

# UM DISPOSITIVO DE RASTREAMENTO BASEADO EM CABOS

**André Luis Kuckartz Cesar** (alcesar@cpts.pucrs.br)

**Leonardo Cavalheiro Langie** (langie@cpts.pucrs.br)

**Márcio Serolli Pinho** (pinho@inf.pucrs.br)

**Airton Cabral de Andrade** (kbral@pucrs.br)

Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)

FACIN - Faculdade de Informática / Av. Ipiranga, 6681 - Prédio 30 / BLOCO 4

90619-900 - Porto Alegre - RS – Brasil

## ABSTRACT

This document presents the development of a low cost movement capture platform. This device uses a wired mechanical system. Besides this device, it was also created a library that calculates, through data read from the device, the position of a point in the space.

## RESUMO

Este trabalho apresenta o desenvolvimento de uma plataforma de baixo custo para captura de movimento através de um sistema mecânico com cabos. Juntamente com o dispositivo foi criada uma biblioteca que possuiu rotinas que calculam, através de dados lidos do dispositivo, a posição de um ponto no espaço.

## 1 INTRODUÇÃO

A Realidade Virtual é uma poderosa ferramenta que pode auxiliar o homem nas mais diversas áreas. Com ela é possível simular diversas situações antes mesmo que elas aconteçam. Além disto, com a utilização desta tecnologia pode-se colocar um homem virtualmente em qualquer lugar do planeta, ou fora dele. Pode-se também manipular materiais perigosos à distância, operar uma pessoa que se encontra em outro continente ou até criar um mundo totalmente virtual que interaja com o homem em tempo real.

Um dos primeiros requisitos em ambientes virtuais é a captura (rastreamento) do movimento de um ponto no espaço.

É através do rastreamento que se pode, capturar e compreender os movimentos e comandos do usuário e, com isto, criar ambientes virtuais onde o usuário interaja. Quanto melhores forem a precisão e a velocidade no rastreamento, mais real poderá ser a simulação.

Existem inúmeros dispositivos para rastreamento, comumente chamados de “*tracking devices*”[20]. Estes dispositivos estão subdivididos em cinco grandes grupos: **acústicos** (que utilizam-se de som para determinar uma posição), **inerciais** (que são baseados em medidores de aceleração ou rotação de massas), **magnéticos** (que se baseiam em campos magnéticos gerados por emissores e detectados por um ou mais sensores), **mecânicos** (que utilizam uma estrutura rígida articulada ou com cabos para suas medições) e **óticos** (que utilizam câmeras que captam a imagem do objeto a ser rastreado e através de técnicas de triangulação determinam sua posição).

Este projeto desenvolveu uma plataforma para captura do movimento através de um sistema mecânico com cabos. O objetivo foi criar uma plataforma de rastreamento de posição para ambientes de Realidade Virtual com baixo custo. Juntamente com o

dispositivo foi criada uma biblioteca que possuiu rotinas e procedimentos que calculam, através de dados lidos do dispositivo, a posição de um ponto no espaço.

A plataforma de rastreamento (Figura 1) criada neste projeto tem a seguinte configuração: uma estrutura de alumínio de forma cúbica com aresta medindo 80 cm, dentro da qual, através de um sistema de quatro cabos pode-se determinar a posição do ponto de união dos cabos. Este ponto de união dos quatro cabos será a partir de agora chamado de **ponto-de-junção**.



**Figura 1 - Plataforma do projeto**

Com base no comprimento de cada um dos cabos pode-se determinar através de métodos algébricos (Capítulo 7) a localização do **ponto-de-junção**.

Durante o movimento dos cabos, estes devem permanecer sempre esticados a fim de permitir sua correta medição. Para tanto existe um sistema de “**recolhimento**” que foi implementado com a utilização de molas espirais.

Para que existisse uma maior facilidade de interação, foi presa ao ponto de junção, uma **caneta-cursor** que possui dois botões.

Para a determinação do comprimento de um cabo este foi enrolado em um eixo. À medida que o cabo se move, conta-se o número de voltas do eixo e determina-se o deslocamento do cabo e conseqüentemente o seu comprimento.

## **2 SISTEMAS SEMELHANTES**

Dos sistemas que trabalham através de cabos para determinar posições no espaço os mais comuns são os SPIDAR's ou “*Space Interface Device for Artificial Reality*” [29][30][31]. Existem ainda o 3-D Mouse Versão 1 e o 3-D Mouse Versão 2 [16] que foram desenvolvidos a partir dos SPIDAR's.

O **SPIDAR** (Figura 2), é um sistema aplicado para fisioterapia e reabilitação desenvolvido no **Precision and Intelligence Laboratory, Tokyo Institute of Technology** em Yokohama, Japão [30]. É um sistema que visa auxiliar na recuperação de pessoas feridas em acidentes de trânsito, esportivos ou de trabalho, auxiliando pacientes que tenham dificuldades motoras com as mãos. A idéia central do projeto é que simulando tarefas “perigosas” do cotidiano em ambientes virtuais, pode-se fazer um treinamento sem riscos para o paciente. O SPIDAR é um sistema que além de rastrear a

posição de dois pontos (dedos) ao mesmo tempo, trabalha com o sistema de *force feedback*[9] ligado a estes pontos.



Figura 2 – SPIDAR sendo utilizado[30]

O **BIG SPIDAR** [29] também desenvolvido no **Tokyo Institute of Technology** tem uma armação, bem maior que a do SPIDAR, possibilitando que um usuário entre com o corpo inteiro dentro dele. Assim como o SPIDAR, ele trabalha rastreando dois pontos (as mãos) simultaneamente e utiliza um sistema de *force feedback*. Uma característica deste sistema é que as imagens geradas pelo computador através do rastreamento, são projetadas em um grande telão na parede.

O **SPIDAR-8** [31], que também desenvolvido no **Tokyo Institute of Technology** captura os movimentos dos dedos do usuário (4 dedos em cada mão). . O SPIDAR-8 trabalha com o aplicativo estilo o cubo de RUBIK (cubo mágico) em realidade virtual (Figura 3). Neste aplicativo, o usuário brinca com o “cubo mágico” . A impressão para o usuário é de ele que realmente está segurando um cubo pois o SPIDAR-8 possui *force feedback* em cada cabo de conexão com os dedos do usuário.



Figura 3 – SPIDAR-8 sendo operado[31]

O **3-D Mouse** [16], em sua primeira versão, utiliza três sensores para o cálculo de posição. Este sistema foi montado a partir de dispositivos bidimensionais (*mouse* convencional) e tinha como objetivo criar um *mouse* tridimensional com baixo custo. Sua plataforma funciona com três cabos que partem de um mesmo plano. Este sistema possuía uma série de problemas de calibração e tinha de ser re-calibrado constantemente durante o seu uso. Os pesquisadores atribuíram estes problemas ao uso de apenas três sensores e, para corrigir este problema, foi desenvolvida uma segunda versão do **3-D**

**Mouse** com quatro sensores. Esta versão do **3-D Mouse** funciona com quatro cabos sendo que três partem de um mesmo plano e um de um plano oposto (Figura 4). Nesta versão foram resolvidos os problemas de calibração, mas permaneceram alguns problemas com o sistema de recolhimento dos cabos pois como utilizam contrapeso deve-se possuir um espaço adicional para que os pesos possam correr livremente.

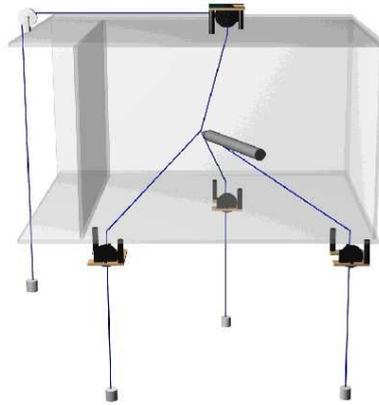


Figura 4 – Esquema do 3-D mouse com quatro sensores[16]

### 3 O RASTREADOR

O rastreador aqui apresentado captura o movimento de um ponto a partir do comprimento de um conjunto de quatro cabos presos a este ponto (**ponto-de-junção**).

A obtenção destes comprimentos passa por uma série de requisitos. São requisitos de natureza mecânica, elétrica e matemática e computacional.

Para medir o comprimento dos cabos, cada um deles é enrolado em um eixo de forma que seu movimento faça o eixo girar. O número de voltas produzido sobre o eixo permite calcular o quanto o cabo se deslocou. Neste projeto este eixo é obtido da adaptação de um eixo de um *mouse* (seção 4.1).

Para a correta medição do comprimento de um cabo é essencial que este permaneça esticado durante todos os movimentos. Para tanto se faz necessário um sistema de recolhimento dos mesmos (seção 4.2).

A desmontagem do circuito elétrico de *mouse* para uso no projeto é descrita no capítulo 5.

Para interpretar os dados provenientes do *mouse* criou-se um *driver* específico para o projeto (capítulo 6).

Depois de obtidos os comprimentos de cada cabo uma formulação matemática baseada em triangulação é usada para determinar as coordenadas do ponto (capítulo 7).

## 4 Montagem da estrutura mecânica

### 4.1 Montagem dos leitores de comprimento dos cabos

Sob o *mouse* existe uma esfera que serve de mecanismo para movimentação. Esta esfera encontra-se encostada em dois pequenos eixos que giram de acordo com a movimentação da esfera (Figura 5). Dependendo do movimento que empregarmos sobre o *mouse*, um dos eixos ou ambos, irá rotacionar.

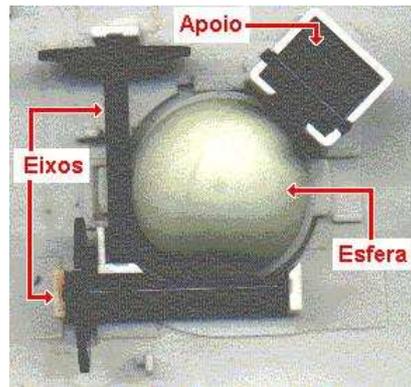


Figura 5 – Disposição dos eixos

Cada um destes eixos possui em sua extremidade um disco facetado. Esse disco fica entre um emissor e um sensor fotoelétrico (duplo) de forma a permitir ou não a passagem de sinais de luz.

Para detectar o movimento do *mouse*, o seu sistema de leitura conta o número de vezes que o receptor recebe luz do emissor dentro de um dado intervalo de tempo. O recebimento de um pulso de luz pelo receptor é denominado “pick”.

Estes sensores podem perceber a mudança de luminosidade. Desta forma existirão dois estados: com luz e sem luz.

Para que o cabo não se movimentasse ao longo do eixo foram coladas pequenas polias nos eixos, sobre as quais se enrolou os cabos. Estas polias serviram também para aumentar a circunferência do eixo e, por conseguinte, a área de atrito do cabo com o sistema de medição.

Na Figura 6 é mostrado um eixo com uma polia.

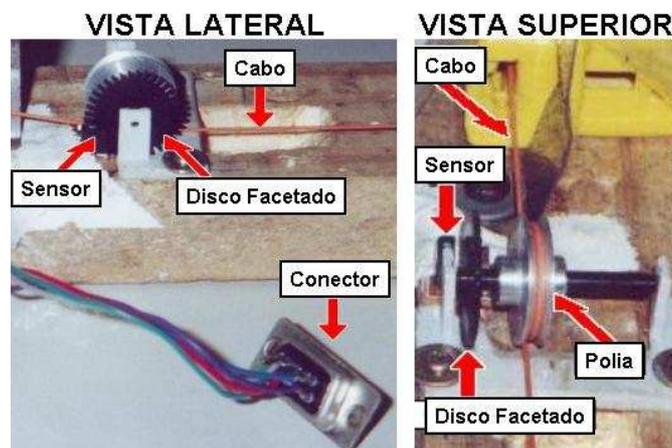


Figura 6 - Contato em todo eixo[1]

## 4.2 Sistema de recolhimento dos cabos

Para que a mão do usuário tenha liberdade de movimento os cabos devem permanecer sempre esticados. Para tanto se faz necessário um sistema de recolhimento dos mesmos.

Foram utilizadas para o sistema de recolhimento molas espirais de quatro trenas (de medição). Optou-se por este tipo de equipamento devido ao seu baixo custo e facilidade de obtenção. Para o uso no sistema, foi retirada a fita métrica de cada trena e

em seu lugar foi colocado o cabo. Os cabos usados no projeto são cabos de aço revestidos de nylon.

Para que o sistema fique balanceado as forças aplicadas pelos contrapesos em cada cabo devem ser iguais. Se uma das forças for diferente das outras o sistema ficará desbalanceado exigindo do usuário a aplicação de mais força para a realização de um movimento. Para obter esta igualdade de forças a pressão das molas espirais deve ser a mesma em todos os cabos.

## 5 Sistema elétrico dos leitores

Para que fosse possível a montagem dos leitores de movimento dos cabos, foram desmontados dois EASY MOUSE da GENIUS modelo FSUGMZE3. Optou-se por este modelo de *mouse* devido ao seu baixo custo e facilidade de obtenção.

Como os quatro pontos de leitura do movimento dos cabos ficam distantes uns dos outros (nos cantos do cubo) e os botões do *mouse* também, foi necessária a desmontagem da parte mecânica do *mouse* e a separação de alguns componentes eletrônicos (sensores, discos facetados e botões).

Os sensores e discos facetados foram utilizados (junto com as trenas e cabos de aço) para a montagem dos **Leitores**. Na Figura 7 pode-se observar o interior de um leitor.



Figura 7 – Visão interna de um Leitor[1]

As placas originais (sem os componentes acima referenciados) foram utilizadas para a montagem do **conversor-serial** (na Figura 8 pode-se observar o seu interior).

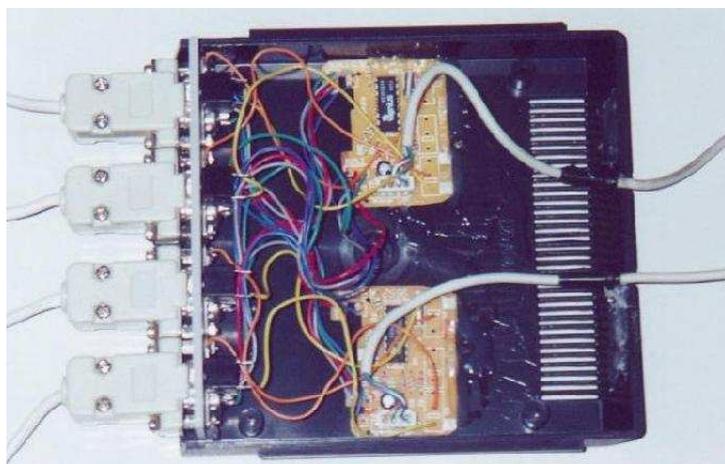


Figura 8 – Visão interna do conversor serial[1]

## 6 Bibliotecas de leitura do dispositivo

Existem basicamente dois protocolos de *mouse* serial: o padrão Microsoft e o padrão Logitech. Como a diferença mais importante entre estes dois protocolos é o número de botões, optou-se por utilizar o padrão Microsoft por ser o mais popular e por não necessitarmos de um número grande de botões. Com base neste padrão foram desenvolvidas duas bibliotecas, apresentadas a seguir.

### 6.1 Biblioteca de nível I

Para efetuar a leitura do movimento dos eixos de um *mouse* e o pressionamento de um dos botões, foi desenvolvida uma biblioteca capaz de estabelecer um “*link* de comunicação” com as portas seriais e efetuar a leitura e interpretação dos dados enviados pelo *mouse*. Essa biblioteca foi desenvolvida para funcionar com base no protocolo padrão Microsoft e sua funcionalidade equivale a um *driver* de *mouse*.

A biblioteca de *mouse* utiliza-se de funções da API (*Application Programming Interface*) do Windows para prover suas funcionalidades. Através das funções da API, é possível ter acesso a um determinado recurso do sistema (como um dispositivo de comunicação), no caso da biblioteca de *mouse*, esse recurso é a porta serial.

A biblioteca desenvolvida permite que o usuário trabalhe apenas com o número da porta serial onde o *mouse* está conectado, de forma que a manipulação de funções da API e de estruturas de dados fique transparente.

Para efetuar a **leitura de dados** da porta serial, foi criada uma rotina chamada **readSerial**. A função dessa rotina é efetuar a leitura de todos os *bytes* existentes na porta serial (em um determinado instante de tempo), e processá-los, enviando ao usuário os valores dos deslocamentos efetuados pelo *mouse* e o estado dos botões. A função de leitura retorna um valor lógico, indicando se dados foram lidos e processados.

### 6.2 Biblioteca de nível II

A biblioteca de Nível II foi desenvolvida para fornecer um meio de interação entre o programa do usuário e a plataforma de rastreamento, tendo como principais objetivos fornecer ao usuário: a posição do ponto-de-junção e o estado dos botões da Caneta-Cursor.

O rastreamento do ponto-de-junção é feito com base no comprimento dos cabos. Para isso, é necessário ter acesso aos valores gerados pelo Conversor Serial, que é o *hardware* responsável pelo gerenciamento dos Leitores e da Caneta-Cursor, os quais, respectivamente, captam os deslocamentos dos cabos e as alterações nos estados dos botões.

Para obter os valores gerados pelo Conversor Serial, a biblioteca de rastreamento realiza acesso à biblioteca de Nível I.

A biblioteca de Nível II disponibiliza três funções, que fornecem suporte para utilização da plataforma de rastreamento: **initCubeTracker**, **resetCubeTracker** e **readCubeTracker**.

A função **initCubeTracker**, tem como objetivo efetuar um processo de inicialização, essencial para a utilização da biblioteca. Ela devolve ao usuário um valor lógico indicando sucesso ou falha de sua execução.

A função **resetCubeTracker**, tem como objetivo efetuar a inicialização do comprimento dos cabos, que é um processo de definição do comprimentos dos cabos, através de um cálculo algébrico, usando como base o valor da aresta, e não os valores lidos do Conversor Serial.

A função **readCubeTracker** tem como objetivo calcular a posição do ponto-de-junção e fornecer ao usuário o estado dos botões da Caneta-Cursor.

## 7 Cálculo da posição do ponto-de-junção

A partir do momento em que se tem o comprimento de cada cabo, passa-se a etapa de cálculo da posição do ponto de junção. A seguir é apresentada a formulação algébrica usada para este fim.

Tomando-se a plataforma como um sistema cartesiano e os cabos como vetores é possível determinar o ponto dentro deste sistema cartesiano.

A distância **d** entre dois pontos  $P_1(x_1, y_1, z_1)$  e  $P_2(x_2, y_2, z_2)$  é o módulo do vetor  $\overrightarrow{P_1P_2}$  [35], isto é:

$$d(P_1, P_2) = \left| \overrightarrow{P_1P_2} \right|$$

Como se tem um cubo, todos os vértices ficam em uma mesma distância do centro (eixo cartesiano está no centro) e o que muda é o sinal dos pontos, **a** é um valor genérico que representa a metade do tamanho da aresta do cubo.

Toma-se o ponto a ser rastreado como  $P(x, y, z)$  e o comprimento dos cabos até cada vértices  $A_0, A_1, A_2,$  e  $A_3$  como  $C_0, C_1, C_2$  e  $C_3$ . Se os cabos iniciam nos respectivos vértices ( $A_0, A_1, A_2,$  e  $A_3$ ) não adjacentes do cubo, teremos um sistema como na Figura 9[30].

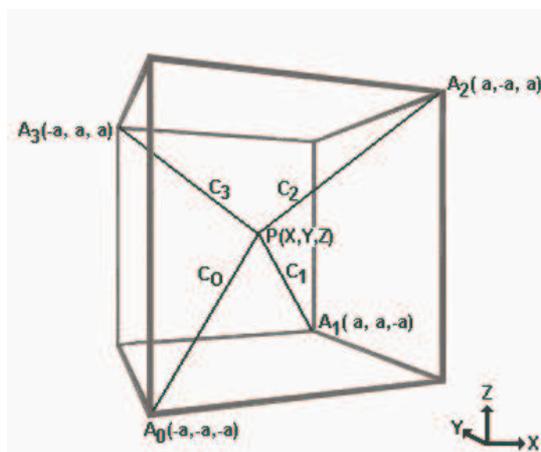


Figura 9 – Sistema de cálculo[1]

Empregando a fórmula da distância entre dois pontos para o modelo acima referido e simplificando as equações tem-se:

$$C_0^2 = (x + a)^2 + (y + a)^2 + (z + a)^2 \quad \text{(Equação 1)}$$

$$C_1^2 = (x - a)^2 + (y - a)^2 + (z + a)^2 \quad \text{(Equação 2)}$$

$$C_2^2 = (x - a)^2 + (y + a)^2 + (z - a)^2 \quad \text{(Equação 3)}$$

$$C_3^2 = (x + a)^2 + (y - a)^2 + (z - a)^2 \quad (\text{Equação 4})$$

Se for feita a diferença entre as duas respectivas equações adjacentes (equações de 1 a 4) para que se possa diminuir o número de variáveis de cada equação e, se resolver simultaneamente as equações, ter-se-á :

$$8 * a * y = C_0^2 - C_1^2 + C_2^2 - C_3^2$$

$$8 * a * z = C_0^2 + C_1^2 - C_2^2 - C_3^2$$

$$8 * a * x = C_0^2 - C_1^2 - C_2^2 + C_3^2$$

Juntando as três fórmulas anteriores ter-se-á uma equação para determinar o ponto a partir do comprimento dos cabos:

$$\left\{ \begin{array}{l} x = \frac{C_0^2 - C_1^2 - C_2^2 + C_3^2}{8 * a} \\ y = \frac{C_0^2 - C_1^2 + C_2^2 - C_3^2}{8 * a} \\ z = \frac{C_0^2 + C_1^2 - C_2^2 - C_3^2}{8 * a} \end{array} \right.$$

onde  $C_0$ ,  $C_1$ ,  $C_2$  e  $C_3$  são os comprimentos dos cabos e  $a$  é a metade do tamanho da aresta do cubo.

## 8 Processo de Calibração do Dispositivo

Um ponto importantíssimo para o funcionamento deste projeto é a calibração. Este item envolve aspectos de natureza mecânica e computacional.

A plataforma de rastreamento utiliza sensores para detectar o movimento dos discos facetados. Os sensores apresentam diversas características próprias que podem ser influenciadas por efeitos externos.

**Calibração** é um conjunto de operações que estabelece, sob condições especificadas, uma forma de obter valores que representem relações entre propriedades particulares de um sistema [40].

Ou seja, o processo de calibração consiste em obter a relação entre os comprimentos físicos dos cabos e os valores fornecidos pela biblioteca de nível I. Ao final da calibração tem-se para cada cabo uma relação do número de **picks** gerados pelos eixos do *mouse* a cada milímetro de deslocamento dos cabos.

Esta medição é baseada na contagem dos **picks** gerados (pelos sensores da plataforma) durante a movimentação controlada dos cabos. Durante este processo, os cabos serão movidos por uma distância pré-definida e com base no número de **picks** gerados durante este movimento, será definido o número de **picks** por milímetro. Com base neste número, a plataforma estará calibrada.

## 9 CONCLUSÃO

Este trabalho apresentou o desenvolvimento de uma plataforma de rastreamento de posição construída com um custo bastante baixo (menos de US\$ 200,00). Na URL <http://grv.inf.pucrs.br/Pagina/Projetos/TrackerCabos> podem ser observadas algumas animações e figuras do funcionamento do rastreador.

Os testes desenvolvidos mostraram que o uso da plataforma em aplicações de realidade virtual é bastante simples, mesmo no caso de adaptações em aplicações já construídas.

Comparado a um rastreador magnético *Polhemus*, a plataforma mostrou ser tão precisa quanto este dispositivo, não tendo o inconveniente de sofrer interferências eletromagnéticas.

Durante o desenvolvimento as maiores dificuldades encontradas estiveram ligadas à forma de passagem dos cabos pelos eixos do *mouse*. Para tanto foram necessários inúmeros testes a fim de encontrar o diâmetro e o formato ideais da polia por onde os cabos passam. Foi necessário usar uma polia em um torno, pois estas não foram encontradas disponíveis no mercado.

Pretende-se que futuramente, a partir deste projeto, possa ser implementado em trabalhos de pesquisa um sistema que insira resistência aos cabos de nossa plataforma, implementando um sistema de *force feedback*[9] para que aumente ainda mais a sensação de imersão do usuário, facilitando cada vez mais a interação homem-máquina.

## 10 BIBLIOGRAFIA

- [1] Cesar, A e Langie, L. – **Um dispositivo de rastreamento baseado em cabos**. Trabalho de conclusão do curso de informática da PUCRS. Porto Alegre, 2001.
- [2] Barbarello, J. - **PC Hardware Projects, volume 1**, 1ª edição. Prompt publications, Indianapolis, 1997.
- [3] Barbarello, J. - **PC Hardware Projects, volume 2**, 1ª edição. Prompt publications, Indianapolis, 1997.
- [4] **BEI Technologies Inc - Industrial Encoder Division**. Capturado em 30 de Setembro de 2000. Disponível na Internet <http://www.beiied.com/>
- [5] Bergsman, P. – **Controlling the World with your PC**, 1ª edição. HighText Publications Inc., Solana Beach, 1994.
- [6] **Black Box Network Services**. Capturado em 18 de Setembro de 2000. Disponível na Internet <http://www.blackbox.nl/techweb/pc/mousegi.html>
- [7] Brook, D. - **Haptic Interfaces in Virtual Reality**. University of Southampton Multimedia Systems Coursework 6<sup>th</sup>, 1997. Capturado em 21 de Agosto de 2000. Disponível na Internet <http://gales.ecs.soton.ac.uk/~dtcb98r/vrhap/>
- [8] Burdea, G. & Coiffet, P. - **Virtual Reality Technology**, 1ª edição. Wiley-Interscience, Paris, 1993.
- [9] Burdea, G., – **Force and Touch Feedback for Virtual Reality**, 1ª edição. Wiley-Interscience, New York, 1996.
- [10] Cai, Y. & Yamashitab, J. & Fukuib, Y. - **Haptic display devices in constructing virtual reality environments**. Capturado em 22 de Agosto de

2000. Disponível na Internet <http://tmc03.human.waseda.ac.jp/~wwdu97/PrHP/105/index.html>
- [11] **Digital Image Design Incorporated.** Capturado em 20 de Agosto de 2000. Disponível na Internet <http://www.didi.com/www/areas/products/cricket/>
- [12] Durlach, N. & MAVOR, A., editores - **Virtual Reality Scientific and Technological Challenges**, 1ª edição. National Academy Press, Washington, 1995.
- [13] **Gyration.** Capturado em 29 de Julho de 2000. Disponível na Internet <http://www.gyration.com/kore/catalog/products/content.html>
- [14] **Helsinki University of Technology.** Capturado em 22 de Setembro de 2000. Disponível na Internet <http://www.hut.fi/~then/mytexts/mouse.html>
- [15] **HONTKO CO.** Capturado em 1 de Outubro de 2000. Disponível na Internet <http://www.encoder.com.tw/>
- [16] Hosoda, M. & Kobayashi, Y. – **Simple 3-d Mouse with Auto-error-correction Function.** ICAT'99, 1999.
- [17] **Industrial Encoder Corporation.** Capturado em 30 de Setembro de 2000. Disponível na Internet <http://www.encoder.de.vu/>
- [18] Jacobs, J. - **Delphi Developer's guide to opengl**, 1ª edição. Wordware Publishing, Texas, 1999.
- [19] **Intelligent Mobile Manipulation Laboratory - KIST** - Capturado em 27 de Agosto de 2000. Disponível na Internet <http://centaur.kist.re.kr/field.html>
- [20] Kolozs J. - **Position Trackers for Virtual Reality**, 1999. Capturado em 22 de Agosto de 2000. Disponível na Internet <http://www.cc.utah.edu/~jk6038/>
- [21] **Krystal Technologies.** Capturado em 20 de Setembro de 2000. Disponível na Internet <http://www.krystal.com/nsfaq/Q.12.html>
- [22] Laszlo, M. - **Computational Geometry and Computer Graphics in C++**, 1ª edição. Prentice Hall, New Jersey, 1996.
- [23] Lesurf, J. – **Potentiometers** - University of Andrews, Scotland. Capturado em 3 de Setembro de 2000. Disponível na Internet <http://www.st-and.ac.uk.htm>
- [24] **Logitech.** Capturado em 10 de Agosto de 2000. Disponível na Internet <http://www.logitech.com.html>
- [25] **Mindtel.** Capturado em 29 de Agosto de 2000. Disponível na Internet <http://www.mindtel.com.html>
- [26] **MSC – Motion Control Solutions.** Capturado em 12 de Outubro de 2000. Disponível na Internet <http://www.motioncontrolsolutions.com.htm>
- [27] Pinho, M. – **Realidade Virtual e Interação em Ambientes Virtuais.** Exame de qualificação, Programa de Pós-Graduação em Ciência da Computação, Porto Alegre, 2000.
- [28] Rourke, J. – **Computational Geometry in C**, 4ª edição. Cambridge University Press, Cambridge, 1996.
- [29] Sato, M. – **A Human-Scale Direct Motion Intruction System Device for Education Systems.** IEICE Trans. Inf. & Syst., vol. E80-D, No 2, fevereiro 1997.

- [30] Seahak, I. & Somsak, W. & Koike, Y. & Sato, M. - **Personal VR system for rehabilitation to hand movement**. ICAT'98, 1998.
- [31] Sato, M. - **Tokyo Institute of Technology**. Capturado em 28 de Agosto de 2000. Disponível na Internet <http://sklab-www.pi.titech.ac.jp/>
- [32] **SensAble Technologies** – Capturado em 20 de Agosto de 2000. Disponível na Internet <http://www.sensable.com.html>
- [33] **Simandl, Petr**. Capturado em 21 de Setembro de 2000. Disponível na Internet <http://simandl.jinak.cz/stranky/elektro/mouse/mouse.htm>
- [34] Srinivasan, M. & Basdogan, C. – **Haptics in virtual environments: Taxonomy, research status, and challenges**. Computers and Graphics, Vol. 21, No. 4, 1997.
- [35] Steinbrush, A., Winterle, P. – **Geometria Analítica**, 2ª edição. MCGRAW – HILL, São Paulo, 1987.
- [36] Stuart, R. - **The design of virtual environments**, 1ª edição. MCGRAW - HILL, New York, 1996.
- [37] **Switch Channel**. . Capturado em 1 de Outubro de 2000. Disponível na Internet <http://www.switchchannel.com/product/encoder/rei.htm>
- [38] Thalmann, N. e Thalmann, D., Tradução de Tortello, J. – **Mundos Virtuais e Multimídia**, 1ª edição. LTC Editora, Rio de Janeiro, 1993.
- [39] Vince, J. - **Virtual Reality Systems**, 1ª edição. ACM Press Books, Cambridge, 1995.
- [40] **SIQ - Sistemas de Informação para Qualidade**. Capturado em 30 de Maio de 2001. Disponível na Internet. Disponível na Internet <http://www.siq.com.br/faq/faq001.htm>
- [41] UNC Tracker Research - **Wide-Area Tracking, Navigation Technology for Head-Mounted Displays**. The University of North Carolina at Chapel Hill College of Arts and Sciences. Capturado em 20 de Agosto de 2000. Disponível na Internet <http://www.cs.unc.edu/~tracker/>