do VRML com o JavaScript. Isto porque o VRML dispõe de recursos para realizar as interações necessárias;

Professores e alunos do ensino superior que experimentaram as técnicas de RV propostas avaliaram as mesmas como satisfatórias – o que conduz à conclusão de que é sempre importante explorar estas características da Realidade Virtual para providenciar um ambiente mais amigável de aprendizagem.

## 7.2 Trabalhos Futuros

Como trabalhos futuros, sugere-se:

- A criação de outras superfícies NURBS para que o usuário tenha a opção de escolher qual superfície deseja editar;
- Investigar técnicas computacionais que possibilitem o corte nas superfícies editadas;
- Adotar o NURBS Edit no ensino superior a fim de identificar mais limitações e aprofundar a pesquisa sobre como explorar as técnicas de RV na busca de criação de ambientes mais qualificados pedagogicamente para o ensino.

## REFERÊNCIAS

- [1] TORI, R.; KIRNER, C.; SISCOOTTO, R.; Fundamentos e Tecnologia de Realidade Virtual e Aumentada, Livro do Pré-Simpósio, VIII Symposium on Virtual Reality, 2006. p. 2
- [2] VENDRUSCOLO, F. e Outros; Escola TRI – Legal – Um Ambiente Virtual como Ferramenta de Apoio ao Ensino Fundamental Através de Jogos Educacionais. Colabora – A revista digital da CVA-RICESU – Volume 3 – nº 9, 2005. p 4
- [3] CARDOSO, A.; LAMOUNIER, E.; Aplicações de RV e RA na Educação e Treinamento. XI Symposium on Virtual and Augmented Reality, 2009, p.103.
- [4] ROUSSOU, M., A VR Playground for Learning Abstract Mathematics Concepts, Makebelieve Design & Consulting ,January and February, 2009. p. 82-85
- [5] BRAGA, M.; Realidade Virtual e Educação, Revista de Biologia e Ciências da Terra, Volume 1, 2001. p. 1-3
- [6] TORI, R.; KIRNER, C.; SISCOOTTO, R.; Fundamentos e Tecnologia de Realidade Virtual e Aumentada, Livro do Pré-Simpósio, VIII Symposium on Virtual Reality, 2006. p. 304-312
- [7] FÉLIX, L.R.; SILVA, A. B. A. e FÉLIX, N. M.R.; Entre Béziers e NURBS: Ensino de Formas Livres no Contexto Arquitetônico. Universidade Federal de Pelotas, 2004, SIGRADI-UNISINOS. 1-3 p.

[8] ROGERS, D. F.; An Introduction to NURBS With Historical Perspective. Annapolis. Aerospace Engineering United States Naval Academy.

[9] MINETTO, C. F.; Um estudo Sobre Curvas NURBS. Dissertação de Mestrado em Matemática Aplicada, UFRGS, Porto Alegre, 2003. p. 14.

[10] PIEGL, L.; TILLER, W.; The NURBS Book. Florida: University of South Florida, 2ª Edição, 1997.

[11] FORTI, F. S. D., Apostila Rhinoceros, 2008. p.1-2.

[12] LEAKE, J.M., Development of an Advanced Course in Computer-Aided Design, Analysis and Prototyping, 7274\_JL\_curriculum\_overview.doc, 2004. p. 4-9

[13] WHITE\_DUNE (2011a) Disponível em:

[http://vrml.cip.ica.uni-stuttgart.de/dune/docs/usage\\_docs/dune\\_en.html#fenste](http://vrml.cip.ica.uni-stuttgart.de/dune/docs/usage_docs/dune_en.html#fenste).

Acesso em: 06 ago. 2011

[14] WHITE\_DUNE (2011b) Disponível em:

[http://129.69.35.12/dune/docs/usage\\_docs/dune\\_en\\_tutorial.html](http://129.69.35.12/dune/docs/usage_docs/dune_en_tutorial.html). Acesso em: 06 ago. 2011

[15] BENTO, J. J. F.; Desenvolvimento e avaliação de um ambiente de aprendizagem 3D. Dissertação Mestrado em Tecnologias de Informação e Comunicação na Educação e Formação. Bragança, 2011. p.24.

[16] NETO, A.F. M. Software [livre] na arte computacional. Brasília, 2010. p.33.

[17] ARAUJO, J.O.; Desenvolvimento de Jogos 3D com Ferramentas Gratuitas. Santos, 2010. p. 35.

[18] VIANA, T. C. Projeto e desenvolvimento de um jogo de futebol utilizando o motor de jogo Unity 3D. São Carlos, 2009. p.10.

[19] CARDOSO, A.; LAMOUNIER, E.; Livro VRML SVR03. VI Symposium on Virtual Reality. Ribeirão Preto – SP, 2003, p.1-3.

[20] CARDOSO, A.; LAMOUNIER, E.; Realidade Virtual – Uma Abordagem Prática. Minicursos SVR 2004, VII Symposium on Virtual Reality. São Paulo – SP, 2004, p.48-50.

[21] ROSSO, D. H. S.; ROJAS, S. G.; Livro Mundos Virtuaes 3D Com VRMR97, Universidad de Málaga. P. 112.

[22] SILVA, M. E.; Uma Aplicação das técnicas de Realidade Virtual na Visualização e Corte de Poliedros Não-Convexos. Dissertação Mestrado em Ciências. Uberlândia, 2007. p.27.

[23] ISONORM 9126-1 disponível em: <<http://www.sqa.net/iso9126.html>> acesso em 02 set 2012

[24] ESPATH, L. F. R.; Otimização de forma de Cascas Via Deformação Livre de Forma Baseada em NURBS. Dissertação de Mestrado em Engenharia, UFRGS, Porto Alegre, 2009, p. 22.

[25] NAKAMOTOA, P.T., TAKAHASHI, E.; MENDES, E.; CARDOSO, A. e LAMOUNIER JR, E.; O Uso de Mapas Conceituais e Realidade Virtual para o Ensino de Física no Ensino Médio; IX Encontro Nacional de Pesquisa em Ensino de Física, 2005.

## ANEXO I

### 1.0 Introdução

As duas formas de se representar analiticamente curvas, superfícies e sólidos são funções implícitas ou funções paramétricas. A função implícita de uma curva contida no plano “ $xy$ ” toma a forma  $f(x,y) = 0$ . Esta equação descreve uma relação implícita entre as quantidades  $x$  e  $y$ . Na forma paramétrica, cada quantidade é representada de forma separada na forma de função explícita em função de um, dois ou três parâmetros independentes, para curvas, superfícies e sólidos, respectivamente [24].

### 1.1 Funções Paramétricas

Para se obter um conjunto de pontos com variáveis cartesianas de forma a não estipular uma relação obrigatória entre as mesmas é conseguida, com uma variável a parte, que define os valores das coordenadas, mas não é utilizada diretamente no gráfico, esta variável é chamada de parâmetro.

Assim, vejamos a seguir, uma função simples, usando parâmetro:

Seja a equação  $x = y^2$ , pode-se observar que a mesma não pode ser representação de uma função se tomarmos  $x$  como abscissa, assim tem-se mais de um valor possível para  $y$ , o que fazemos neste caso é dizer que  $y = u$  e  $x = u^2$ . ou seja:

$u$  parâmetro (variável independente)

$y = u$  (variável dependente)

$x = u^2$  (variável dependente).

A representação de um ponto na curva, em sua forma paramétrica, se dá pela função explícita e com parâmetros independentes

$$C(u) = (x(u), y(u)) \quad a \leq u \leq b \quad \text{Eq. (1.0)}$$

Com  $C(u)$  uma função vetorial com sua variável independente,  $u$ . No intervalo  $[a,b]$  qualquer.

Assim, adicionando  $z$  na forma paramétrica 1.0, tem-se uma função paramétrica de uma curva no espaço:

$$\mathbf{C}(u) = (\mathbf{x}(u), \mathbf{y}(u), \mathbf{z}(u)) \quad \mathbf{a} \leq u \leq \mathbf{b} \quad \text{Eq. (1.1)}$$

Logo,  $\mathbf{C}(u)$  é uma função vetorial de uma variável independente,  $u$ . Para superfícies são duas variáveis independentes e para sólidos, três variáveis independentes, resultando nas equações:

$$\mathbf{S}(u,v) = (\mathbf{x}(u,v), \mathbf{y}(u,v), \mathbf{z}(u,v)) \quad (u,v) \in [\mathbf{a},\mathbf{b}] \times [\mathbf{c},\mathbf{d}] \quad \text{Eq. (1.2)}$$

$$\mathbf{G}(u,v,w) = (\mathbf{x}(u,v,w), \mathbf{y}(u,v,w), \mathbf{z}(u,v,w)) \quad (u,v,w) \in [\mathbf{a},\mathbf{b}] \times [\mathbf{c},\mathbf{d}] \times [\mathbf{e},\mathbf{f}] \quad \text{Eq. (1.3)}$$

Assim temos que para a forma parametrizada existem infinitas parametrizações que descrevem analiticamente a mesma geometria [24].

## 1.2 – Bézier

As curvas de Bézier de  $n$ -ésimo grau, apresenta a seguinte equação:

$$\mathbf{C}(u) = \sum_{i=0}^n B_{i,n}(u) P_i \quad u \in [0,1] \quad \text{Eq. (1.4)}$$

Onde as funções de base (*blending*),  $\{B_{i,n}\}$ , são os clássicos polinômios de *Bernstein* de  $n$ -ésimo grau, divulgados por Bernstein (1912), que são dados por:

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \quad \text{Eq. (1.5)}$$

Onde,  $\frac{n!}{i!(n-i)!}$  é o número ou coeficiente binomial e os coeficientes geométricos,  $\{P_i\}$ , são chamados de pontos de controle, pois definem e controlam a geometria parametrizada [21]

## 1.3 – Coordenadas homogêneas

Através das Coordenadas Homogêneas podemos introduzir o conceito de infinito na computação. Considerando dois números reais,  $a$  e  $w$ , e calculando  $a/w$  é possível fixar o valor de  $a$  e variar o valor de  $w$ . Assim temos que se  $w$  diminui o valor de  $a/w$  aumenta; então se  $w$  tende a zero o valor de  $a/w$  tende a infinito. Deste modo, o conceito de infinito para ser

utilizado são empregados dois números  $a$  e  $w$ , os quais representam um valor  $v$ ,  $v = a/w$ . Se  $w$  não é zero, o valor é  $a/w$ . Caso contrário, identificamos o valor de infinito como  $(a, 0)$ . Portanto o conceito de infinito pode ser representado como um par ordenado de números  $(a, w)$  ou como um quociente  $a/w$ .

Utilizando o conceito de infinito para o plano de coordenadas  $xy$ , podemos substituir  $x$  e  $y$  por  $x/w$  e  $y/w$ , tal que uma função  $f(x,y) = 0$  torna-se  $f(x/w,y/w) = 0$ . Se a função  $f(x,y) = 0$  é um polinômio de grau  $u$ , multiplicando-o por  $w^n$  os denominadores serão eliminados. Por exemplo, tomemos uma reta  $Ax + By + C = 0$ . substituindo  $x$  e  $y$  por  $x/w$  e  $y/w$ , temos  $A(x/w) + B(y/w) + C = 0$ . multiplicando por  $w$ , temos:

$$Ax + By + Cw = 0 \quad \text{Eq. (1.6)}$$

Tomemos um polinômio de grau 2,  $Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0$ , substituindo  $x$  e  $y$  por  $x/w$  e  $y/w$  e multiplicando por  $w^2$ , temos:

$$Ax^2 + 2Bxy + Cy^2 + 2Dxw + 2Eyw + Fw^2 = 0 \quad \text{Eq. (1.7)}$$

Tomando os polinômios 1.6 e 1.7 constatamos que os graus de todas as condições são iguais. No caso da reta,  $x$ ,  $y$  e  $w$  são de grau 1, e no polinômio de grau 2, as condições,  $x^2$ ,  $xy$ ,  $y^2$ ,  $xw$ ,  $yw$ ,  $w^2$ , são de grau 2.

Portanto, para um polinômio de grau  $n$ , depois de aplicar  $w$ , todas as condições passam para grau  $n$ . Assim estes polinômios são chamados de polinômios homogêneos e as coordenadas  $(x, y, w)$  são chamadas de coordenadas homogêneas [19].

## 1.4 – Bézier racional

Através dos conceitos das seções 1.2 e 1.3, chegamos ao conceito de funções racionais que podem ser escritas na forma:

$$x(u) = \frac{X(u)}{W(u)}; \quad y(u) = \frac{Y(u)}{W(u)}; \quad z(u) = \frac{Z(u)}{W(u)} \quad \text{Eq. (1.8)}$$

Em que  $X(u)$ ,  $Y(u)$ ,  $Z(u)$  e  $W(u)$  são polinômios, e cada coordenada tem denominador comum. Definindo assim uma curva Bézier Racional de  $n$ -ésimo grau por:

$$c(u) = \frac{\sum_{i=0}^n B_{i,n}(u) w_i P_i}{\sum_{i=0}^n B_{i,n}(u) w_i}, \quad u \in [0,1] \quad \text{Eq. (1.9)}$$

Onde,  $\{B_{i,n}\}$ , são os polinômios de *Bernstein* de  $n$ -ésimo grau, divulgados por Bernstein (1912),  $\{P_i\}$ , são chamados de pontos de controle e  $w_i$  são escalares, chamados de pesos.

Assim  $W(u) = \sum_{i=0}^n B_{i,n}(u) w_i$ ,  $W(u) > 0$  para todo  $u \in [0,1]$ . Assim escrevemos 1.9 da seguinte

forma:

$$C(u) = \sum_{i=0}^n R_{i,n}(u) P_i \quad u \in [0,1] \quad \text{Eq. (1.10)}$$

Onde

$$R_{i,n}(u) = \frac{B_{i,n}(u) w_i}{\sum_{j=0}^n B_{j,n}(u) w_j} \quad \text{Eq. (1.11)}$$

As  $R_{i,n}(u)$  são as funções de base racionais [24].

## 1.5 – Funções de base B-spline

Há muitas maneiras de definir as funções de base B-spline, a mais conhecida foi apresentada por deBoor (1972) e por Cox (1972), definição Recursiva.

Seja  $U = \{u_0, \dots, u_m\}$  uma sequência de números reais crescentes, isto é,  $u_i \leq u_{i+1}$ ,  $\forall i \in \mathbb{N}$ . Os  $u_i$  são os nós (*knots*), e  $U$  é o vetor de nós (*knots vector*). A  $i$ -ésima função de base B-spline de grau  $p$  (ordem  $p + 1$ ), denotado por  $N_{i,p}(u)$ , é definida como:

$$N_{i,0}(u) = \begin{cases} 1 & \text{se } u_i \leq u \leq u_{i+1} \\ 0 & \text{caso contrario} \end{cases} \quad \text{Eq. (1.12)}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad \text{Eq. (1.13)}$$

Note que:

$N_{i,0}(u)$  é uma função de grau unitário, igual a zero em todo o seu domínio, exceto no intervalo semi-aberto  $u \in [u_i, u_{i+1})$ ;

Para  $p > 0$ ,  $N_{i,p}(u)$  é uma combinação linear de duas funções de base de grau  $(p-1)$ ;

O calculo de um conjunto de funções de base requer a especificação do vetor de nós,  $U$ , e do grau  $p$ ;

A expressão (1.13) pode produzir o quociente 0/0 (zero sobre zero); e este é definido como zero;

As  $N_{i,p}(u)$  funções polinomiais por partes estão definidas no conjunto dos reais; em geral somente interessam no intervalo semi-aberto  $[u_0, u_m)$ ;

O intervalo semi-aberto  $[u_i, u_{i+1})$ , é chamado de período do  $i$ -ésimo nó; este pode ter comprimento zero já que os nós podem não ser distintos;

O computo das funções de base geram uma tabela triangular truncada.

Tabela 1.2: tabela triangular truncada das  $N_{i,p}(u)$ .

$$\begin{array}{cccc}
 & & & N_{0,0} \\
 & & & \\
 & & & N_{0,1} \\
 & & N_{1,0} & N_{0,2} \\
 & & N_{1,1} & N_{0,3} \\
 & N_{2,0} & N_{1,2} & \\
 & & N_{2,1} & N_{1,3} \\
 & & N_{3,0} & N_{2,2} & \vdots \\
 & & & N_{3,1} & \vdots \\
 & & & & \vdots \\
 & N_{4,0} & & & \\
 & & & & \vdots
 \end{array}$$

Por simplificação  $N_{i,p}(u)$  pode ser escrito como  $N_{i,p}$  [20].

Sobre a nomenclatura, o termo pontos de parada (*breakpoints*) é usado quando  $u_i < u_{i+1}$ , isto é uma sequência monótona crescente, o termo vetor de nós (*knots vector*) é usado quando  $u_i \leq u_{i+1}$ , isto é uma sequência crescente. O termo B-spline significa que se está fazendo uso das funções de Bernstein ( $B$ ) por partes (*spline*).

Um caso particular das curvas e superfícies B-spline devem ser as curvas e superfícies Bézier. Esta particularização vem dada sempre que o vetor de nós vier dado na forma:

$$\mathbf{u} = \left\{ \underbrace{0, \dots, 0}_{p+1}, \underbrace{1, \dots, 1}_{p+1} \right\}, \quad \text{Eq. (1.14)}$$

sendo

$U = \{ u_0 = 0, u_1 = 0, u_2 = 0, u_3 = 1, u_4 = 2, u_5 = 3, u_6 = 4, u_7 = 4, u_8 = 5, u_9 = 5, u_{10} = 5 \}$  e  $p = 2$  as funções de base, não nulas, de grau zero, um e dois são apresentadas nas figuras 1.5.1, 1.5.2 e 1.5.3, a seguir.

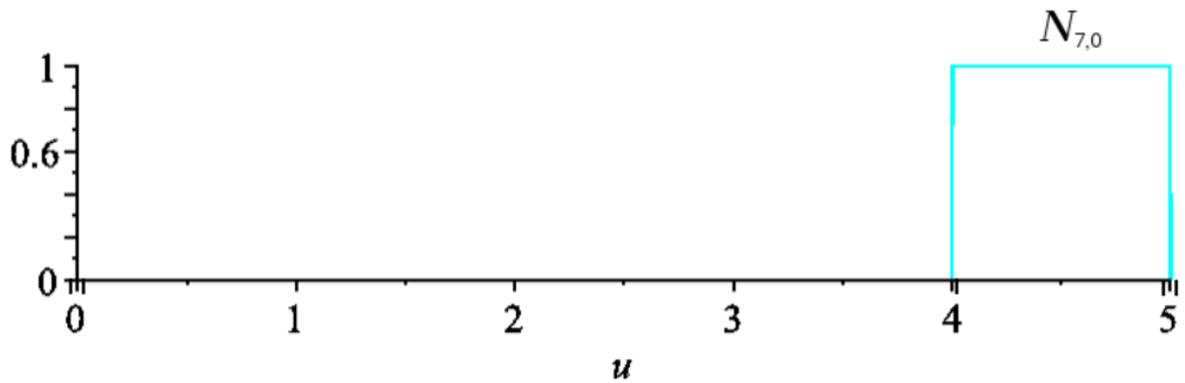


Figura 1.5.1: Funções de base não-nulas de zero grau.

Fonte: Espath, Luis F. R., *Otimização de forma de cascas via deformação livre de forma baseada em NURBS*

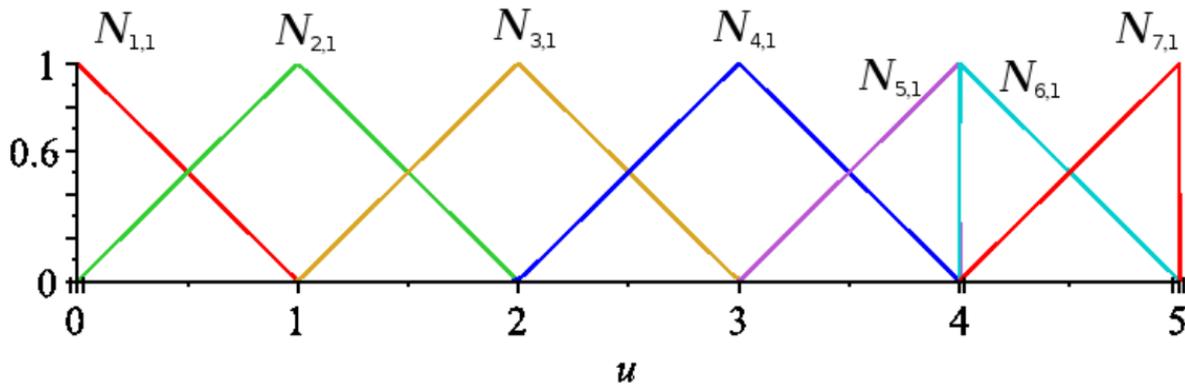


Figura 1.5.2: Funções de base não-nulas de primeiro grau.

Fonte: Espath, Luis F. R., *Otimização de forma de cascas via deformação livre de forma baseada em NURBS*

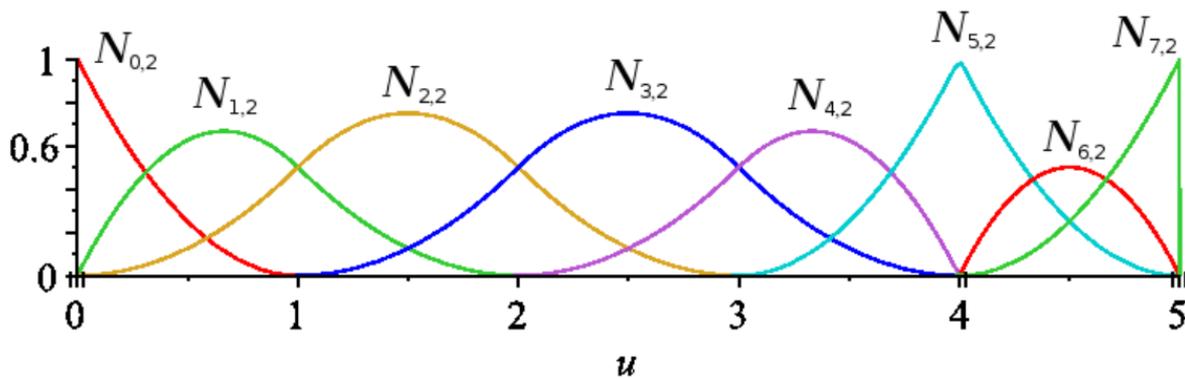


Figura 1.5.3: Funções de base não-nulas de segundo grau.

Fonte: Espath, Luis F. R., *Otimização de forma de cascas via deformação livre de forma baseada em NURBS*

## 1.6 – Curvas e superfícies B-spline

### 1.6.1 – Definição da curva B-spline

Uma curva B-spline de grau  $p$  é definida por:

$$C(u) = \sum_{i=0}^n N_{i,p}(u) P_i \quad a \leq u \leq b \quad \text{Eq. (1.15)}$$

Onde  $\{P_i\}$  são os pontos de controle e  $\{N_{i,p}(u)\}$  são as funções base B-spline de grau  $p$  definidas sobre o vetor de nós não-periódicos e não-uniformes

$$U = \left\{ \underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{b, \dots, b}_{p+1} \right\} \quad \text{Eq. (1.16)}$$

De  $(m+1)$  nós.

### 1.6.2 – Definição das superfícies B-spline

Uma B-spline é obtida através de uma rede bidirecional de pontos de controle, dois vetores de nós, e o produto das funções de base B-spline univariantes. Então temos:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) P_{i,j} \quad (u, v) \in [a, b] \times [c, d] \quad \text{Eq. (1.17)}$$

$$\text{Com } U = \text{Eq. 1.17 e } V = \left\{ \underbrace{c, \dots, c}_{q+1}, v_{g+1}, \dots, v_{s-g-1}, \underbrace{d, \dots, d}_{g+1} \right\}$$

$U$  possui  $r+1$  nós,  $V$  possui  $s+1$  nós. Então temos:

$$r = n + p + 1 \quad \square \quad s = m + q + 1 \quad \text{Eq. (1.18)}$$

## 1.7 – NURBS – Non Uniform Rational B-Spline

### 1.7.1 – Definição

As NURBS são definidas como uma curva de grau  $p$ :

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u) w_i P_i}{\sum_{i=0}^n N_{i,p}(u) w_i} \quad u \in [a,b] \quad \text{Eq. (1.19)}$$

Em que  $\{P_i\}$  são os pontos de controle,  $\{w_i\}$  são os pesos e as  $\{N_{i,p}(u)\}$  são as funções base B-spline de grau  $p$  definidas sobre um vetor de nós não-periódicos e não-uniformes

$$U = \left\{ \underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{b, \dots, b}_{p+1} \right\} \quad \text{Eq. (1.20)}$$

de  $(m + 1)$  nós.

Sem perda de generalidade, podemos assumir que  $a = 0$ ,  $b = 1$  e  $w_i > 0 \quad \forall i$ . Logo, tomando:

$$R_{i,p}(u) = \frac{N_{i,p}(u) w_i}{\sum_{i=0}^n N_{i,p}(u) w_i} \quad \text{Eq. (1.21)}$$

Assim podemos escrever 1.19 da seguinte forma:

$$C(u) = \sum_{i=0}^n R_{i,p}(u) P_i \quad \text{Eq. (1.22)}$$

As  $\{R_{i,p}(u)\}$  são funções base racionais; elas são funções racionais por partes sobre  $u \in [0,1]$ . As propriedades das  $R_{i,p}(u)$  são derivadas de (1.21) e das correspondentes propriedades das  $N_{i,p}(u)$ .

### 1.7.2 – Propriedades

Temos abaixo 14 propriedades das funções NURBS:

**P1** - Função positiva semi definida:  $R_{i,p}(u) \geq 0, \forall i,p \text{ e } u \in [0,1]$ ;

**P2** - Partição da unidade:  $\sum_{i=0}^n R_{i,p}(u) = 1, \forall u \in [0,1]$ ;

**P3** -  $R_{0,p}(0) = R_{n,p}(1) = 1$

**P4** - Toda função NURBS  $R_{i,p}(u)$  para  $p > 0$  possui um mínimo e um máximo no intervalo  $u \in [0,1]$ ;

**P5** - Suporte local:  $R_{i,p}(u) = 0$  para  $u \notin [u_i, u_{i+p+1})$ . Assim, em um período dado  $[u_i, u_{i+1})$  no máximo  $p+1$  das  $R_{i,p}(u)$  são não nulas, chamadas de  $R_{i-p,p}, \dots, R_{i,p}$ ;

**P6** - As derivadas de  $R_{i,p}(u)$  existem no interior do período, estas são funções racionais de denominador não nulo. Sobre o nó,  $R_{i,p}(u)$  é  $C^{p-k}$  contínuo, onde  $k$  é a multiplicidade do nó;

**P7** - Se  $w_i = 1$  para  $\forall i$ , então  $R_{i,p}(u) = N_{i,p}(u)$  para  $\forall i$ , então  $N_{i,p}(u)$  é um caso particular de  $R_{i,p}(u)$ ;

**P8** - De P3 temos  $C(0) = P_0$  e  $C(1) = P_1$ ;

**P9** - Transformações lineares: as transformações tais como rotação, translação, escala e cisalhamento aplicadas à curva são realizadas aplicando tal transformação ao polígono de controle;

**P10** - Região convexa: a curva esta contida na região convexa formada pelo polígono de controle; de fato, se  $u \in [u_i, u_{i+1})$ ,  $p \leq i \leq m - p - 1$ , então  $C(u)$  estará contida no polígono de controle  $P_{i-p}, \dots, P_i$ ;

**P11** -  $C(u)$  é infinitamente diferenciável no interior do intervalo de nós, e é  $C^{p-k}$  contínua sobre o nó de multiplicidade  $k$ ;

**P12** - Nenhuma reta ou plano, contém mais interseções com uma curva NURBS do que esta reta ou plano contém com o polígono de controle da curva;

**P13** - Uma curva NURBS sem os nós internos é uma curva Bézier racional, já que  $N_{i,p}(u)$  se convertem nas  $B_{i,n}(u)$ ;

**P14** - Esquema de modificação local: modificando  $P_i$  ou  $w_i$  a curva  $C(u)$  modifica sua forma somando no intervalo  $[u_i, u_{i+p+1})$ , que resulta da propriedade P5;

Assim a propriedades P14, dá a motivação necessária para todos os tipos de algoritmos de modificação de forma baseada em NURBS. A partir desta podemos alterar modificar o conjunto  $P_i$  e  $w_i$  sem alterar toda a entidade, más apenas um subdomínio [21]

### 1.7.3 – Definição das superfícies NURBS

As superfícies NURBS de grau  $p$  na direção  $u$  e de grau  $q$  na direção  $v$  é uma função vetorial bivalente racional por partes, da forma:

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j}} \quad (u, v) \in [0,1] \times [0,1] \quad \text{Eq. (1.23)}$$

Os  $\{P_{i,j}\}$  formam uma rede de controle bidirecional, os  $\{w_{i,j}\}$  são os pesos e as  $\{N_{i,p}(u)\}$  e  $\{N_{j,q}(v)\}$  são as funções de base B-spline não racionais definidas sobre o vetor de nós

$$U = \left\{ \underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{1, \dots, 1}_{p+1} \right\}$$

$$V = \left\{ \underbrace{0, \dots, 0}_{q+1}, v_{q+1}, \dots, v_{s-q-1}, \underbrace{1, \dots, 1}_{q+1} \right\}$$

Onde:  $r = n + p + 1$  e  $s = m + q + 1$

Introduzindo a funções de base racionais

$$R_{i,j}(u, v) = \frac{N_{i,p}(u) N_{j,q}(v) w_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j}} \quad (u, v) \in [0,1] \times [0,1] \quad \text{Eq. (1.24)}$$

Assim a superfície 1.23 pode ser reescrita como;

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,j}(u, v) P_{i,j} \quad \text{Eq. (1.25)}$$

## APÊNDICE I

Avaliação do Sistema NURBS Edit		
Avaliador:		Data Avaliação:
( ) professor ( ) aluno	Escolaridade: ( ) Ensino Médio Superior	( ) Ensino

Assinale, por favor, a opção que melhor traduz a sua opinião.

<b>1. FUNCIONALIDADE-</b> Evidência que o conjunto de funções atende às necessidades explícitas e implícitas para a finalidade que se destina o software <b>(Possibilita a manipulação de NURBS, permite realizar deformações na superfície gerada, para que o usuário possa interagir com o ambiente virtual).</b>	Muito Satisfeito	Satisfeito	Insatisfeito
1.1. O software auxilia no processo de aprendizagem na finalidade proposta			
1.2. Como avalia globalmente a funcionalidade do software.			

<b>2. USABILIDADE-</b> Evidência a facilidade de utilização de software	Muito Satisfeito	Satisfeito	Insatisfeito
2.1. Em relação à facilidade de entender como funciona o programa			
2.2. Quanto à manipulação da superfície NURBS inicial			
2.3. Como avalia globalmente a usabilidade do software.			

<b>3. CONFIABILIDADE-</b> Evidência que o desempenho se mantém ao longo do tempo em condições estabelecidas.	Muito Satisfeito	Satisfeito	Insatisfeito
3.1. Capacidade de continuar a funcionar corretamente, após erros do próprio software ou erros de manipulação de dados.			
3.2. Como avalia globalmente a confiabilidade do software.			

<b>4. EFICIÊNCIA-</b> Evidência que os recursos e os tempos envolvidos são compatíveis com o tempo de desempenho requerido para o produto.	Muito Satisfeito	Satisfeito	Insatisfeito
4.1. Tempo de geração de uma superfície NURBS			
4.2. Como avalia em termos globais a Eficiência do software			

5. <b>Sobre o Sistema</b> – Metrifica o sistema em um contexto de disciplina e conteúdo.	Muito importante	Importante	Não é importante
5.1. A importância que atribui ao software para o aprendizado em geral.			
5.2. A importância em manipular NURBS em um ambiente virtual			
5.3. A importância geral do sistema em disciplinas correlatas			

6. <b>Avaliação de Desempenho</b> – Avaliação do usuário sobre seu desenvolvimento em usar o NURBS Edit.	Experiente	Intermediário	Iniciante
6.1. Qual seu nível de conhecimento de NURBS antes do sistema			
6.2. Qual sua experiência em edição de NURBS			
6.3. Qual seu nível de conhecimento depois de usar o software			
6.4. Qual sua experiência em editar NURBS depois da disciplina			

Comentários/ Observações:
---------------------------

*Este questionário foi elaborado com base na ISONORM 9126-1 e também considerou alguns aspectos de software educacional descritos por Campo e Campos (2001).*