

Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática
Programa de Pós-Graduação em Ciência da Computação

Avaliação do uso de Realidade
Virtual na Robótica

Mauro César Charão dos Santos

Dissertação apresentada como
requisito parcial à obtenção do
grau de mestre em Ciência da
Computação

Orientador: Prof. Dr. Márcio Serolli Pinho

Porto Alegre
Dezembro de 2006



Dados Internacionais de Catalogação na Publicação (CIP)

S237a Santos, Mauro César Charão dos
Avaliação do uso de realidade virtual na robótica /
Mauro César Charão dos Santos. - Porto Alegre, 2006.
107 f.

Diss. (Mestrado) - Fac. de Informática, PUCRS.
Orientador: Prof. Dr. Márcio Serolli Pinho.

1. Informática. 2. Robótica. 3. Realidade Virtual.
I. Título.

CDD 006.3

Ficha Catalográfica elaborada pelo
Setor de Processamento Técnico da BC-PUCRS



TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "**Avaliação do Uso de Realidade Virtual na Robótica**", apresentada por Mauro César Charão dos Santos, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Sistemas Interativos e de Visualização, aprovada em 09/01/2007 pela Comissão Examinadora:

Prof. Dr. Márcio Serolli Pinho –
Orientador (a)

PPGCC/PUCRS

Prof. Dra. Milene Selbach Silveira –

PPGCC/PUCRS

Prof. Dra. Isabel Harb Manssour –

FACIN/PUCRS

Homologada em 18/06/07, conforme Ata No. 14 pela Comissão Coordenadora.

Prof. Dr. Fernando Luís Dotti
Coordenador.

Dedico esta dissertação às três pessoas mais importantes da minha vida: meus pais e minha noiva.

Aos meus pais, João César Batista dos Santos e Neida Rosane Charão dos Santos, por nunca terem medido esforços para que eu conseguisse percorrer esta jornada, fornecendo todo tipo de apoio necessário.

À minha noiva, Letícia Siqueira Carvalho, por ter me incentivado a ingressar neste curso, por me convencer a não desistir em um momento de fraqueza, e principalmente por entender minha ausência tão freqüente, não só nestes dois anos de mestrado, mas também nos quatro anos e meio de graduação.

Agradecimentos

Em primeiro lugar, gostaria de agradecer a todos meus colegas, que sempre tiveram espírito de equipe, contribuindo muito para a conclusão deste mestrado. Em especial, gostaria de agradecer aos colegas Rafael Rieder, Régis Kopper e Rodrigo Noll, que não foram apenas colegas de aula, mas amigos que estarão sempre presentes.

Meu agradecimento especial ao Márcio Pinho, pois mais que um dedicado orientador de graduação e mestrado, foi sempre um grande amigo. Quero agradecer também a professora Milene Silveira, que teve grande contribuição nos momentos mais difíceis.

Por último, gostaria de agradecer aos integrantes do SENAI-RS, por permitirem a freqüente troca de horários de trabalho, para conciliação com as aulas, e também por valorizarem o esforço dedicado ao aperfeiçoamento.

Resumo

Este trabalho consiste no estudo de técnicas de robótica e de Realidade Virtual (RV) para desenvolver um simulador que possa ser empregado nas escolas de robótica, possuindo uma visualização adequada e uma forma de interação simples e intuitiva. Para isto, foi desenvolvido um Ambiente Virtual (AV), em linguagem C++, utilizando *OpenGL*. Foram incorporados recursos de RV para melhorar a visualização e facilitar a interação do usuário com o programa. A avaliação do ambiente foi realizada através da análise dos resultados obtidos nos experimentos realizados com usuários, em que foram gerados relatórios automáticos para as questões quantitativas, e preenchidos questionários para as questões qualitativas. Os resultados mostraram que o uso de RV auxilia na execução da tarefa, pois melhora a visualização, reduzindo o tempo total e melhorando a precisão.

Abstract

This work consists of the study of techniques of robotics and Virtual Reality (RV) to develop a simulator that can be used in the robotics schools, having an adequate visualization and a simple and intuitive way of interaction. For this a Virtual Environment (AV) for robotics was developed, in language C++, using OpenGL. RV resources have been incorporated to improve the visualization and to facilitate the interaction of the user with the program. The evaluation of the environment was performed through the analysis of the results obtained in the experiments carried out with users. Automatic reports for the quantitative questions were generated, and questionnaires were fulfilled for the qualitative questions. The results have shown that the use of RV helps in the execution of the task, improving the visualization by reducing the total time and increasing the precision.

Lista de Figuras

Figura 1	Interface principal do VRMath (adaptado de [33])	25
Figura 2	Alunos no ambiente Construct3D (adaptado de [21])	26
Figura 3	Interação no MolecularStudio com as mãos (esquerda) ou com volante (direita) (adaptado de [24])	27
Figura 4	Comparação entre uma cirurgia real (esquerda) e no EyeSi (direita) . . .	27
Figura 5	Ambiente Virtual Haptic Back (adaptado de [28])	28
Figura 6	Simulação de fogo e fumaça do Firefighter (adaptado de [20])	29
Figura 7	Braço mecânico modelo Scrobot-ER VII	31
Figura 8	Sistema de coordenadas das juntas dos robôs Scrobot (representado pelos ângulos) (adaptado de [19])	32
Figura 9	Sistema de coordenadas cartesianas (adaptado de [19])	34
Figura 10	Programa Robocell (extraído de [18])	40
Figura 11	Simulador VR Robot	41
Figura 12	Simulador HandlingPRO	42
Figura 13	arquitetura do ambiente (extraído de [12])	43
Figura 14	visualização do robô no ambiente virtual (extraído de [12])	43
Figura 15	Tela da simulação em VRML (extraído de [29])	44
Figura 16	Interface VRML com as regiões de colisão (extraído de [30])	45
Figura 17	Interface entre o servidor e o cliente (extraído de [30])	46
Figura 18	Laboratório CIM (extraído de [5])	48
Figura 19	Simulador Asimov do SENAI-RS com seus quatro ambientes	50
Figura 20	Programa VR Robotics com suas três janelas	51
Figura 21	Representações 3D dos objetos do Editor de Cenários	53
Figura 22	Teach Pendant	54
Figura 23	Rastreamento com o Isotrack II (2 sensores)	55
Figura 24	Rastreamento com uma câmera (2 sensores)	56
Figura 25	Uso do VR Robotics com monitor	56
Figura 26	Uso do VR Robotics com Óculos HMD	57
Figura 27	Uso do VR Robotics com Tela de Projeção, usuários à esquerda e tela à direita	58
Figura 28	Diagrama de classes simplificado do VR Robotics	60
Figura 29	Diagrama de atividades simplificado do VR Robotics	61
Figura 30	Ambiente de testes	64
Figura 31	Vista superior do ambiente de testes do VR Robotics	64
Figura 32	Teach Pendant 3D utilizado nos testes-piloto	66
Figura 33	Gráfico do tempo médio para realização da tarefa	70
Figura 34	Gráfico da média de tempo para pegar as peças	71
Figura 35	Gráfico da média de erro na posição final das peças	72
Figura 36	Gráfico da média da distância da garra à peça ao tentar pegar	72

Figura 37	Gráfico da média de tentativas até pegar as peças	73
Figura 38	Config - Programa de configuração do VR Robotics	83
Figura 39	Servidor de Rastreamento por Câmera reconhecendo um padrão e não reconhecendo outro	85

Lista de Tabelas

Tabela 1	As sete configurações para os testes com os usuários	67
Tabela 2	Variáveis Dependentes	68
Tabela 3	Distribuição dos testes realizados	69
Tabela 4	Lista de equipamentos para todas configurações do VR Robotics	74
Tabela 5	Estimativa de custos para implantação do VR Robotics nas escolas	75
Tabela 7	Estatísticas do VR Robotics - Monitor - Sem rastreamento - Sem Estereoscopia	99
Tabela 8	Estatísticas do VR Robotics - Monitor - Sem rastreamento - Com Estereoscopia	100
Tabela 9	Estatísticas do VR Robotics - HMD - Com estereoscopia - Com rastreamento	102
Tabela 10	Estatísticas do VR Robotics - HMD - Com estereoscopia - Sem rastreamento	103
Tabela 11	Estatísticas do VR Robotics - Tela de projeção - Sem rastreamento - Com estereoscopia	104
Tabela 12	Estatísticas do VR Robotics - Tela de projeção - Sem rastreamento - Sem estereoscopia	105
Tabela 13	Estatísticas do Asimov	107

Lista de Abreviaturas

AV	Ambiente Virtual	11
RV	Realidade Virtual	11
AVI	Ambientes Virtuais Imersivos	23
TP	Teach Pendant	24
RA	Realidade Aumentada	26
VHB	Virtual Haptic Back	27
SVE	Simple Virtual Environment	28
FDS	Fire Dynamic Simulator	28
TCP	Tool Center Point	33
VRML	Virtual Reality Modeling Language	44
HMD	Head Mounted Display	54
DSVL	DirectShow Video Processing Library	59
WTK	World Toolkit	59

Sumário

1	Introdução	23
2	RV no Ensino	25
3	Noções de Robótica	31
3.1	Componentes	31
3.2	Tecnologia de Movimento	32
3.3	Sistemas de Coordenadas	33
3.4	Sensores	33
4	Programação de Robôs	35
4.1	Programação textual	35
4.1.1	Programação orientada a juntas	35
4.1.2	Programação orientada a manipuladores	36
4.1.3	Programação orientada a tarefas	36
4.2	Programação não-textual	37
4.2.1	Walk-through	37
4.2.2	Lead-through	37
5	Sistemas computacionais para ensino de robótica	39
5.1	Sistemas comerciais	39
5.1.1	Simulador RoboCell	40
5.1.2	Simulador VR Robot	41
5.1.3	Simulador HandlingPRO	42
5.2	Sistema de realidade virtual para ensino de robótica	42
5.3	Internet (Simulado)	44
5.4	Experimentos de controle de robôs reais pela Internet	45
6	Ensino de Robótica nas Escolas	47
7	Ambiente Virtual VR Robotics	51
7.1	Interface do VR Robotics	52
7.1.1	Ambiente 3D	52
7.1.2	Editor de Cenários	53
7.1.3	<i>Teach Pendant</i> 2D	53
7.2	Rastreamento	54
7.2.1	Rastreadores Magnéticos	54
7.2.2	Rastreador Ótico	55
7.3	Visualização	55

7.3.1	Monitor	56
7.3.2	HMD	57
7.3.3	Tela de Projeção	57
7.4	Escolha da configuração ideal	58
7.5	Aspectos de implementação	59
8	Testes com Usuários	63
8.1	Objetivos	63
8.2	Plataforma de testes	63
8.3	Testes-piloto	65
8.3.1	Modificações baseadas nos testes-piloto	66
8.4	Variáveis dos Testes	66
8.4.1	Variáveis Independentes	67
8.4.2	Variáveis Dependentes	68
8.5	Protocolo de realização dos testes	69
8.6	Resultados dos testes	70
8.6.1	Tempo Gasto	70
8.6.2	Precisão no posicionamento das peças	71
8.6.3	Número de tentativas para pegar as peças	72
8.6.4	Conclusões dos testes	73
9	Conclusões	77
	Referências	79
	Apêndice A - Aplicações adicionais do VR Robotics	83
A.1	Config	83
A.2	Servidor de Rastreamento por Câmera	84
A.3	Servidor de rastreamento interno	85
	Apêndice B - Componente curricular de Robótica - SE-NAI Gravataí	87
	Apêndice C - Questionário Pré-Teste	91
	Apêndice D - Questionário Pós-Teste	93
	Apêndice E - Documento de descrição dos testes	95
	Apêndice F - Estatísticas dos testes dos usuários	99

1 Introdução

A utilização da informática na educação tem crescido nos últimos anos, sempre com o objetivo de facilitar o processo de ensino / aprendizagem. Para criar situações que representem a realidade, foram desenvolvidos diversos tipos de ferramentas. Uma das técnicas mais utilizadas são os **simuladores**. Primeiramente adotou-se o uso de simuladores textuais, que geralmente recebiam um arquivo de entrada ou dados digitados e geravam um relatório com os resultados. Os primeiros simuladores gráficos criados trabalhavam com gráficos bidimensionais, possuindo uma representação simbólica da realidade. Após isto, passaram a ser criados Ambientes Virtuais (AV), que exibem um AV tridimensional, com o qual é possível realizar interações, simulando a situação real. Por não utilizarem recursos de RV, estes ambientes são chamados de ambientes virtuais não-imersivos. Este tipo de sistema é muito utilizado em diversas áreas.

Apesar desta tendência, o uso de ambientes virtuais sem imersão em alguns casos pode não ser a melhor forma de criar um ambiente eficaz para atingir o objetivo esperado, de criar aplicações com fácil interação e que representem o mundo real de forma adequada. Um exemplo disto são as aplicações que necessitam manipular objetos tridimensionais, como aplicações para o ensino de geometria espacial, no qual é possível criar e manipular os objetos com os formatos geométricos estudados, ou robótica em que é necessário manipular um braço de robô para efetuar uma certa operação.

Uma alternativa para solucionar este problema são os Ambientes Virtuais Imersivos (AVI), com uso de equipamentos de RV como óculos estereoscópicos, luva e rastreador. Desta forma, pode haver um ganho significativo em relação às aplicações tradicionais, pois a interação torna-se mais próxima da realidade, sendo também mais intuitiva.

Com o uso de RV os sentidos da percepção humana podem ser melhor aproveitados, pois a visualização estereoscópica em conjunto com o rastreamento dos movimentos do usuário permitem gerar melhor visualização, uma interação mais robusta e a locomoção ao redor do cenário. Além disto, recursos como tato pode permitir a geração de sensações de toque, melhorando a percepção do ambiente.

Segundo Bell [10], a experiência é uma poderosa ferramenta de aprendizagem, pois pode proporcionar um entendimento detalhado dos conteúdos e processos. Os ambientes de RV possuem a característica de simular o mundo real, permitindo que os alunos experimentem as tarefas várias vezes, tendo mais chances de aprendizagem. Além disso, com RV é possível realizar um treinamento com segurança, o que nem sempre é possível no mundo real.

Os ambientes de RV podem ainda ajudar a diminuir os custos, nos casos em que os equipamentos reais ou os materiais de reposição são muito caros, gerando aplicações mais próximas

da realidade, pois permitem interação mais semelhante à realizada no mundo real.

Neste contexto, este trabalho consiste no estudo de técnicas de robótica e de RV para desenvolver um simulador que possa ser empregado nas escolas de robótica, possuindo uma visualização adequada e uma forma de interação simples e intuitiva.

O ambiente desenvolvido, chamado de VR Robotics, simula o braço mecânico Scorbot-ER VII, fabricado pela empresa israelense *Eshed Robotec* [7]. Este simulador possui um AV tridimensional, que permite realizar a navegação no cenário, um editor de cenários, que permite a manipulação de todos objetos da cena, e o *Teach Pendant* (TP), uma espécie de controle remoto com fio, que permite fazer os movimentos necessários com o robô.

Para melhorar a visualização este ambiente incorporou recursos de estereoscopia, para tornar a interação mais intuitiva foi utilizado o rastreamento da cabeça do usuário, e para descobrir qual dos dispositivos de visualização gera melhor resultado, foram utilizados três diferentes tipos.

Para validar o programa foram realizados testes com usuários, utilizando os três tipos de dispositivos de visualização, com ou sem rastreamento e estereoscopia.

O restante da dissertação é estruturado da seguinte forma: o Capítulo 2 aborda o uso de RV no ensino em geral. O Capítulo 3 discute aspectos conceituais importantes sobre robótica. O Capítulo 4 apresenta várias formas existentes para programação de robôs. O Capítulo 5 apresenta vários tipos de sistemas computacionais para ensino de robótica. O Capítulo 6 mostra como é ensinado robótica em três escolas pesquisadas. O Capítulo 7 detalha o ambiente desenvolvido. Finalmente, o Capítulo 8 detalha os testes realizados e os resultados obtidos. No Anexo A são detalhadas as aplicações adicionais desenvolvidas, no Anexo B é exibido o componente curricular de robótica de uma das escolas pesquisadas, no Anexo C está o questionário pré-teste, no Anexo D o questionário pós-teste, no Anexo E o documento de descrição dos testes e no Anexo F estão as estatísticas dos testes dos usuários.

2 RV no Ensino

Ambientes de RV com objetivo educacional foram desenvolvidos para diversas áreas, como ferramentas de apoio ao ensino. Segundo Sánchez [31], o uso de ambientes de RV voltados para o ensino é mais apropriado para conhecimentos abstratos, que possuem conceitos mais complexos, como teorias, regras e processos. Isto se deve ao fato de que eles podem não ter uma correspondência clara no mundo real, por serem abstratos, teóricos ou entidades genéricas. Além disso, estes conhecimentos são difíceis de visualizar ou imaginar, sendo então difíceis de representar em papel ou verbalmente. Por isso os sistemas educacionais com esta finalidade têm crescido muito. São desenvolvidas aplicações para ensinar matemática [21], principalmente para trabalhar com formas tridimensionais, para ensinar biologia [24], possibilitando a manipulação dos compostos orgânicos, além de sistemas para trabalhar com radiologia [13] sem correr o risco da exposição à radiação, etc.

Uma das áreas em que o uso de RV se mostrou eficiente é na Geometria Espacial, pois muitos alunos têm dificuldade de imaginar as formas tridimensionais, e com um AV isto pode ser facilmente reproduzido. O sistema VRMath [33], visto na Figura 1, foi desenvolvido para ajudar crianças no entendimento da geometria espacial, permitindo que o usuário crie objetos, manipule e navegue pelo cenário criado. Este programa permite que crianças criem formas tridimensionais, através de uma interface de programação de eventos, com comandos intuitivos, a partir de uma área de visualização e interação e de uma área com um fórum hipermídia, que permite o diálogo com outras crianças para tirar dúvidas.

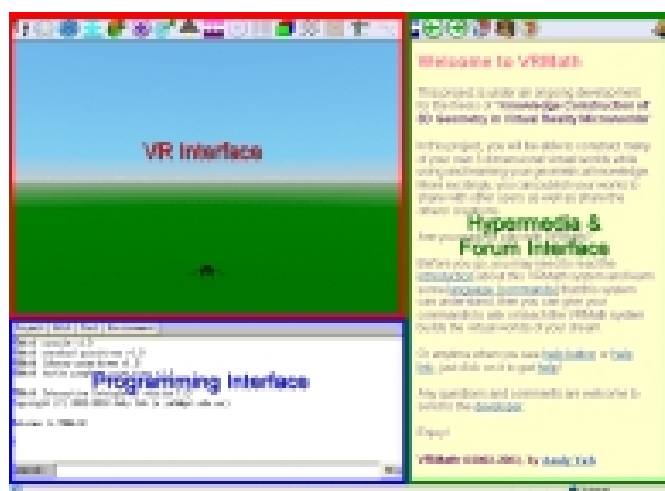


Figura 1: Interface principal do VRMath (adaptado de [33])

Outra ferramenta de RV com o objetivo de ensinar geometria espacial, que também ensina

outros conceitos de matemática, foi desenvolvida por Kaufmann [21]. Esta ferramenta se chama Construct3D, e consiste de uma ferramenta colaborativa com o uso de Realidade Aumentada (RA) [9], em que uma tarefa é executada por dois alunos, ou por um aluno auxiliado pelo professor, que também interage no AV, como visto na Figura 2.



Figura 2: Alunos no ambiente Construct3D (adaptado de [21])

Por ser um sistema de RA, o óculos utilizado é semi-transparente, permitindo que o usuário veja o mundo real com imagens virtuais sobrepostas a ele, formando o cenário desejado. Esta técnica é interessante porque muitas pessoas perdem a orientação ao utilizar um ambiente totalmente virtual, assim, a visão do mundo real possibilita que o usuário não perca a referência. Para criar as formas tridimensionais, em uma das mãos o usuário utiliza uma caneta equipada com um rastreador e dois botões, e na outra mão uma prancheta, onde é projetado o menu da aplicação. No menu existem os botões com as formas geométricas possíveis e também as operações para carregar e salvar cenários, apagar objetos, etc.

Um sistema educacional de biologia chamado MolecularStudio, visto na Figura 3, foi desenvolvido por Lu [24], visando ensinar algumas estruturas bio-moleculares. Este sistema foi criado para permitir que os alunos aprendam a estrutura das moléculas navegando entre os átomos ao invés de utilizar os tradicionais modelos bio-moleculares aumentados feitos de madeira, metal ou plástico, que além de difíceis de construir, são estruturas estáticas. No sistema MolecularStudio é possível interagir com as moléculas e navegar por dentro das mesmas. Para isto pode ser utilizada uma luva ou volante utilizados para jogos, além do óculos do tipo *shutter*. [27]

Um exemplo de uso do sistema é no estudo da estrutura do vírus HIV, que no mundo real é limitado apenas a pessoas muito especializadas. Além de reduzir o custo, pois um laboratório deste tipo requer grandes investimentos e reposição periódica de materiais, um experimento bio-molecular real pode ser muito perigoso. Neste sistema um grande número de estruturas bio-moleculares estão disponíveis para serem observadas, tocadas e manipuladas sem risco algum e com facilidade, sendo que os dispositivos de manipulação utilizados, segundo o autor, são intuitivos.

Na área da medicina, um treinamento cirúrgico algumas vezes não pode ser executado com pessoas vivas, devido ao risco para os pacientes. Com cadáveres não se tem o resultado es-



Figura 3: Interação no MolecularStudio com as mãos (esquerda) ou com volante (direita) (adaptado de [24])

perado, além de ser caro e não permitir o corte do mesmo órgão várias vezes. Com animais também está se tornando inviável por razões econômicas, além destes animais não possuem exatamente a mesma estrutura anatômica humana ou deficiência necessária para realizar o treinamento. Além disto, existe a questão ética de não oferecer riscos à pessoas ou animais vivos ao se realizar este tipo de treinamento. Para estes casos o uso de aplicações com RV é muito útil, pois possibilita que o médico faça um treinamento para operações tantas vezes quanto necessário, até atingir experiência e confiança suficientes para realizar a cirurgia em pacientes reais. Neste casos, deve ser tomado muito cuidado para que o AV esteja muito semelhante ao real, para que seja possível atingir o resultado esperado.

Um ambiente com essas características, chamado EyeSi, foi desenvolvido por Wagner, e pode ser visto na Figura 4 [32]. O programa foi criado para simular cirurgias nos olhos, que, por serem micro-cirurgias, requerem precisão milimétrica por parte do médico.



Figura 4: Comparação entre uma cirurgia real (esquerda) e no EyeSi (direita)

Para que o ambiente ficasse mais próximo do real, este foi adaptado a um aparelho semelhante ao microscópio utilizado neste tipo de cirurgias, onde foram colocados pequenos displays de cristal líquido, que exibem as imagens estereoscópicas. No local onde estaria o paciente foi colocada uma face artificial. As ferramentas disponibilizadas no AV são muito semelhantes às utilizadas na cirurgia real.

Outra área da medicina em que a RV tem sido usada é no treinamento para diagnósticos

pelo tato, através da palpação. Um sistema com esse propósito chamado Virtual Haptic Back (VHB) [28], foi desenvolvido para treinamento em medicina osteopática, terapia física, terapia por massagem e áreas relacionados. O sistema utiliza dois PHANToMs 3.0 para permitir o treinamento de palpação com as duas mãos, como visto na Figura 5. Existem dois níveis de retorno tátil, sendo o primeiro para simular a pele e o segundo para simular os ossos. Para informar se a vértebra escolhida é a correta, o usuário pressiona um botão com o pé, então um retorno sonoro é emitido informando se está correto ou errado.



Figura 5: Ambiente Virtual Haptic Back (adaptado de [28])

As deficiências ósseas podem ser geradas automaticamente ou definidas pelo professor antes da realização do teste. Existem dois tipos de deficiência, sendo o primeiro uma rotação anormal de uma vértebra e o segundo uma dificuldade de dobrar a coluna em alguma vértebra. O sistema também permite gravar as ações de um profissional para reproduzi-las aos alunos posteriormente.

Outro ambiente de RV para treinamento, chamado de Firefighter, foi desenvolvido por Julien [20]. O objetivo é realizar o treinamento de oficiais do corpo de bombeiros no comando de 4 a 8 bombeiros para combater um incêndio. Neste ambiente é possível ver um cenário com uma casa mobiliada em chamas, outros bombeiros, um caminhão de bombeiros, um hidrante e várias ferramentas utilizadas pelos bombeiros no combate a um incêndio. Para desenvolver este simulador foram empregadas várias ferramentas, a fim de tornar o resultado visual mais realístico e diminuir o tempo de desenvolvimento. Para criar o AV foi utilizada a biblioteca Simple Virtual Environment (SVE) [22], para simular o fogo e a fumaça foi utilizada uma ferramenta chamada NIST's Fire Dynamic Simulator (FDS) [11], gerando o efeito visto na Figura 6, que implementa uma simulação de fogo e fumaça considerando fatores físicos como vento, água jogada pelos bombeiros, etc.

A entrada dos comandos é feita por meio de uma interface gráfica, que exhibe os comandos possíveis e permite a escolha. Entre os comandos disponíveis estão: cortar, despedaçar, caminhar, gatinhar, escalar, puxar, molhar com a mangueira e carregar diversas ferramentas.

A execução dos comandos em uma ordem correta faz com que o fogo seja apagado. Segundo Julien [20], os treinamentos reais feitos pelos bombeiros ocorrem sempre no mesmo local, custam caro, são perigosos e poluem o ambiente. Com o uso de RV todos esses problemas podem ser resolvidos, pois podem ser simuladas diversas situações em cenários diferentes, não

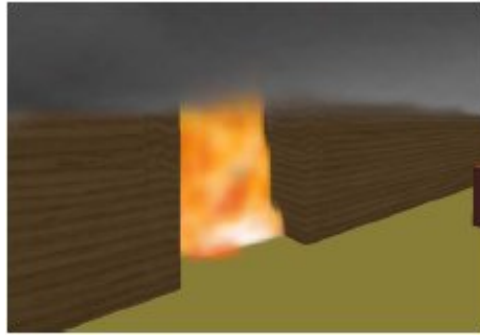


Figura 6: Simulação de fogo e fumaça do Firefighter (adaptado de [20])

gerando poluição e sem ter custo de reposição. Além disso, para ser promovido a comandante, geralmente é aplicada apenas uma prova teórica restrita aos bombeiros experientes. Mas em algumas unidades a demanda para apagar fogos é pequena e o bombeiro não vivenciou muitas situações de risco, que podem ser simuladas no AV.

Nos ambientes apresentados neste capítulo, é possível perceber que alguns utilizam os recursos de RV para criar formas de interação que se assemelhem com a realidade, o que torna estes sistemas mais intuitivos e realísticos. Porém, alguns sistemas que segundo os autores são de RV, utilizam linhas de comando como forma de interação, o que pode dificultar o uso do sistema, por não possuir uma forma direta de interação.

3 Noções de Robótica

Um robô é um manipulador multifuncional programável, composto por mecanismos que incluem vários graus de liberdade, geralmente tendo aparência de um ou vários braços, finalizado por um punho, capaz de segurar uma ferramenta, uma peça ou algum outro dispositivo [8]. Na Figura 7 é mostrado um robô modelo Scorbot-ER VII. Este robô é capaz de mover materiais, partes, peças, ferramentas ou dispositivos específicos, através de movimentos programados, armazenados na memória, para a realização de uma grande variedade de tarefas.



Figura 7: Braço mecânico modelo Scorbot-ER VII

Existem alguns conceitos importantes para o entendimento do funcionamento de um braço mecânico, como os segmentos que o formam, as tecnologias utilizadas para movimentá-lo, os sistemas de coordenadas existentes para posicionamento e movimentação, e os tipos de sensores utilizados. Estes conceitos serão apresentados nas seções a seguir.

3.1 Componentes

Os braços mecânicos são compostos por uma quantidade variável de segmentos, dependendo da aplicação, como visto na Figura 8, sendo os componentes básicos explicados a seguir:

- tronco (*torso, body sweep*): é a primeira junta, presa ao corpo do robô, permitindo o giro do robô em torno do eixo vertical;
- ombro (*shoulder*): é a segunda junta, presa ao tronco de forma que, quando o tronco gire, todo sistema sofra uma rotação. Nesta junta é preso o braço superior, que pode ser rotacionado para cima ou para baixo;
- cotovelo (*elbow*): é a terceira junta, presa entre os braços superior e inferior, que pode ser rotacionado para cima ou para baixo;
- punho (*wrist*): o punho é a última junta, presa entre o braço inferior e a garra, que pode realizar três tipos de movimentos, sendo um para rotacionar a garra em torno do eixo horizontal, mudando a elevação (*pitch*), outro de rotação da garra em torno de seu centro (*roll*) e o último de rotação da garra em torno do eixo vertical (*yaw*);
- garra (*gripper*): existem vários tipos de garra, dependendo da aplicação, como por exemplo, para pegar objetos, pintura, solda, etc. Dependendo do tipo existem comandos para abrir, fechar, ativar, desativar, etc.

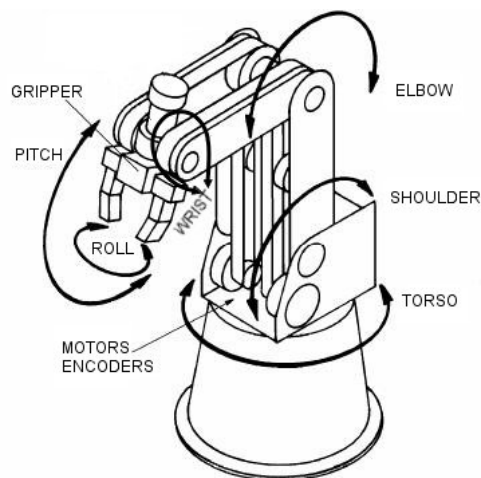


Figura 8: Sistema de coordenadas das juntas dos robôs Scorbot (representado pelos ângulos) (adaptado de [19])

3.2 Tecnologia de Movimento

Outro aspecto importante quando se analisam robôs é a tecnologia usada para criar os movimentos. Os robôs industriais podem ser movidos de três formas, dependendo da aplicação [8]. A primeira forma é através de **atuadores pneumáticos**, utilizados quando o robô não necessita transportar uma grande carga. A segunda forma é através de **atuadores hidráulicos**, que

umentam o poder de carga, mas possuem alguns problemas, pois é necessário filtrar o fluido, além do mecanismo ser suscetível a vazamentos. A terceira forma é através de **motores elétricos**, sendo de mais fácil controle, porém necessita de redutores de velocidade, que podem causar erros de operação.

3.3 Sistemas de Coordenadas

Para definir a posição atual do ponto central da ferramenta, conhecido como *Tool Center Point* (TCP), existem dois tipos de sistemas de coordenadas: o sistema de coordenadas das juntas e o sistema de coordenadas cartesianas (ou XYZ). Para os dois sistemas, as posições podem ser absolutas ou relativas [19].

Para representar uma coordenada no sistema de coordenadas das juntas, é necessário armazenar o ângulo de cada junta, além dos ângulos da garra. Na Figura 8 é possível identificar os ângulos necessários para definir uma posição nos robôs Scorbot, que são *Base (Torso)*, *Shoulder*, *Elbow*, *Pitch* e *Roll*. Uma posição neste sistema pode ser representada de duas formas, sendo a absoluta, que utiliza os ângulos reais de cada junta, e a relativa, que utiliza as diferenças entre os ângulos atuais e os da posição desejada, indicando o quanto cada junta deve andar para atingir a posição relativa. Para obter a posição atual da garra do robô neste caso é utilizada a cinemática direta, realizando um cálculo utilizando os ângulos de cada parte do robô.

Para representar uma coordenada no sistema de coordenadas cartesianas, é necessário indicar ainda o *pitch* e o *roll* da ferramenta, além do ponto central da ferramenta no espaço tridimensional, como visto na Figura 9. Este ponto central é representado por um ponto, que indica as coordenadas (X, Y, Z) da ferramenta em relação ao ponto de origem, localizado no centro da base do robô, quando as coordenadas forem absolutas, ou em relação ao ponto atual, no caso de coordenadas relativas. Para obter o ângulo de cada uma das juntas, dado o TCP, é utilizada a técnica de cinemática inversa [8].

3.4 Sensores

Segundo James [25], para realizar as trajetórias o robô deve evitar colisão com obstáculos, portanto os robôs podem incluir sensores, sendo alguns deles listados abaixo:

- sensor de visão: com uma câmera acoplada podem ser reconhecidos objetos indesejáveis, fazendo com que o robô mude a trajetória para atingir o ponto desejado;
- sensor de toque: permite que o robô reconheça que bateu em um objeto, evitando danos;
- sensor de proximidade: permite identificar a presença de um objeto. A vantagem deste

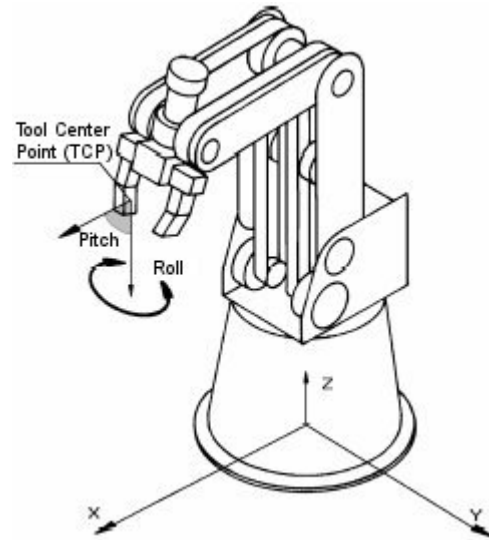


Figura 9: Sistema de coordenadas cartesianas (adaptado de [19])

sensor sobre o de toque está em não necessitar de contato físico;

- sensor de voz: este sensor não tem a função de evitar colisões, mas permite a programação do robô através do reconhecimento de comandos de voz.

4 Programação de Robôs

Existem várias formas de programar os robôs, sendo que algumas requerem que o usuário conheça uma linguagem de programação, e através dela, emita "comandos" para o robô. Estas formas são chamadas de programação textual. Já uma segunda categoria, chamada de programação não-textual, permite que o robô seja "ensinado" através da movimentação física do braço mecânico, quando a trajetória desejada é armazenada de maneira automática ou manual, dependendo do tipo de programação utilizada.

Estas duas formas de programação não são mutuamente-exclusivas, podendo ser combinadas, utilizando uma forma não-textual para posicionar o robô em locais específicos, e a forma textual para realizar os movimentos, passando por estas posições.

4.1 Programação textual

A programação textual consiste em escrever um programa contendo os comandos necessários para o robô realizar a tarefa desejada. Alguns modelos trazem acoplados um monitor e um teclado, permitindo a programação diretamente no robô, outros permitem a conexão a um computador pela porta serial ou USB, outros permitem a entrada do programa por disquete. Um dos problemas enfrentados neste caso, é que existem diversas linguagens de programação, pois não existe um padrão entre os fabricantes.

Os comandos dos robôs incluem deslocamento para uma posição absoluta, ou para uma posição relativa à posição atual. Para isto é necessário gerar uma trajetória, que é calculada pelo micro-controlador do robô.

A programação textual pode ser dividida em três classes: orientada a juntas, orientada a manipuladores e orientada a tarefas. As características de cada classe são apresentadas a seguir.

4.1.1 Programação orientada a juntas

Na programação orientada a juntas, o programa é baseado no nível das juntas, sendo que em cada movimento são informados detalhes sobre cada um dos atuadores do robô. Tipicamente uma instrução indica uma seqüência de movimentos das juntas do robô, da seguinte forma: JMOVE q1, q2, q3, q4, q5, q6. Neste caso as variáveis de q1 a q6 representam cada junta do

robô. O valor informado é um ângulo ou um deslocamento, dependendo do tipo de junta. Um exemplo de linguagem deste tipo é ARMBASIC, para o *Microbot Minimover 5* [8].

4.1.2 Programação orientada a manipuladores

Na programação orientada a manipuladores os comandos guiam o ponto central da ferramenta (TCP - Tool Center Point) de um local a outro. Em muitos sistemas o programa pode ser construído nas coordenadas cartesianas, cilíndricas ou da ferramenta. Ainda podem ser incorporados comandos para informar tempos de espera, abrir ou fechar a garra, etc. Neste tipo de programação, a conversão da posição desejada para os valores dos ângulos em cada junta é feita automaticamente pelo *software* do robô. Uma instrução de deslocamento neste caso é representada na seguinte forma: MOVE px, py, pz, Θ , ϕ , ψ . Onde px, py, pz representam a posição da ferramenta e Θ , ϕ , ψ representam os ângulos de orientação global do robô. Exemplos deste tipo de linguagem de programação são VAL para o Unimation PUMA, SIGLA para o Olivetti SIGMA e TEACH para o Bendix PACS [8].

4.1.3 Programação orientada a tarefas

A programação orientada a tarefas é considerada a segunda geração de linguagens de programação de robôs. Este tipo de linguagem permite executar tarefas mais complexas, provê construções para controlar múltiplos braços mecânicos em cooperação, aceita instruções mais abstratas e menos detalhadas, além de ter uma melhora na eficiência e ser mais fácil de programar, por ser mais similar à linguagem falada.

Os programas construídos precisam modelar todo o ambiente de trabalho dos robôs, além de registrarem os movimentos dos objetos reais. Normalmente, para que os robôs possam aceitar os comandos, é necessário ter uma descrição precisa do cenário, com os seguintes itens:

- descrição geométrica de todos objetos;
- descrição das características físicas de todos objetos, como massa e inércia;
- descrição geométrica dos robôs;
- descrição das características dos robôs, como limite das juntas, aceleração, e características dos sensores;
- posições de todos objetos.

Estas linguagens aceitam instruções como: COLOQUE A PEÇA NO ORIFÍCIO.

Exemplos de linguagens dessa categoria são: AL, desenvolvida no Laboratório de IA de Stanford [14], AUTOPASS da IBM [23], LAMA da MIT [26] e LAMA-S da IRIA, na França.

4.2 Programação não-textual

A programação textual permite a criação de programas complexos, controlando até mesmo mais de um robô ao mesmo tempo, porém, o programador precisa conhecer a fundo a linguagem de programação, além de ser demorado. Em face destes problemas, foram desenvolvidas algumas formas de programar-se um robô sem precisar conhecer a linguagem de programação do mesmo. Algumas dessas formas são explicadas nas seções a seguir.

4.2.1 Walk-through

Walk-through é o tipo de programação em que o operador faz o robô percorrer a trajetória desejada com as mãos. Nesse caso, chaves e botões no painel de controle são usadas para especificar quando entrar ou sair do modo de programação, indicar a taxa de amostragem para armazenar os pontos, executar o movimento gravado, ou excluir um movimento.

Para gerar um programa, o operador guia o braço do robô com as mãos, percorrendo a trajetória desejada, na velocidade desejada. Para que seja possível realizar essa trajetória, o robô possui mecanismos que permitem que o braço fique leve quando entra no modo de programação. Enquanto isto ocorre, os sinais analógicos gerados pelo movimento são convertidos para a linguagem do robô e gravados, de modo que o robô possa percorrer a trajetória novamente.

A vantagem deste tipo de programação está na facilidade de utilização, pois nenhum conhecimento de linguagem de programação é necessário. Já a desvantagem é que caso seja cometido um erro, todo movimento deve ser feito novamente.

4.2.2 Lead-through

Lead-through é um dos tipos mais comuns de programação de robôs industriais. A programação é feita através de um dispositivo de comando manual, como um controle remoto com fio, que permite fazer os movimentos necessários com o robô. Este dispositivo é comumente chamado de *Teach Pendant* (TP), e possui funções para mover ou rotacionar cada junta, abrir ou fechar a garra, registrar a posição atual do TCP, etc.

Neste caso, para realizar a programação, o operador move o robô com o TP, realizando a trajetória desejada e salvando posições-chave do movimento desejado. Enquanto o robô é ensinado, as posições, velocidade e tempo de espera são gravados, sendo que durante a execução do programa, o controlador se encarrega de realizar os movimentos necessários para levar o robô até a posição desejada. A trajetória criada pode ser gravada como um programa, que o robô é capaz de executar, repetindo o movimento armazenado.

As vantagens deste tipo de programação são:

- a possibilidade de se programar muitas funções especiais, como tempo de espera e velocidade de movimento;
- a capacidade de edição é melhor que no *walk-through*;
- a criação de programas maiores que no *walk-through*, pois com o TP normalmente são armazenados apenas os pontos extremos de cada reta da trajetória.

Algumas desvantagens deste tipo de programação são:

- permite a programação do robô somente no modo on-line;
- permite apenas a criação de programas simples, com ramificações limitadas;
- exige grande habilidade do operador para movimentos mais complexos.

Existem ainda outras formas de programação não-textual, que não serão abordadas aqui por serem pouco utilizadas, mas podem ser obtidas no livro “Fundamentals of Robotics” [8].

5 Sistemas computacionais para ensino de robótica

O uso de robôs reais para ensino de robótica é inviável para muitas escolas, devido ao alto custo do equipamento. Segundo Jason [15], existem quatro opções viáveis para ensinar robótica nas escolas, que são:

- utilizar robôs educacionais de baixo custo;
- utilizar simuladores educacionais de baixo custo;
- utilizar simuladores com RV;
- utilizar simuladores juntamente com um ou mais robôs reais, para executar os programas criados nos simuladores.

Os robôs educacionais possuem baixa confiabilidade, os simuladores de baixo custo, em geral, possuem várias limitações, além de dificultar o uso, pois geralmente não possuem uma visualização adequada. Por isto, os simuladores com RV estão se tornando uma alternativa interessante ao ensino.

Existem diversos tipos de sistemas de simulação de robótica, divididos em sistemas comerciais e sistemas não comerciais, sendo que alguns utilizam recursos de RV, outros podem ser acessados pela Internet permitindo inclusive, em alguns casos, acesso a um laboratório real remotamente, através da Internet, como visto nas seções a seguir.

5.1 Sistemas comerciais

Os simuladores comerciais aqui apresentados possuem uma área de visualização 3D, permitem a conexão com o robô real e a programação tanto na forma textual quanto na forma não-textual, podendo-se misturar as duas formas. Segundo os fabricantes, estes programas são simuladores RV, porém não incluem suporte a dispositivos de RV. Então, nas categorias de Jason, estes simuladores se enquadram na categoria de simulador com RV, por não serem simuladores de baixo custo e possuírem visualização 3D.

5.1.1 Simulador RoboCell

Para os modelos de robôs Scorbot fabricados pela empresa israelense *Eshed Robotec* [7], existe o simulador RoboCell, criado pela empresa americana Intelitek [4], visto na Figura 10. Este simulador integra um programa antigo da mesma empresa, chamado de Scorbase [19], com um módulo gráfico 3D, a fim de permitir a criação de programas, simulação da execução e a obtenção dos pontos da trajetória desejada sem a necessidade de um robô real. O simulador permite o envio dos programas criados para o robô real, bem como a simulação sincronizada com o robô real.

Neste simulador, uma célula consiste na descrição do cenário em que o robô está inserido, colocando uma representação dos objetos próximos ao robô, como mesa, peças, caixas, etc. Dessa forma, uma mesma célula pode ser utilizada por vários programas criados.

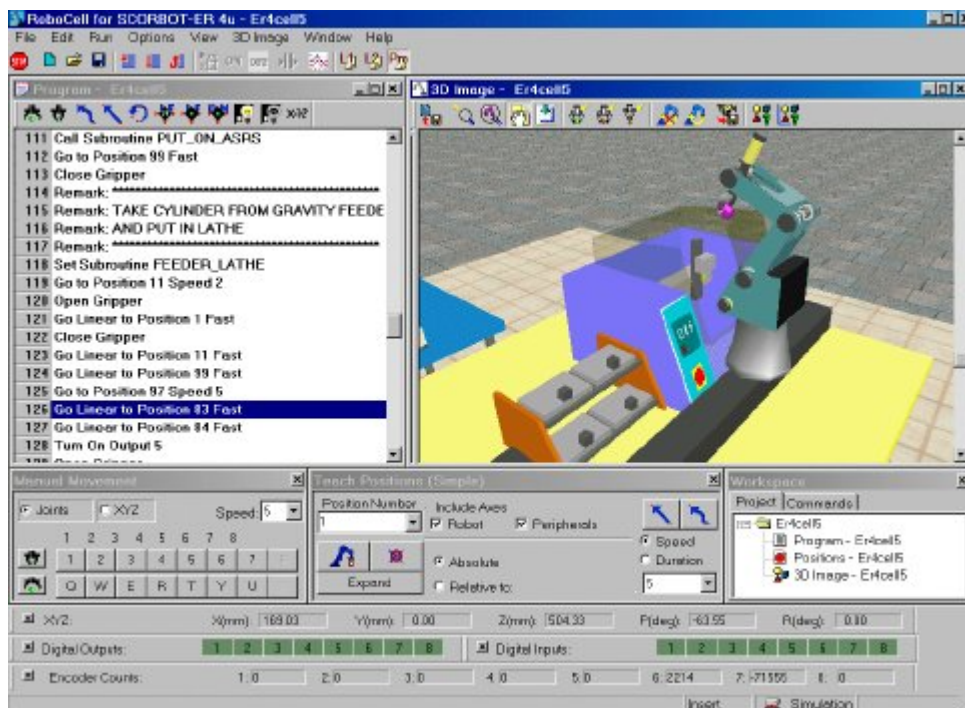


Figura 10: Programa Robocell (extraído de [18])

Para programar o robô podem ser combinadas a forma textual com a forma não-textual. A forma não-textual de programação é feita diretamente na interface gráfica da célula virtual, permitindo a realização de rotações na câmera, a aplicação de *zoom* para aumentar a precisão, a visualização das posições dos objetos, o ajuste das posições gravadas, além de outras funções. Estas posições gravadas são conhecidas como pontos de referência, e são os pontos-chave utilizados para movimentar o robô durante a execução de um programa, sendo usados na maioria das metodologias de programação. Existem duas formas de marcar este pontos:

- movimento manual: em uma tela com funções similares às disponíveis no TP real, é possível realizar manualmente o movimento de uma das juntas do robô;

- movimento automático: o robô movimenta-se automaticamente até uma posição indicada, sendo necessário para isto informar os valores de x, y, z, pitch e roll.

A forma textual de programação é feita através de uma linguagem de programação própria do simulador que possui operadores de comparação, permite a criação de sub-rotinas para melhor organização do código e possui comandos intuitivos e simples, utilizando as posições armazenadas, como por exemplo “vá para a posição 1 com velocidade 5”, “abra/feche a garra” ou “pule para o *label* início”. Nesse caso, os *labels* servem para permitir a criação de laços de repetição, possibilitando a execução repetida de um conjunto de operações. Além disto, existem comandos como o salto condicional, salvamento da posição atual, criação de variáveis, impressão de valores na tela, etc. Para evitar que o usuário tenha que decorar os comandos, eles são selecionados em uma lista com todos os comandos disponíveis, preenchendo-se em seguida os parâmetros adicionais do comando escolhido, como por exemplo o ponto ou *label*.

Com o uso deste programa, o valor gasto para montar um laboratório diminui, pois cada robô Scorbot custa aproximadamente U\$6.500, enquanto cada licença do programa custa aproximadamente U\$1.000. Desta forma, segundo Jason [15], o ideal é ter o programa instalado em várias máquinas e um ou dois robôs reais, para executar os programas depois de prontos, para que todos alunos possam aprender a programar.

Apesar de diminuir o custo, este programa ainda possui um valor elevado, dificultando a criação de laboratórios para grandes turmas.

5.1.2 Simulador VR Robot

Uma alternativa de *software* mais barato é o VR Robot, da empresa Denford [2], visto na Figura 11. Ele custa entre U\$200 e U\$600 por estação, dependendo das opções. Este programa é compatível com o robô Mitsubishi RV-M1. O problema neste caso, é o custo do robô, que é de aproximadamente U\$16.000.

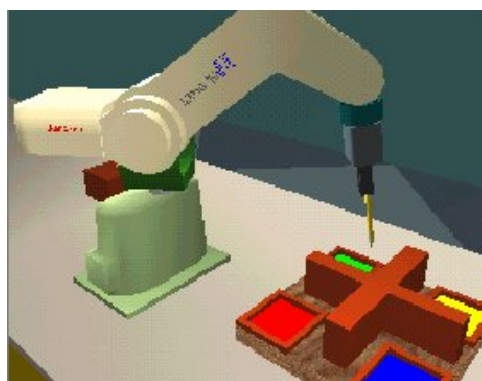


Figura 11: Simulador VR Robot

O VR Robot dispõe de um AV 3D para simulação, sendo possível fazer a integração com o

robô real. Neste simulador o usuário pode programar tanto de forma não-textual, através de uma interface similar ao TP, chamada de *VR teachbox*, quanto de forma textual, com os comandos do robô, permitindo que os programas criados no simulador sejam enviados ao robô.

Juntamente com o simulador existe uma biblioteca de modelos 3D, que podem ser colocados na célula de simulação para tornar o ambiente similar ao real, simulando também a detecção de colisão com os objetos. Além disto, o ambiente permite simular dois robôs ao mesmo tempo, o que é útil para testar algum tipo de trabalho colaborativo entre os robôs.

5.1.3 Simulador HandlingPRO

Para o caso de não ser utilizado nenhum robô real, existe o simulador HandlingPRO, da empresa FANUC Robotics [3], visto na Figura 12, baseado nos robôs da empresa FANUC. Este simulador possui um ambiente 3D para criação de células de trabalho, que permite tanto o uso dos objetos que acompanham o programa quanto os modelados pelo usuário, em ferramentas de CAD. Com isto é possível criar ambientes com mais de um robô e com todos objetos que ficam em volta do robô no ambiente real, possibilitando a verificação de colisão com objetos ou até mesmo outros robôs durante a execução da simulação. Como no caso do RoboCell, este programa permite também a edição dos programas através do TP. Além disso, uma simulação pode ser gravada em um arquivo de vídeo no formato AVI.

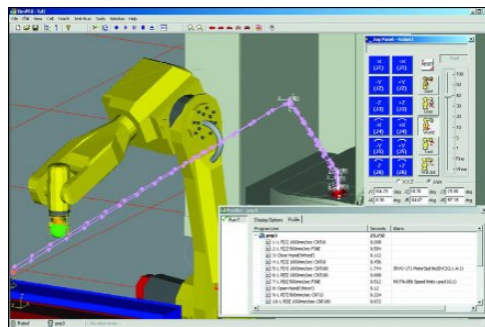


Figura 12: Simulador HandlingPRO

5.2 Sistema de realidade virtual para ensino de robótica

As formas tradicionais de se programar robôs textualmente podem tornar-se pouco intuitivas para usuários iniciantes, pois o operador necessita conhecer a linguagem de programação do robô e ainda deve ser capaz de resolver o problema através de algoritmos, existindo a necessidade de realizar muitos testes até que o robô realize o movimento esperado. Outro fator complicador é a existência de linguagens diferentes, dependendo do fabricante do robô. Já na

forma não-textual, com o uso de um TP, fica difícil criar movimentos complexos.

Procurando melhorar a forma de programar um robô, Coelho [12] desenvolveu uma aplicação para programação de um robô a partir do rastreamento do movimento da mão do usuário. Este sistema foi desenvolvido para atuar em conjunto com o robô Scorbobot-ER VII, que possui cinco articulações e trabalha com a linguagem de programação ACL. Para movimentar o robô o sistema capta a posição da mão do usuário e gera, a partir desta posição, os comandos ACL necessários para levar o robô até esta posição. A comunicação com o robô é feita via porta serial. Para utilizar o ambiente o usuário necessita de uma luva de RV, que permite captar informações para girar e abrir ou fechar a garra do robô, e um rastreador de posições, para captar a movimentação da mão no espaço tridimensional. A arquitetura do ambiente pode ser vista na Figura 13.

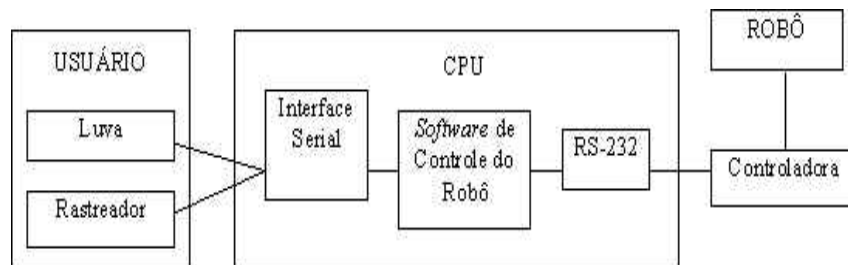


Figura 13: arquitetura do ambiente (extraído de [12])

A taxa de atualização do rastreador é bem superior à taxa permitida pelo protocolo da porta serial, então, para tornar possível esta comunicação é necessário realizar uma filtragem nos dados. A filtragem é feita em duas etapas, primeiro eliminando-se os pontos iguais ou muito próximos ao último ponto enviado, assumindo-se desse modo que o usuário manteve a mão parada por algum tempo. Em seguida são eliminados os pontos colineares desnecessários, pois formando uma reta apenas os extremos são necessários. As trajetórias realizadas podem ser armazenadas, para não ser preciso executar o mesmo movimento várias vezes.

Além da operação remota foi criado um ambiente de RV, visto na Figura 14, que permite visualizar a trajetória do robô no próprio ambiente, utilizando o monitor ou um óculos de RV, onde pode ser acoplado um rastreador de posições, para permitir girar em torno do robô virtual.

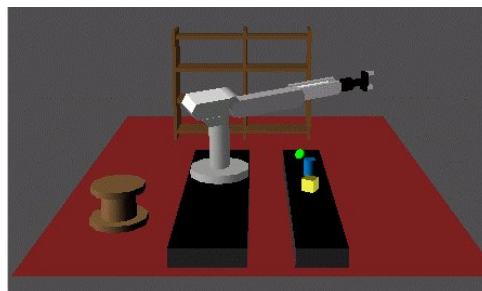


Figura 14: visualização do robô no ambiente virtual (extraído de [12])

Com a criação deste ambiente não é necessário ter o robô físico sempre que se desejar criar uma trajetória, além de permitir que a trajetória seja testada no simulador antes de enviar para o robô, o que diminui o risco no caso de ser cometido um engano.

5.3 Internet (Simulado)

Como alternativa de aplicações de simulação mais acessíveis, foi criado por Martin [29] um simulador de robótica que pode ser executado através de um navegador de Internet, em qualquer computador. O simulador foi criado utilizando a linguagem VRML 2.0 [16], que pode ser executada em qualquer navegador, necessitando apenas de um *plugin*. A dificuldade encontrada neste caso foi na manipulação do robô, pois VRML não permite a utilização de um *mouse* 3D. Para resolver este problema foi criada uma interface com ícones, que representam as ferramentas que permitem o controle do robô através do *mouse*. Esta interface pode ser vista na Figura 15.

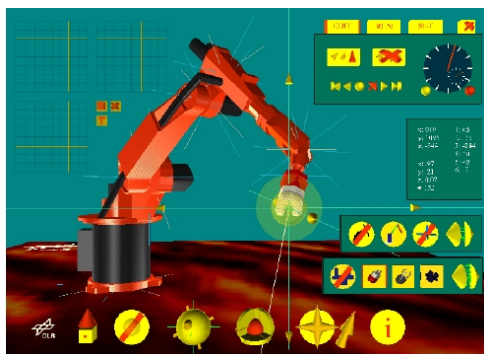


Figura 15: Tela da simulação em VRML (extraído de [29])

Nesta aplicação, todas as movimentações do robô são iniciadas a partir de eventos, chamados a cada mudança de ângulo ou posição. Cada evento contém as informações necessárias para realizar os movimentos desejados, que podem ser feitos de duas formas, definindo diretamente a posição ou rotação de cada junta ou então determinando a posição para onde o robô deve ir. Dessa forma, para posicionar a ferramenta corretamente são utilizados os cálculos de cinemática direta ou inversa, de acordo com o tipo de movimento desejado.

Nesta aplicação, os autores pretendem em uma versão futura possibilitar a realização de experimentos reais. Para isto seria gerada uma visualização do robô real, através de imagens capturadas por uma câmera. Neste caso, a interação com um robô real se daria através de um servidor conectado ao robô real, em um laboratório remoto, que faria a leitura dos eventos gerados pelo usuário e os passaria para o robô real, movendo o robô virtual a cada movimento do robô real, através da geração de um evento no aplicativo VRML.

5.4 Experimentos de controle de robôs reais pela Internet

Aplicações que além de simular a programação do robô permitem executar o programa no robô real, tornam o aprendizado mais rico. Existem formas de tornar possível a execução de programas no robô real através da Internet. Com este tipo de aplicação, é criado um laboratório real acessível de qualquer parte do mundo. Uma aplicação deste tipo foi criada por Riko [30].

Nesta aplicação foi criado um simulador em VRML, visto na Figura 16, com uma célula de trabalho semelhante a uma célula real. Neste simulador foi implementada a detecção de colisão entre o robô e os objetos do cenário, utilizando a biblioteca *V-Collide* [17], pois além de tornar a aplicação mais real, evita colisões entre o robô real e os objetos, o que não poderia ser controlado à distância em tempo hábil. Para mostrar os modelos em VRML e os movimentos no AV, foi utilizada a biblioteca VRaniML [6].

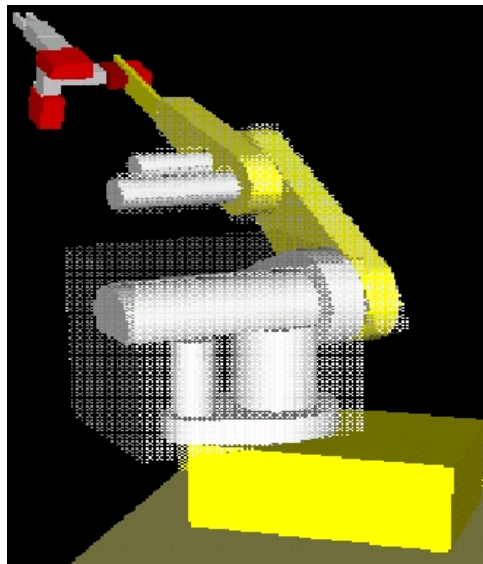


Figura 16: Interface VRML com as regiões de colisão (extraído de [30])

O robô utilizado foi o MA2000, um braço mecânico de seis eixos fabricado pela empresa *TecEquipment Ltd*. Foi feita uma integração entre o programa do robô, as bibliotecas e a interface VRML, possibilitando a execução remota. A interface com o robô real ocorre através do envio do arquivo com a tarefa, através de um formulário em uma página PHP. A Figura 17 mostra a organização do programa.

Quando o arquivo é enviado, um *script CGI* verifica a autenticação do usuário e em seguida verifica a disponibilidade do robô. Caso o robô esteja disponível, é possível executar o programa imediatamente e acompanhar a execução através da videoconferência, com um vídeo de três a quatro quadros por segundo, a fim de diminuir a largura de banda e permitir que usuários com conexão lenta também possam utilizar a aplicação. Este vídeo possui um atraso de cerca de dez segundos, o que segundo o autor, não pode ser evitado devido ao tempo gasto para codificar o vídeo. Além das imagens, informações como por exemplo, erros de posicionamento e tempo

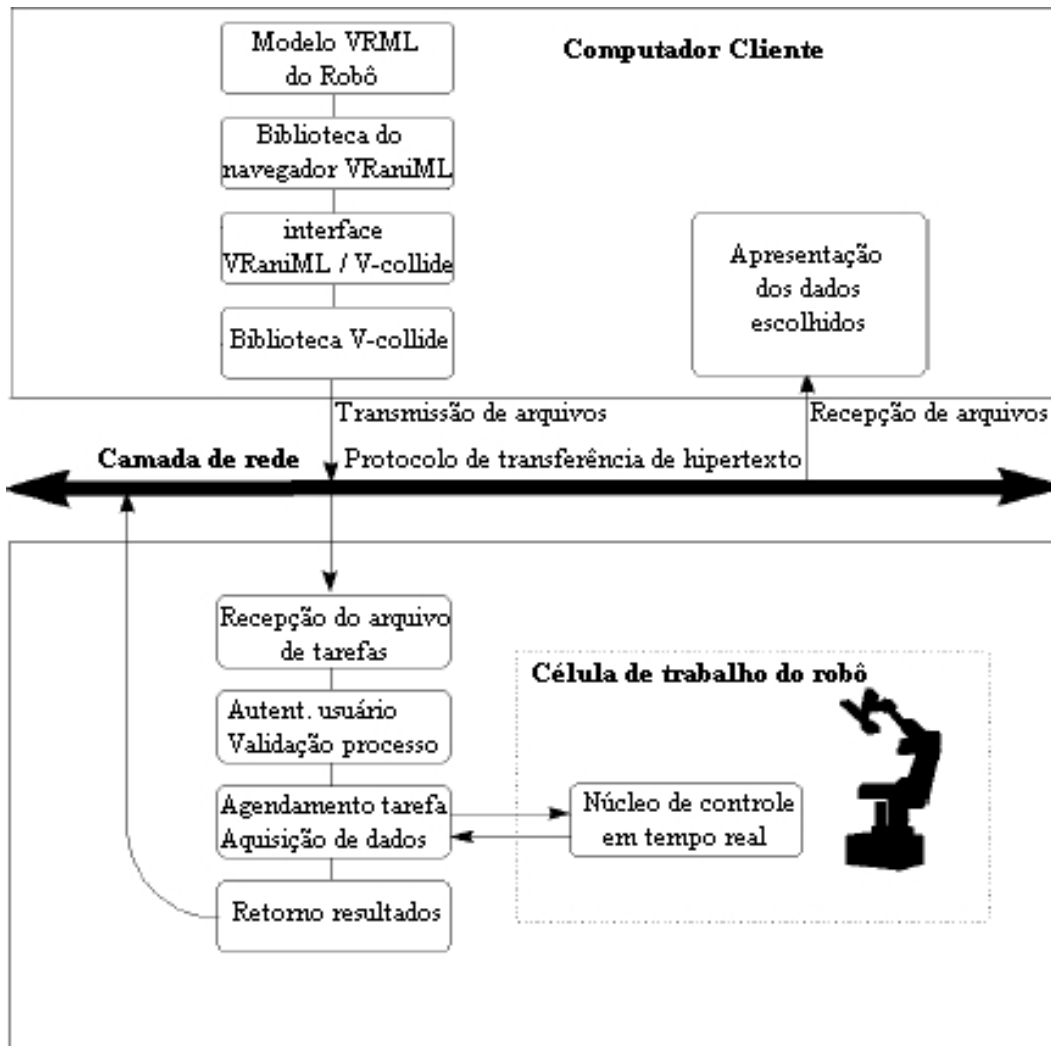


Figura 17: Interface entre o servidor e o cliente (extraído de [30])

decorrido são exibidas durante a tarefa, e ao final da tarefa é exibido um arquivo de log.

Quando o robô não estiver disponível, é possível colocar a tarefa na fila, para ser executada mais tarde. Neste caso, um vídeo é gravado mostrando a execução da tarefa, podendo ser visualizado pelo usuário posteriormente.

6 Ensino de Robótica nas Escolas

Para o desenvolvimento do presente trabalho, foram pesquisadas três escolas que ensinam a programação de robôs: (1) Faculdade de Tecnologia (FATEC) SENAI POA, curso superior em Automação Industrial e curso Técnico em Eletrônica, (2) Laboratório CIM - Faculdade de Engenharia Mecatrônica - Pontifícia Universidade Católica do Rio Grande do Sul, e (3) Escola de Educação Profissional SENAI Ney Damasceno Ferreira - Curso Técnico em Mecatrônica, Gravataí-RS. Nestas escolas são trabalhados conceitos como cinemática direta, cinemática inversa e definição de pontos de referência para a criação de programas.

A metodologia de programação usada nas escolas pesquisadas baseia-se principalmente na definição destes pontos de referência para se criar trajetórias do robô. Segundo esta metodologia, primeiramente o robô deve ser posicionado com o TP no ponto desejado, que é então gravado em uma posição de memória do robô. Nos programas, quando for necessário movimentar o robô para esta mesma posição, utiliza-se apenas o endereço da memória em que o ponto foi gravado.

Das escolas pesquisadas, apenas a FATEC utiliza o simulador RoboCell no laboratório. Devido ao alto valor das licenças, é utilizado um simulador para cada dois alunos. Nesta escola existem três robôs Scorbot ER-4 que são utilizados apenas para testar os programas prontos, depois de funcionarem no simulador. Estes robôs permitem controlar até dois dispositivos externos, no caso, a escola possui apenas dois dispositivos adicionais, sendo uma mesa eletropneumática da marca Festo, com coordenadas X, Y, Z e uma ferramenta de sucção, e uma bancada eletropneumática também da marca Festo.

A PUCRS possui um Laboratório de Manufatura Integrada por Computador (CIM), visto na Figura 18, onde são ministradas duas disciplinas: “Programação de Robôs e Sistemas Robotizados” e “Sistema de Fabricação”. O laboratório é composto por um sistema de manufatura flexível completamente controlado por computadores. O sistema contém seis estações para manufatura, controle de qualidade, montagem de peças e produtos. As estações de manufatura e montagem são integradas por meio de um sistema de transporte automatizado de robôs. Programas específicos integram todos os aspectos do planejamento e operação do sistema, desde o gerenciamento de pedidos de clientes até o controle de dispositivos e máquinas CNC. O laboratório CIM engloba, de forma multidisciplinar, todos os aspectos de um sistema moderno de produção.

A primeira estação é um robô que funciona como um almoxarifado automatizado, onde podem ser buscadas e guardadas peças. As estações dois e três compõem um sistema flexível de manufatura, e são compostas por um robô ER-IX, um robô ER-VII, um torno e uma fresa



Figura 18: Laboratório CIM (extraído de [5])

CNC. O torno CNC pode ser acionado pelos robôs, que podem abrir ou fechar a porta do torno e iniciar um programa gravado na sua memória. A quarta estação é utilizada para controle de qualidade e é composta por um robô MK3, um paquímetro digital e um altímetro. A quinta e última estação é a de montagem, composta por um robô SCARA ER14, visão, mesa XY e uma Parafusadeira Pneumática, permitindo selecionar o local para o qual as peças serão enviadas, soltando no local correto ou então mandando de volta para o almoxarifado.

Na PUCRS foi desenvolvido um simulador textual, capaz de compilar os programas e simular a execução sem representação gráfica. Um dos objetivos deste simulador é eliminar os erros de sintaxe do programa antes de submetê-lo ao robô. Este programa possui também um terminal, que pode trabalhar em dois modos, sendo o primeiro deles conectado ao robô real, enviando os comandos digitados pelo usuário e exibindo as respostas do robô, e o segundo modo consiste em emular o terminal do robô real, através de uma simulação sem visualização gráfica.

Na escola do SENAI de Gravataí, além de dois robôs Scrobot ER-VII, fabricados pela empresa israelense Eshed Robotec [7], existem duas esteiras, que podem ser controladas pelos robôs, através dos programas em ACL, sendo ligados a uma saída do robô, e uma mesa de experiência fabricada na própria escola, que possui quatro saídas e também é ligada ao robô, indicando se existe ou não um objeto sobre cada um de seus sensores.

A título de exemplo, no Anexo B existe uma tabela que mostra o conteúdo de robótica ensinado nesta escola, mostrando os objetivos, assuntos tratados e os recursos didáticos utilizados

em cada aula. De um curso de vinte e sete aulas, quatro (15%) são dedicadas ao conhecimento do TP e a marcação de pontos e doze (44%) são dedicadas à programação, o que indica que a ênfase está na programação do robô com programas em linguagem ACL, a fim de controlar o robô e os periféricos a ele associado.

Apesar de ser ensinada programação nas aulas posteriores ao ensino do uso do TP, a partir do início do uso deste dispositivo, sempre será necessário usá-lo para viabilizar a programação, pois os exercícios passados exigem a manipulação de peças e controle de periféricos ligados ao robô, sendo que a posição para buscar as peças é descoberta com o TP.

Praticamente todas escolas de robótica do SENAI utilizam robôs da linha Scorbot. Apesar do uso de um simulador enriquecer o ensino, muitas escolas não incluíram sua utilização na metodologia de ensino, utilizando apenas a chamada programação OFF-LINE, que consiste em criar o programa em um editor de texto qualquer, realizando o envio ao robô através de um computador conectado à ele. Isto acontece porque o programa que acompanha os robôs da linha Scorbot, chamado Scorbases [19], permite a criação de programas somente em uma linguagem semelhante à língua inglesa (ou portuguesa ao se trocar o idioma), não permitindo a programação em ACL, linguagem nativa dos robôs Scorbot, que é ensinada nestas escolas. O programa não possui uma visualização gráfica, e o TP comanda diretamente o robô real.

Uma alternativa seria adotar o simulador RoboCell, porém o alto valor das licenças inviabiliza sua adoção. Para tentar resolver este problema, o SENAI-RS optou por desenvolver seu próprio simulador de robótica com visualização 3D. Este simulador, chamado de Asimov, foi desenvolvido durante o ano de 2001, no Núcleo de Informática Educacional - NIED, e pode ser visto na Figura 19. Este simulador será detalhado aqui por ter servido de modelo para o desenvolvimento do VR-Robotics, que procurou solucionar alguns problemas apresentados pelo ASIMOV.

O Asimov possui quatro ambientes: Visualização 3D, Teach Pedant, Editor de Células e Cenários e o Terminal de Programação ACL.

Na **Visualização 3D** é exibido o AV 3D, contendo o robô e os outros objetos, conforme o ambiente atual. O **Teach Pendant**, permite manipular o robô e seus periféricos e gravar posições de memória. O **Editor de Células** permite criar e editar novos cenários e o **Terminal de Programação em ACL** é composto pelo terminal de controle do robô e pelo editor de programas, sendo que um programa deve ser escrito no editor e enviado ao robô pelo terminal, que ainda possui os comandos adicionais do robô real, como listar e executar programas, declarar variáveis globais, etc.

Nenhuma escola do SENAI adotou o Asimov como ferramenta nos cursos, devido a alguns problemas que dificultam seu uso, como segue:

- problemas na execução de comandos no Terminal de Programação ACL;
- tamanho reduzido da área de Visualização 3D, com percepção tridimensional pobre;

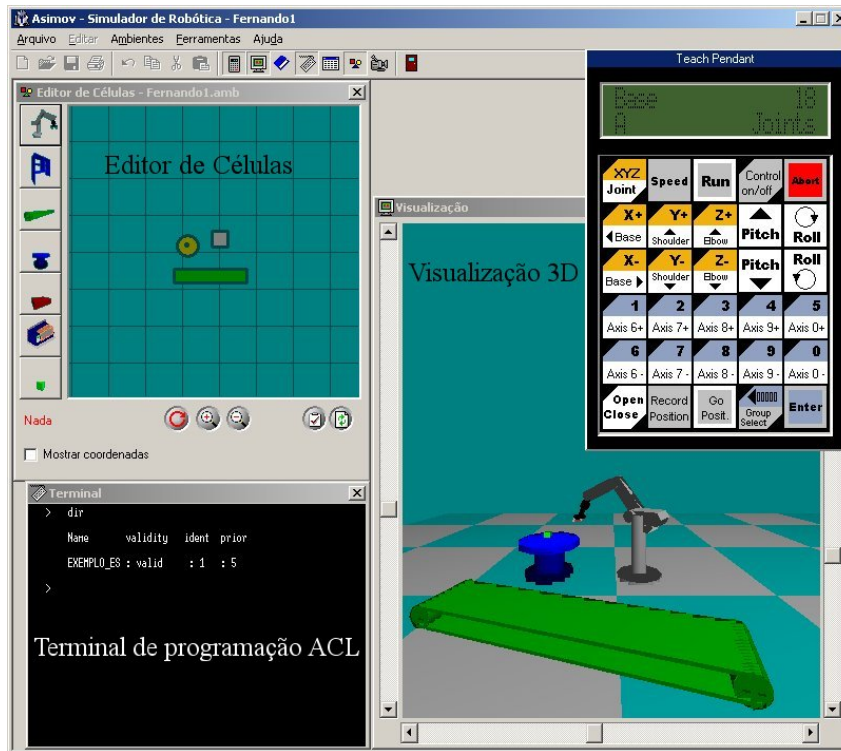


Figura 19: Simulador Asimov do SENAI-RS com seus quatro ambientes

- dificuldade de navegação no cenário, misturando o uso de mouse e barras de rolagem, e não permitindo o deslocamento lateral do usuário;

A conjugação de todos estes problemas inviabilizou o uso do Asimov nas escolas, pois havia grande dificuldade para marcar os Pontos de Referência utilizando o *software*.

No SENAI de Gravataí, por não adotar o Asimov, ainda ensina-se os alunos a programarem no modo OFF-LINE. Esta metodologia mostra-se bem mais trabalhosa, pois os alunos possuem pouco tempo para operar o robô, sendo que no caso do programa conter erros, os alunos precisam consertá-los e esperar novamente para testar no robô.

Em face dos problemas apresentados para adoção de um simulador gráfico nas escolas do SENAI, inicialmente se pretendia solucionar os problemas adaptando o simulador ASIMOV e inserindo nele recursos de RV, mas isto não foi possível pois exigiria uma remodelagem completa do programa, além de exigir o uso de placas de vídeo especiais com suporte a estereoscopia, custando mais caro que uma licença do RoboCell.

Constatou-se então a necessidade do desenvolvimento de um novo simulador, capaz de reduzir os problemas citados acima, em especial, no que diz respeito à forma de navegação e ao posicionamento da garra de maneira mais precisa.

Como visto, as escolas de robótica em sua maioria possuem necessidade de um simulador gráfico eficiente e barato. Para tentar resolver este problema, foi desenvolvido um simulador, apresentado no capítulo seguinte.

7 Ambiente Virtual VR Robotics

O AV denominado VR Robotics, visto na Figura 20, foi desenvolvido com o objetivo de criar um simulador que pudesse ser empregado nas escolas de robótica, possuindo uma visualização adequada e uma forma de interação simples e intuitiva. Isto porque simuladores disponíveis nas escolas apresentaram problemas de visualização e navegação, por isto este simulador foi criado utilizando técnicas para melhorar a visualização e navegação pelo ambiente, empregando diversos dispositivos, para que fosse possível testar a melhor forma de visualização e rastreamento, permitindo assim que se avaliasse qual conjunto de dispositivos melhor resolvia cada problema.



Figura 20: Programa VR Robotics com suas três janelas

No VR Robotics o aluno pode basicamente encontrar os pontos de referência, movendo o robô com o TP até atingir a posição desejada, que é então gravada em uma posição de memória do controlador do robô, podendo ser utilizada posteriormente nos programas ACL.

Este ambiente foi desenvolvido com base no Asimov, descrito no capítulo anterior e também simula o braço mecânico Scorbot ER-VII. Este robô utiliza motores elétricos como tecnologia de movimento e possui duas formas de programação, a forma não textual, que é do tipo *Lead-through* (através do *Teach Pendant*) e a forma textual, com a linguagem ACL, que é orientada a manipuladores. Porém, no VR Robotics a operação do robô é feita apenas através do TP, pois não foi implementada a execução de programas ACL nem o terminal do robô, sendo necessário

a existência de uma integração entre o VR Robotics e o Asimov, que é capaz de realizar estas duas tarefas. Para isto, os pontos gravados com o TP utilizando o VR Robotics são salvos em um arquivo, sendo então utilizados nos programas ACL simulados no Asimov.

Além do VR Robotics foram desenvolvidas aplicações adicionais para auxiliar seu uso, que podem ser vistas no Anexo A. A principal destas aplicações é o Config, que permite informar o conjunto de dispositivos que serão utilizados com o VR Robotics. As outras duas aplicações servem para rastreamento remoto, sendo uma delas com uma câmera e outra com um rastreador magnético.

Para tentar resolver o problema existente no Asimov, referente à área de visualização 3D pequena e com percepção tridimensional pobre, no VR Robotics foi empregado o uso de estereoscopia, que pode ser utilizada com três dispositivos de visualização diferentes.

Já a questão da locomoção e orientação do usuário dentro do AV foi tratada possibilitando a interação tanto com um *mouse* quanto com um teclado ou um *joystick*, sendo que qualquer um destes dispositivos possui acesso às principais funções para posicionamento do usuário e manipulação do robô. Notou-se durante a feira Globaltech 2006, onde o VR Robotics esteve em teste, que usuários que não possuem conhecimentos de informática, mas conhecem algum videogame, conseguem aprender facilmente a se deslocar pelo ambiente e manipular o TP utilizando um *joystick*.

Nas seções a seguir são apresentados a interface do VR Robotics e os dispositivos utilizados para rastreamento e visualização.

7.1 Interface do VR Robotics

O VR Robotics possui como janela principal o cenário 3D, existindo ainda outras duas janelas, uma para o Editor de Cenários, na janela superior esquerda e outra para o TP 2D, na janela superior direita. Nas seções a seguir serão explicadas as funções destas janelas.

7.1.1 Ambiente 3D

O ambiente 3D pode ser exibido em uma janela ou em tela cheia. No ambiente é possível navegar utilizando um *joystick*, *mouse* ou teclado, com comandos de deslocamento, rotação e troca de vistas pré-definidas.

7.1.2 Editor de Cenários

Para permitir a criação de novos ambientes, ou a edição dos ambientes já existentes no VR Robotics, foi criado um Editor de Cenários, que permite adicionar/remover/posicionar no ambiente elementos como robô, peça, esteira, mesa giratória, escaninho, mesa de experiência e torno CNC, cujas representações 3D podem ser vistas na figura 21, sendo que cada objeto está identificado na figura.

Para permitir uma simulação correta, cada ambiente deve possuir somente um robô, que pode ser associado, neste editor, a periféricos como a esteira e a mesa giratória, sendo então acionados através do TP. Este editor de cenários é semelhante ao do Asimov, e os ambientes gerados no VR Robotics podem ser utilizados no Asimov e vice-versa, permitindo assim uma integração entre estas duas ferramentas.



Figura 21: Representações 3D dos objetos do Editor de Cenários

7.1.3 Teach Pendant 2D

Este controle, visto na Figura 22, foi criado para permitir a manipulação do robô e obtenção dos pontos de referência. O TP é exibido em uma janela 2D, e permite a movimentação dos eixos do robô, das mesas e das esteiras, permitindo também a alteração na velocidade de movimentação destes dispositivos. Sua imagem é idêntica a do TP real. Com este controle também é possível gravar a posição atual do robô em uma posição de memória, bem como enviar o robô à uma posição previamente gravada. Além disto, é possível alterar a forma de movimentação do robô, que pode ser por juntas ou por eixos cartesianos X, Y, Z.

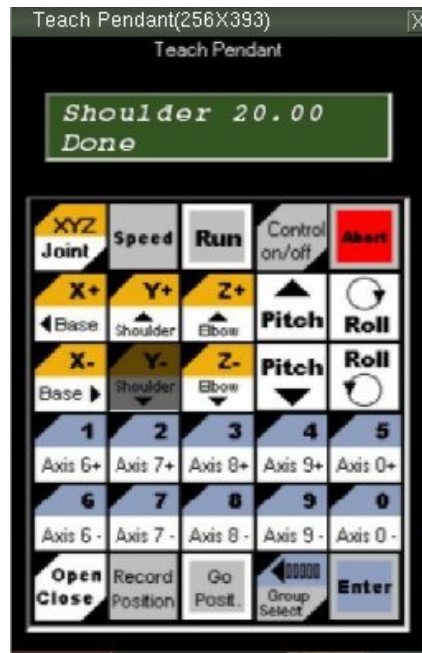


Figura 22: Teach Pendant

7.2 Rastreamento

Para resolver o problema de navegação pouco intuitiva existente no Asimov, foi utilizado rastreamento do usuário, como ferramenta para tornar mais intuitiva a interação com um AV. Com o **rastreamento da cabeça** do usuário, o mesmo pode mudar sua orientação e até mesmo seu posicionamento no AV apenas movimentando a cabeça, sem a necessidade de pressionar tecla alguma, auxiliando na visualização do ambiente de forma semelhante à real. Já com o **rastreamento da mão** do usuário, é possível movimentar um apontador, que pode ser usado, por exemplo, para selecionar um botão do TP.

Para testar qual dos dispositivos de rastreamento traz melhor resultado, no VR Robotics foram utilizadas três formas de rastreamento, como visto nas seções a seguir.

7.2.1 Rastreadores Magnéticos

O rastreador Polhemus Isotrack II possui dois sensores, podendo-se utilizar um deles para rastrear a cabeça do usuário e outro para a mão, como visto na Figura 23.

Cada sensor trabalha com seis graus de liberdade, fornecendo posição e orientação nos três eixos coordenados, porém alguns movimentos foram limitados. Para a cabeça do usuário foi desabilitada a translação no eixo vertical, enquanto que para a mão foi utilizada apenas a translação nos eixos vertical e horizontal, evitando que os usuários realizem movimentos que os façam perder a orientação no AV.



Figura 23: Rastreamento com o Isotrack II (2 sensores)

Outro tipo de rastreador magnético suportado é o rastreador de giro, que vem embutido em alguns modelos mais antigos de óculos Head Mounted Display (HMD). Este rastreador não permite a locomoção do usuário, mas reconhece o giro com boa precisão, permitindo que o usuário olhe em qualquer direção do cenário.

Um terceiro rastreador é o Cyber Track II, que tem características semelhantes ao rastreador do HMD. Este dispositivo pode ser usado apenas para rastreamento da cabeça do usuário, pois a seleção de um botão do TP exige o deslocamento da mão. Para utilizar este dispositivo, é aconselhável acoplá-lo ao óculos HMD, para que ele gire junto com a cabeça do usuário.

7.2.2 Rastreador Ótico

O rastreamento ótico é feito com o uso de uma câmera, como visto na Figura 24 e foi implementado utilizando-se a biblioteca ARToolkit [1].

Esta biblioteca faz o reconhecimento de marcadores (etiquetas), que devem ser presos nos pontos que se deseja rastrear. No caso foram usadas duas etiquetas, uma para o movimento da cabeça do usuário e outra para o movimento da mão. Cada sensor trabalha com seis graus de liberdade, porém se o sensor sair do foco da câmera, perde-se as informações de rastreamento. Por isto, ao utilizar esta forma de rastreamento deve-se evitar a locomoção dos usuários fora do campo de visão da câmera.

7.3 Visualização

Para testar qual dos dispositivos de visualização traz melhor resultado no processo de obtenção dos pontos de referência, no VR Robotics foram utilizados três diferentes tipos de dis-

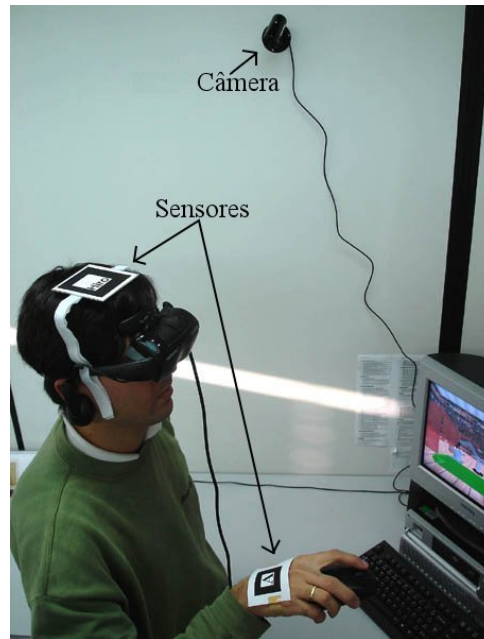


Figura 24: Rastreamento com uma câmera (2 sensores)

positivos, todos suportando o uso de estereoscopia, para melhorar a percepção tridimensional do ambiente. Apesar de ser possível executar o programa sem estereoscopia, é recomendável habilitar este recurso sempre que possível, pois a visualização deficiente foi justamente um dos maiores problemas encontrados no Asimov, e a estereoscopia foi aplicada no VR Robotics justamente para resolver este problema.

7.3.1 Monitor

O monitor foi utilizado por ser um dispositivo que está presente em qualquer computador, permitindo a utilização do simulador sem o uso de nenhum equipamento especial. A ativação da estereoscopia é feita através de um *Shutter Glasses*, como visto na Figura 25.



Figura 25: Uso do VR Robotics com monitor

Uma vantagem deste dispositivo é o uso do simulador em um ambiente em que os usuários já estão acostumados a trabalhar. Além disto, o monitor permite o uso da maior resolução de vídeo suportada pelo computador, o que nos outros dispositivos pode não ser viável.

7.3.2 HMD

O HMD foi utilizado por ser o dispositivo que proporciona uma maior imersão do usuário no AV, uma vez que a visão do mundo real fica bloqueada. Este equipamento, visto na Figura 26, permite que a visualização seja feita diretamente nele, utilizando ou não estereoscopia, pois possui um visor para cada olho. Isto possibilita maior interação, considerando que o usuário fica livre para girar a cabeça e até mesmo realizar pequenos movimentos sem perder a visualização.



Figura 26: Uso do VR Robotics com Óculos HMD

Com o uso do HMD, a visualização do ambiente é feita em primeira pessoa, dessa forma, os movimentos do usuário são refletidos no ambiente como se o mesmo estivesse dentro do cenário virtual, podendo-se, dependendo do tipo de rastreamento, olhar em qualquer direção e também locomover-se.

A vantagem de uso desta configuração está na facilidade de interação, sendo a mais intuitiva no deslocamento e orientação do usuário, uma vez que a visualização está no próprio óculos e o rastreamento adapta o vídeo do usuário de acordo com o movimento da cabeça. A desvantagem desta configuração está na resolução da imagem, que não pode ser muito alta por limitações do HMD. A maioria dos modelos de HMD trabalham com uma resolução fixa, de 640x480 ou 800x600 pontos.

7.3.3 Tela de Projeção

A tela de projeção foi testada como dispositivo de visualização por permitir a geração de imagens que podem ser visualizadas por várias pessoas ao mesmo tempo, como visto na Figura

27, porém, apenas um usuário pode utilizar um teclado, mouse ou *joystick* para comandar o robô.

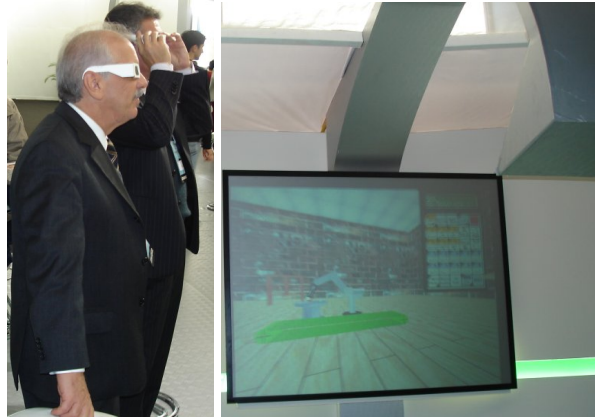


Figura 27: Uso do VR Robotics com Tela de Projeção, usuários à esquerda e tela à direita

Para ativar a estereoscopia neste dispositivo é necessário o uso de dois projetores em conjunto com lentes polarizadoras, de forma que as duas imagens sejam projetadas uma sobre a outra, em uma tela especial, que não desfaz a polarização quando recebe a projeção, permitindo que cada olho receba a imagem de apenas um dos projetores com o uso de um óculos polarizado de papel.

7.4 Escolha da configuração ideal

Cada um dos recursos apresentados neste capítulo tem vantagens e desvantagens. Uns são mais fáceis de usar mas tem um custo elevado, outros permitem que vários usuários observem o ambiente e outros são restritos a apenas um usuário.

Além disto, há a possibilidade de várias combinações entre dispositivos de rastreamento e dispositivos de visualização.

Para avaliar estas diversas alternativas tanto sob o aspecto de eficiência (precisão e rapidez para a obtenção dos pontos de referência) quanto de aplicabilidade (viabilidade econômica e didática) decidiu-se realizar um conjunto de testes com usuários.

Nestes testes foram feitas comparações de diversas configurações do VR Robotics entre sí, bem como comparações do VR Robotics com o Asimov. O detalhamento dos testes é apresentado no próximo capítulo.

7.5 Aspectos de implementação

O simulador VR Robotics foi implementado em C++, utilizando a biblioteca OpenGL para desenhar o cenário virtual, juntamente com a GLUT para criação da janela principal do programa. A biblioteca SmallVR¹ foi utilizada para o gerenciamento dos objetos no cenário virtual, carregamento dos modelos 3D e comunicação com alguns dos equipamentos de RV. A biblioteca ARToolkit [1] foi utilizada para o rastreamento ótico, exigindo ainda a utilização das bibliotecas *DirectShow Video Processing Library* (DSVL)² e DirectX para comunicação com a câmera. Para comunicação com o rastreador interno foi utilizada a biblioteca *World Toolkit* (WTK). Para a criação das janelas exibidas no cenário virtual foi desenvolvida uma biblioteca, pois nenhuma das bibliotecas encontradas atendia os requisitos necessários. O diagrama de classes simplificado do VR Robotics é exibido na Figura 28 e o diagrama de atividades simplificado é apresentado na Figura 29.

Para a resolução do problema da cinemática inversa foi realizada uma conversão da função existente no Asimov, que é desenvolvido em Delphi e resolve este problema corretamente. Para facilitar esta conversão, o modelo 3D do robô utilizado no VR Robotics possui os eixos com as mesmas dimensões do robô utilizado no Asimov.

Além do VR Robotics foram desenvolvidas aplicações auxiliares, sendo a principal delas a “config.exe”, criada para permitir a escolha dos parâmetros de execução do programa principal. Para executar o rastreamento por câmera no modo remoto foi criado o programa “server_camera.exe”, cuja função é fornecer a posição dos padrões para o programa principal. Já para o rastreamento com o Polhemus Isotrack Interno foi criado o programa “server_internal_track.exe”, cuja função é fornecer as informações de rastreamento. Estas aplicações são melhor detalhadas no Anexo A.

¹<http://grv.inf.pucrs.br>

²<http://ims.tuwien.ac.at/~thomas>

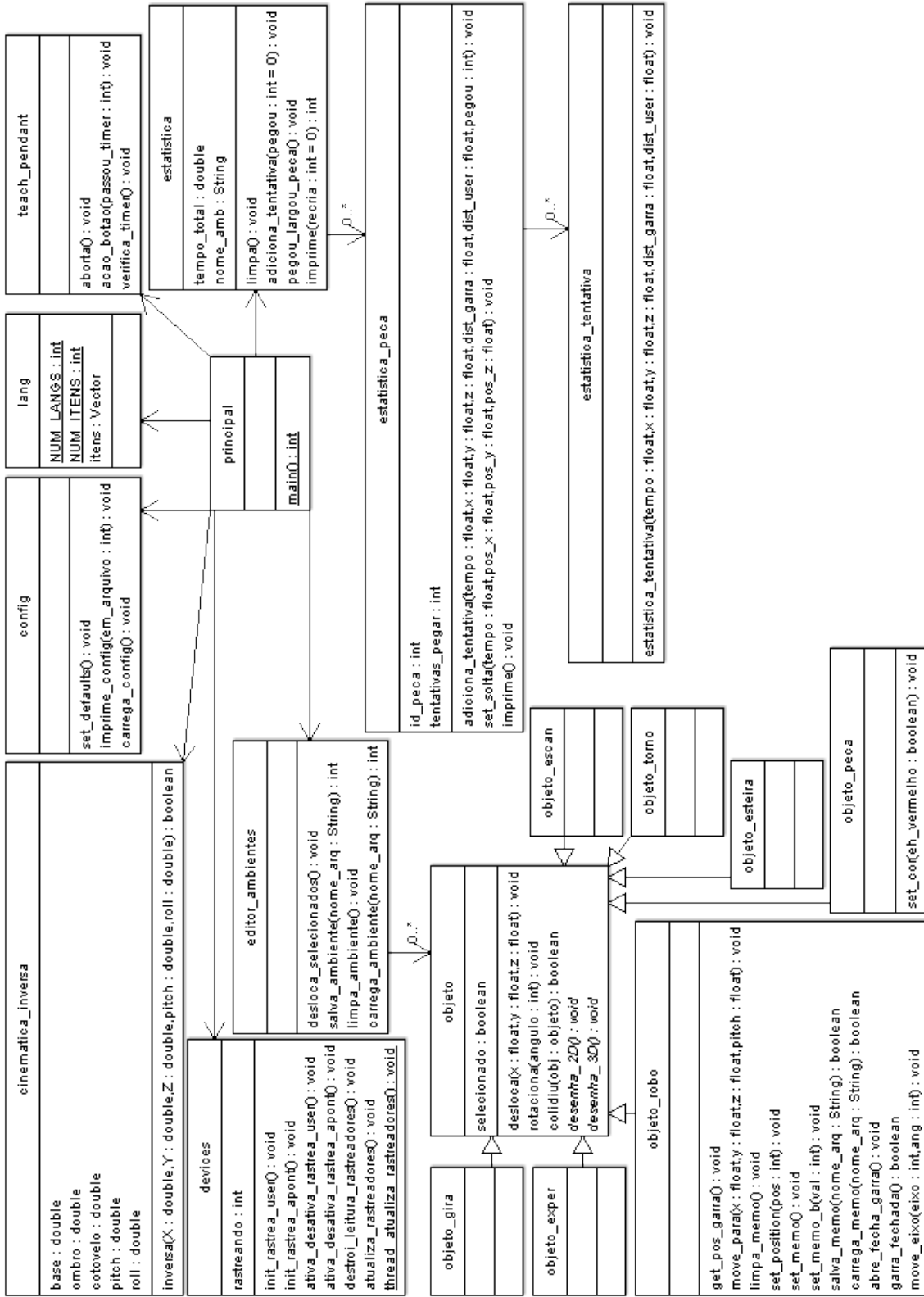


Figura 28: Diagrama de classes simplificado do VR Robotics

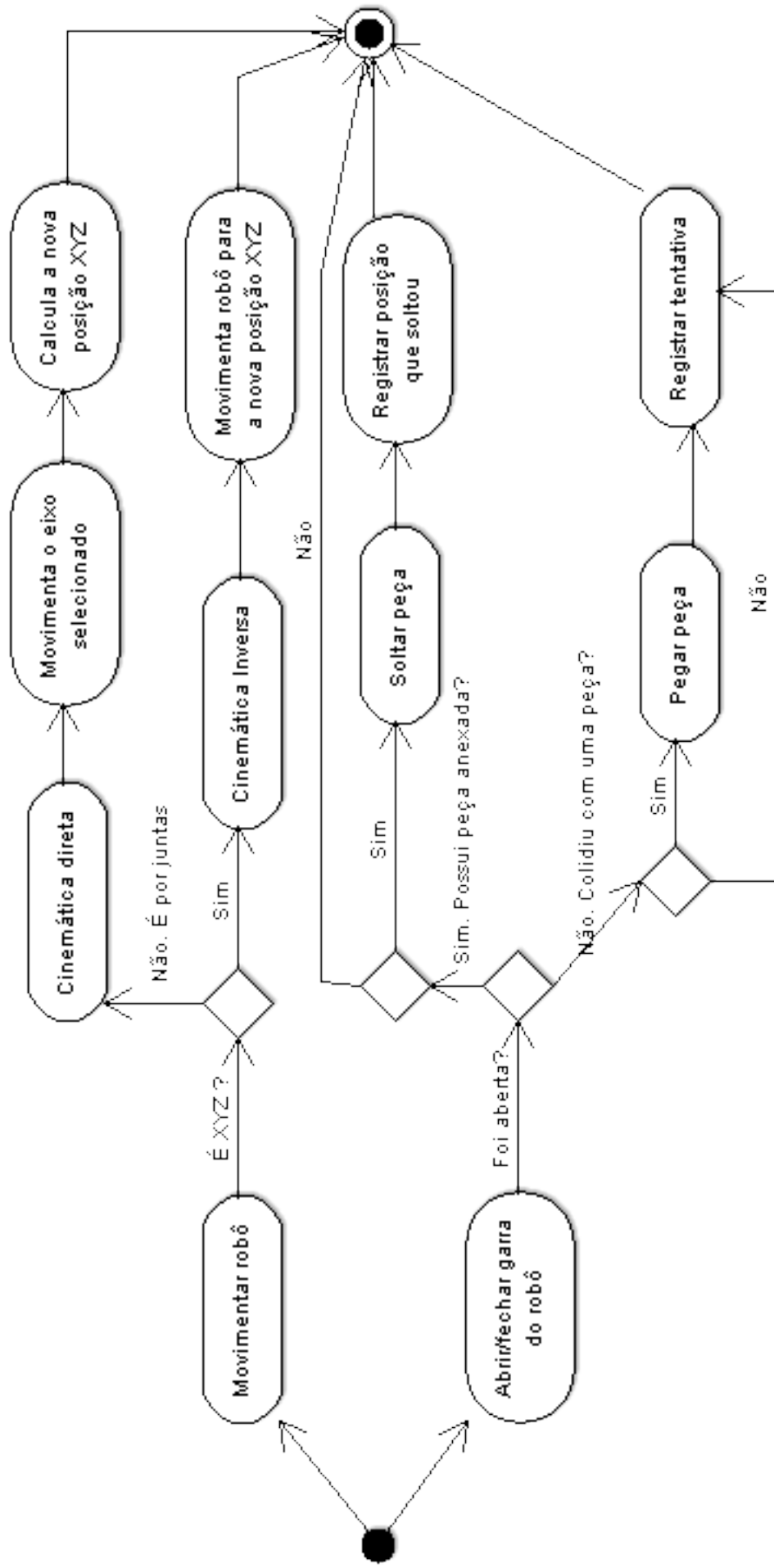


Figura 29: Diagrama de atividades simplificado do VR Robotics

8 Testes com Usuários

8.1 Objetivos

O objetivo geral deste trabalho foi realizar um estudo de técnicas de robótica e de RV para desenvolver um simulador que possa ser empregado nas escolas de robótica, possuindo uma visualização adequada e uma forma de interação simples e intuitiva.

Para concretizar este objetivo foi construído um simulador do robô Scrobot ER-VII. Neste simulador o usuário controla um robô virtual dentro de uma célula de manufatura¹, utilizando recursos de RV, como estereoscopia e rastreamento de posição.

Em seqüência ao desenvolvimento desta aplicação procedeu-se a realização de um conjunto de testes com o intuito de avaliar o desempenho do *software*, dadas várias configurações de dispositivos possíveis.

Este capítulo apresenta a metodologia usada nestes testes, bem como os resultados obtidos.

8.2 Plataforma de testes

Para realizar os testes foi criado um AV que representa uma célula de manufatura (Figura 30) no qual um usuário controlando um braço robótico deveria mover um conjunto de peças. Este ambiente foi testado com o Asimov e com o VR Robotics e possui do lado esquerdo duas mesas giratórias, uma esteira e uma mesa de experiência. Sobre as mesas encontram-se cinco peças, todas ao alcance do robô, que está no centro do AV, sobre uma esteira que foi utilizada para posicioná-lo perto de qualquer uma das peças. Existem ainda outras cinco mesas giratórias posicionadas no lado direito do ambiente.

O ambiente de testes foi criado de forma que exigisse a movimentação do robô sobre a esteira, pegando peças em diferentes alturas, profundidades e distâncias e soltando as mesmas em diferentes distâncias, fazendo com que o braço robótico tivesse que ser completamente esticado em alguns momentos ou fechado em outros. Também houve a preocupação de posicionar algumas peças em locais de difícil visualização, para que o usuário fosse obrigado a navegar no AV e posicionar-se perto da peça que pretendia pegar.

A tarefa executada consistiu em pegar cada uma das cinco peças do lado esquerdo da célula,

¹Célula de Manufatura consiste em um ambiente de produção automatizado, com o uso de um ou mais robôs e periféricos adicionais, que são controlados pelos robôs.

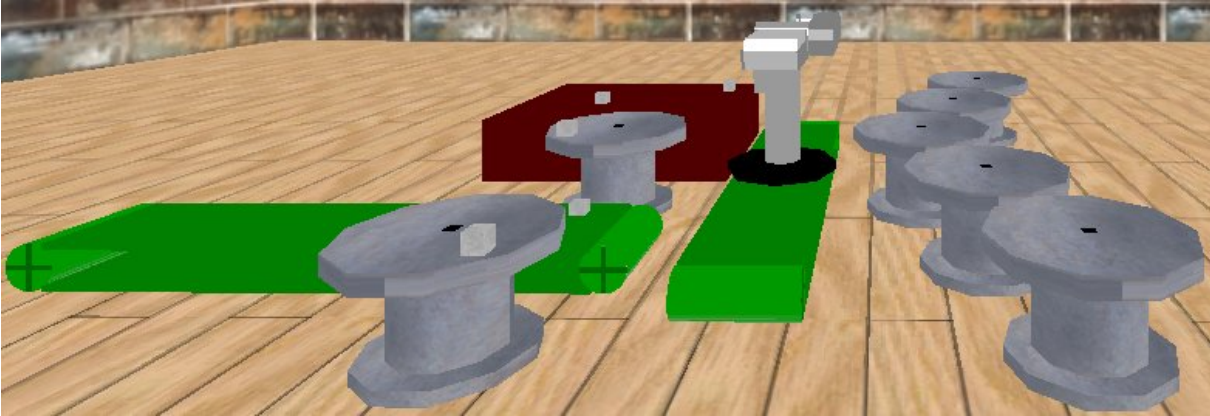


Figura 30: Ambiente de testes

e soltá-las sobre a mesa do lado direito que estivesse mais alinhada à posição da qual a peça foi pega. Na Figura 31 observa-se à esquerda cinco peças que deveriam ser movidas para cinco posições sobre as mesas à direita. Para manipular o robô durante a execução desta tarefa, os usuários utilizaram o TP.

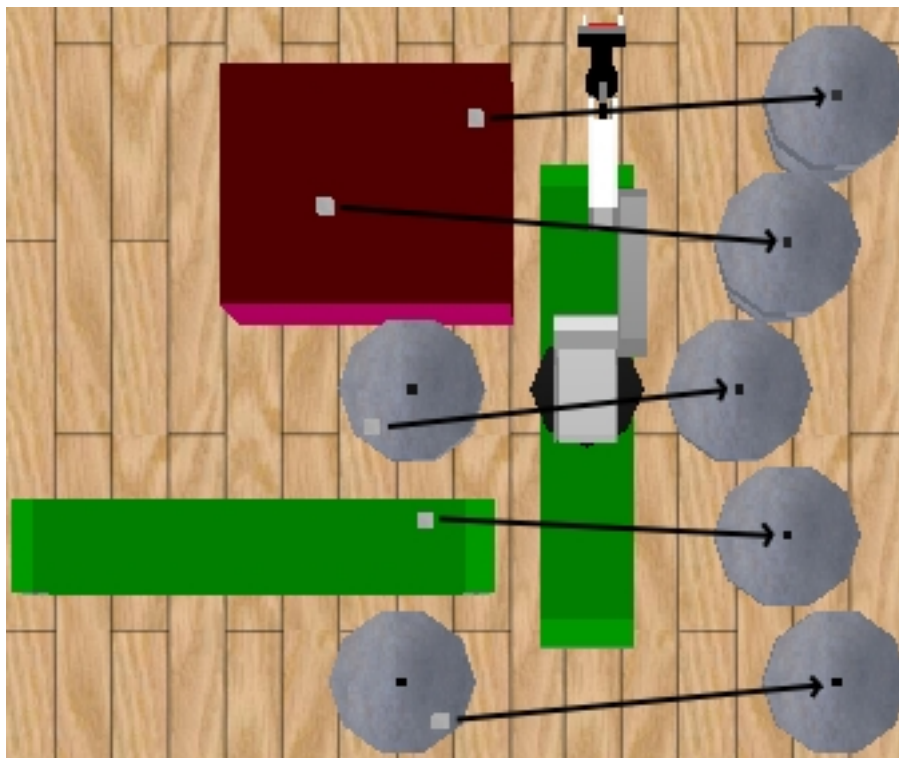


Figura 31: Vista superior do ambiente de testes do VR Robotics

Para que a tarefa fosse realizada com a maior precisão possível, aconselhou-se os usuários para que adotassem as seguintes diretrizes:

- fechar a garra do robô somente quando desejassem pegar uma peça;
- soltar as peças o mais próximo possível do centro das mesas (realçadas com um retângulo preto);

- abrir a garra do robô somente quando a tentativa de pegar uma peça falhasse ou para soltar uma peça sobre uma mesa do lado direito do cenário;
- não pegar novamente uma peça que foi solta, mesmo que o posicionamento final não tenha sido satisfatório;
- realizar a tarefa o mais rapidamente possível, porém sem interferir nas regras anteriores.

A tarefa realizada exigiu que fossem criadas trajetórias para pegar e soltar as peças, sendo este o mesmo procedimento adotado para obtenção dos pontos de referência, utilizados para ensinar robótica nas escolas, levando o usuário à realização desta atividade de forma implícita.

Com a realização dos testes foram coletados dados quantitativos através de um relatório gerado automaticamente e dados qualitativos através dos questionários pré-teste e pós-teste, preenchidos pelos usuários. Estes questionários estão disponíveis nos Anexos C e D, e no Anexo F estão as estatísticas coletadas para obtenção dos resultados.

8.3 Testes-piloto

Com o objetivo de avaliar de forma geral as ferramentas de interação utilizadas no VR Robotics, primeiramente foram realizados testes piloto, utilizando quatro diferentes configurações para estes testes: monitor com e sem estereoscopia, HMD com estereoscopia e rastreamento, e tela de projeção com estereoscopia. Cada configuração foi testada com pelo menos três usuários.

Nestes primeiros testes havia dois tipos de TP, o 3D, que ficava dentro do AV (como um objeto virtual preso ao usuário) e o 2D, que ficava em uma janela da aplicação, sem sofrer movimentações em função dos movimentos do usuário.

Nas versões com RV, os usuários manipulavam o TP 3D utilizando um *joystick*, que controlava o movimento de um cursor sobre o TP, como visto na Figura 32. Para selecionar um comando, primeiro era necessário posicionar o apontador sobre o botão desejado, e acioná-lo com um botão do *joystick*.

Esta forma de interação, apesar de simples, mostrou que os usuários que a utilizavam levaram em média 80% mais tempo para executar a tarefa do que aqueles que utilizavam a versão com o TP fixo. Além disto, alguns usuários responderam nos questionários que o *joystick* gerava confusão por exigir o uso de muitos botões. Outros usuários responderam que eventualmente o robô ocultava parte do TP 3D, prejudicando a visão do mesmo.

Já na versão sem estereoscopia, os usuários manipulavam o TP 2D, visto na Figura 22, clicando com o *mouse* sobre os botões do mesmo, sem se preocupar com o botão selecionado.

O tempo foi considerado um fator importante, pois pode indicar maior facilidade de interação e conforto na utilização dos equipamentos. Além disto, é importante a realização da tarefa



Figura 32: Teach Pendant 3D utilizado nos testes-piloto

mais rapidamente nos casos em que houver quantidade de computadores inferior ao número de alunos. Considerando-se os resultados apresentados e a importância do menor tempo de execução foram feitas modificações na interface do TP, descritas na seção a seguir.

8.3.1 Modificações baseadas nos testes-piloto

A solução para reduzir o tempo de teste, além de resolver o problema da oclusão seria utilizar sempre o TP 2D, porém, o uso de janelas não é possível no modo tela cheia. Para resolver este problema foi criado um gerenciador de janelas em OpenGL, com o qual foram recriadas as janelas do editor de cenários e do TP, tornando a interação exatamente igual nas versões com ou sem RV, permitindo uma comparação mais precisa entre todas as configurações do programa.

Dessa forma, os testes com RV feitos antes desta conversão foram descartados, e os testes da versão sem RV foram mantidos, pois já utilizavam um TP acionado por clique do *mouse*.

8.4 Variáveis dos Testes

Para realização dos testes foram considerados dois tipos de variáveis, as independentes de usuário e as dependentes de usuário.

8.4.1 Variáveis Independentes

As variáveis independentes de um teste definem os aspectos destes que não dependem do desempenho do usuário. No caso deste projeto, são as configurações de testes possíveis, que no caso, dizem respeito ao conjunto de dispositivos utilizados, ao uso ou não de estereoscopia e ao uso do VR Robotics ou do Asimov. As variáveis independentes dos testes foram:

- tipo de dispositivo de visualização: monitor, HMD ou tela de projeção;
- uso de estereoscopia;
- uso de rastreamento da cabeça do usuário;
- utilização do VR Robotics ou do Asimov.

Os comandos de movimentação do usuário e controle do robô foram realizados sempre com o *mouse*, pois durante a realização dos testes-piloto o uso do *joystick* exigiu maior tempo para a realização da tarefa, apesar de ter sido simples e de fácil aprendizagem por parte dos usuários. Dessa forma, como pode ser visto na Tabela 1, foram utilizadas sete configurações de ambiente para realização dos testes.

Tabela 1: As sete configurações para os testes com os usuários

Config. Teste	Monitor	HMD	Projeção	Estereoscopia	Rastr. Usuário	Uso Asimov
1	X					
2	X			X		
3		X		X		
4		X		X	X	
5			X			
6			X	X		
7	X					X

Não foram utilizadas todas as combinações de configurações possíveis, pois algumas não faziam sentido ou eram semelhantes a outras existentes.

O rastreamento da cabeça do usuário foi utilizado apenas com o HMD, pois é o único dos dispositivos de visualização utilizados em que é natural a mudança na orientação da visualização do ambiente conforme a orientação da cabeça do usuário.

O HMD foi utilizado apenas com a estereoscopia ativada, pois sem o uso de estereoscopia a imagem seria semelhante à gerada no monitor, não obtendo melhora na qualidade.

8.4.2 Variáveis Dependentes

As variáveis dependentes são os parâmetros medidos a partir do desempenho dos usuários. Neste trabalho estes parâmetros foram divididos em duas categorias, de acordo com seu objetivo. Na Tabela 2 pode-se observar estes atributos agrupados por objetivos.

Tabela 2: Variáveis Dependentes

Objetivo	Parâmetro	Como é obtido
Avaliar o tempo para execução da tarefa	Tempo para pegar a peça	Tempo medido desde o momento que a tarefa iniciou ou a última peça foi solta, até a próxima peça ser pega
	Tempo para soltar a peça	Tempo medido desde o momento que o usuário pegou uma peça até o momento que a soltou
	Tempo para a execução da tarefa	Tempo medido do início ao final da tarefa
Avaliar a qualidade da visualização	Número de tentativas para pegar a peça	Cada vez que a garra do robô é fechada, é registrada uma tentativa de pegar a peça mais próxima à garra
	Posição que soltou a peça (precisão na colocação da peça)	A posição (X, Y, Z) da peça no momento em que a garra do robô foi aberta. A peça cai verticalmente, e a posição (X, Z) final foi utilizada para medir a distância em relação à posição correta (o centro da mesa giratória)
	Distância da garra à peça a cada tentativa	A distância de erro entre a garra e a peça ao tentar pegá-la, sendo zero quando a mesma foi pega
	Número de peças que não foram pegas	O número de peças que ao final da tarefa o usuário não conseguiu pegar
	Distância do usuário à peça a cada tentativa	A distância entre o usuário e a peça que tentou pegar, para avaliar se os erros tem relação com a distância da câmera

Estes dados foram obtidos de maneira automática, tanto no VR Robotics quanto no Asimov, gerando-se um relatório no final da execução. Para medir estes parâmetros no Asimov foi necessário alterar o programa, mediante autorização do SENAI, onde o mesmo havia sido desenvolvido.

A precisão na colocação das peças foi calculada medindo-se a distância entre a posição correta, que é o centro das mesas giratórias, e a posição registrada no relatório gerado pelo programa.

Algumas pessoas realizaram o teste em mais de uma configuração de dispositivos. Para isto, procurou-se alternar a ordem de configurações para estes usuários, a fim de não gerar um falso resultado, caso o segundo teste ficasse sempre mais rápido do que o primeiro.

8.5 Protocolo de realização dos testes

Os experimentos com os simuladores foram realizados dentro de um protocolo que objetivou dar um tratamento uniforme a todos participantes.

Foram realizados 29 testes no total, com 15 indivíduos (13 homens e 02 mulheres), sendo que cada usuário realizou dois testes, à exceção de um usuário, que realizou apenas um, sendo a distribuição destes testes apresentada na Tabela 3. A maioria dos usuários era formada por alunos de cursos de Ciência da Computação em nível de Graduação ou Mestrado, tendo, portanto, bons conhecimentos prévios de Informática.

Tabela 3: Distribuição dos testes realizados

Usuário	SEXO	Monitor sem estéreo	Monitor com estéreo	HMD sem rastrea-mento	HMD com rastrea-mento	Projeção sem es-téreo	Projeção com es-téreo	Asimov
06	M	X						X
07	M	X						X
09	M	X						X
11	M		X	X				
12	F		X	X				
13	M		X		X			
14	M		X		X			
15	M		X	X				
16	M		X	X				
17	M		X	X				
18	M					X	X	
19	F					X	X	
20	M					X	X	
21	M					X	X	
22	M				X			

As etapas do protocolo, apresentadas a seguir, não tinham uma duração máxima ou mínima pré-estabelecida, porém o tempo total da realização de cada experimento ficou sempre entre 15 e 45 minutos. As cinco etapas do protocolo são descritas a seguir.

Na primeira etapa, aplicação do questionário inicial, os usuários recebiam um questionário com perguntas sobre sua idade, frequência de utilização semanal de computadores e categorias de programas mais utilizados. No Anexo C pode-se observar a íntegra deste questionário.

A apresentação do experimento e dos equipamentos de RV era a segunda etapa do protocolo. Após responder ao questionário os usuários recebiam uma folha com a descrição do experimento e seus objetivos, vista no Anexo E. Após a leitura do documento procedia-se a apresentação dos equipamentos a serem usados durante os experimentos.

Na sessão introdutória ao AV, terceira etapa do protocolo, os usuários vestiam os equipamentos de RV, dependendo da configuração utilizada e recebiam algumas instruções básicas, a fim de permitir a interação com o ambiente e a manipulação do robô com o TP. Em seguida podiam interagir livremente em um AV criado para a ambientação. Esta fase durava, em média, cinco minutos e encerrava-se quando os usuários assim desejassem.

Na etapa quatro, onde se realizavam os testes propriamente ditos, os usuários eram inseridos no AV e a tarefa que deveria ser executada era novamente apresentada. A coleta das variáveis dependentes durante a execução da tarefa era feita automaticamente.

Após a conclusão da tarefa os usuários eram solicitados a retirar os equipamentos de RV e a responder o questionário de avaliação (Anexo D), em que podiam informar entre outras coisas, se haviam sentido algum tipo de desconforto durante a execução do mesmo, ou se havia algum comentário adicional.

Nas seções a seguir, são apresentados os resultados dos experimentos com todas configurações utilizadas.

8.6 Resultados dos testes

Nas seções seguintes são apresentados os resultados obtidos com base nas variáveis dependentes de usuários. As tabelas completas com estes dados estão no Anexo F.

8.6.1 Tempo Gasto

A Figura 33 mostra uma comparação do tempo médio para realização da tarefa em todas configurações testadas.

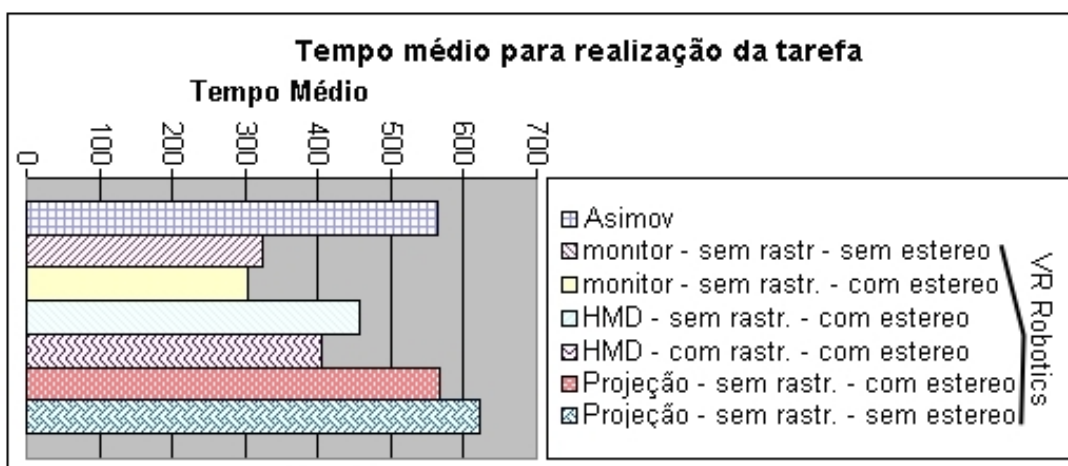


Figura 33: Gráfico do tempo médio para realização da tarefa

Analisando o gráfico é possível notar que o uso de estereoscopia diminuiu o tempo para realização da tarefa utilizando tanto um monitor quanto uma tela de projeção. O rastreamento da cabeça do usuário, utilizado apenas com o HMD, também diminuiu o tempo de realização da tarefa. A tela de projeção foi o dispositivo que obteve maior tempo para realização da tarefa, contudo este foi o dispositivo com a maior precisão, como pode ser comprovado na próxima seção.

A média de tempo para pegar as peças apresentou comportamento idêntico à média de tempo total da tarefa, como visto na Figura 34. Isto indica ser o monitor com estereoscopia a configuração que possibilita a realização da tarefa mais rapidamente.

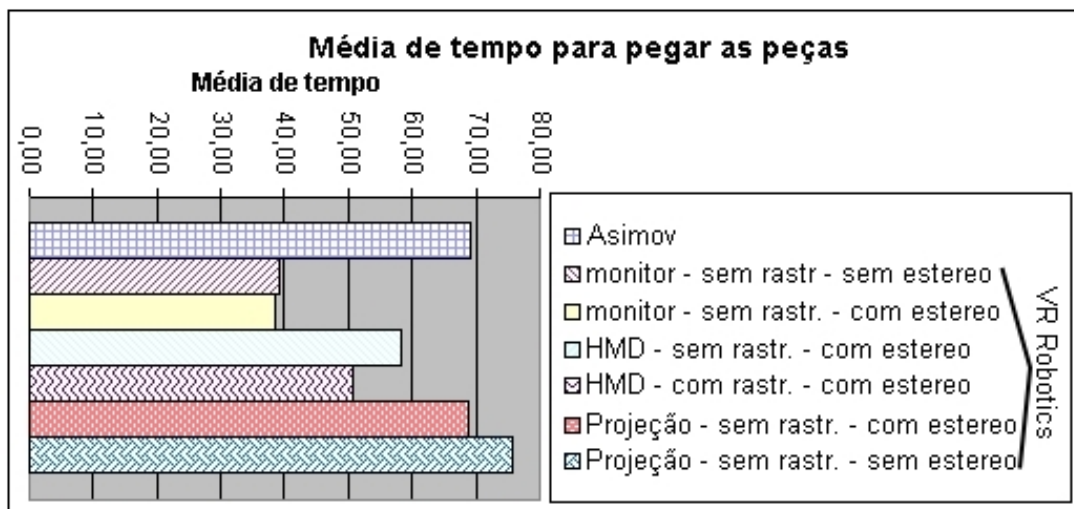


Figura 34: Gráfico da média de tempo para pegar as peças

8.6.2 Precisão no posicionamento das peças

Um dos objetivos da construção do VR Robotics foi a tentativa de diminuição do erro no posicionamento do braço robótico. Conforme pode ser visto na Figura 35 a configuração que apresentou a maior precisão foi a tela de projeção com estereoscopia. Já o HMD com rastreamento da cabeça do usuário foi a configuração com maior erro, o que pode indicar que mesmo que o tempo de realização da tarefa com rastreamento tenha sido menor, quando a interação é realizada com o *mouse* este dispositivo não ajuda, pois o usuário fica parado em frente ao monitor.

No geral a precisão na posição final das peças foi maior no VR Robotics do que no Asimov, o que indica que o objetivo de melhorar a precisão foi alcançado.

Outro fator que deve ser considerado ao analisar a qualidade da visualização é a distância da garra à peça quando se tenta pegar uma peça. Analisando a Figura 36 podemos perceber que o dispositivo que apresentou o pior resultado foi a tela de projeção sem estereoscopia,

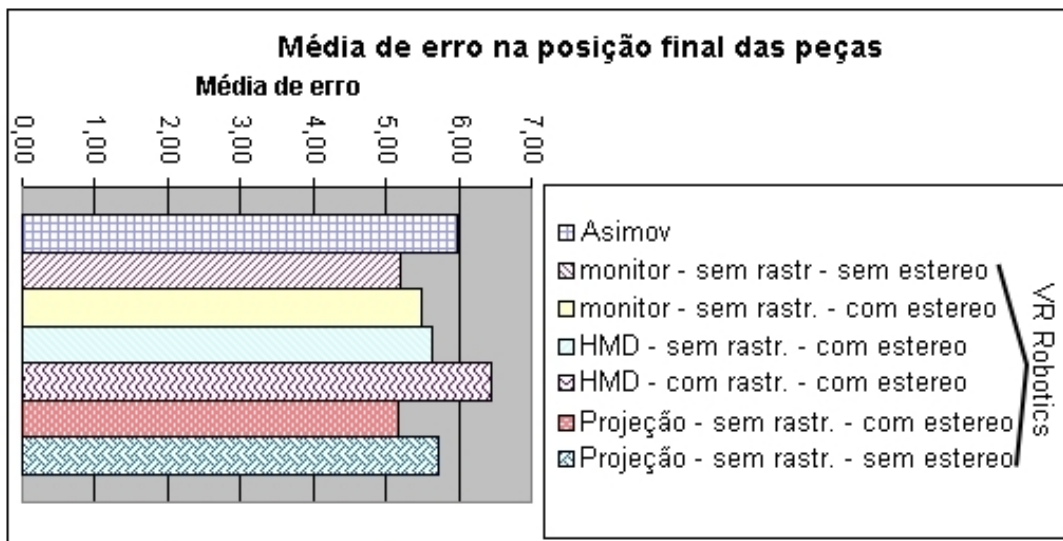


Figura 35: Gráfico da média de erro na posição final das peças

porém, neste mesmo dispositivo quando foi utilizada estereoscopia praticamente não houve erro ao pegar as peças, indicando que a estereoscopia auxiliou na realização da tarefa, o que é confirmado pelas respostas dos usuários nos questionários pós-teste.

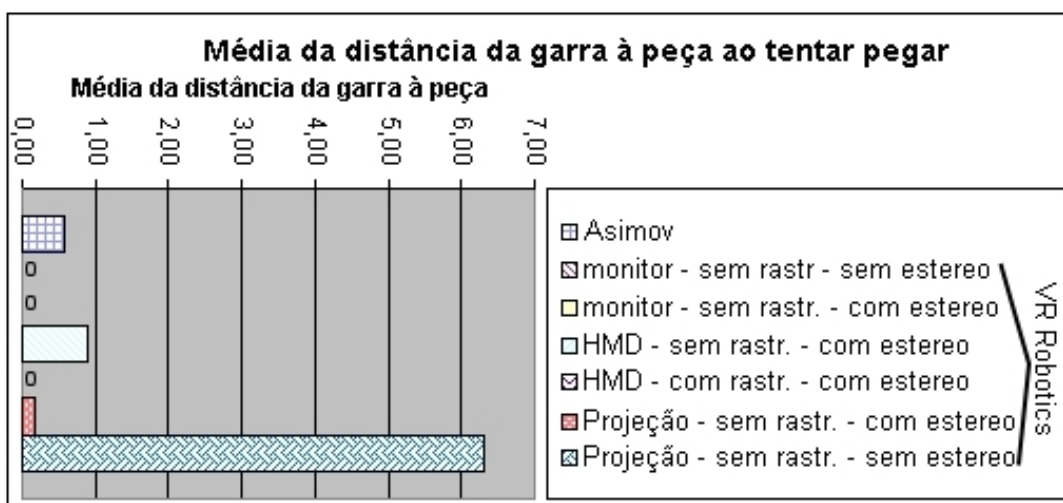


Figura 36: Gráfico da média da distância da garra à peça ao tentar pegar

8.6.3 Número de tentativas para pegar as peças

O número de tentativas para pegar as peças pode indicar a qualidade na visualização, pois tentativas erradas indicam que o usuário não conseguiu perceber que a garra não estava sobre a peça. Na Figura 37 podemos perceber que na metade das configurações com o VR Robotics foi necessário apenas uma tentativa para pegar as peças, mas a configuração da tela de projeção

sem estereoscopia e sem rastreamento necessitou um número maior de tentativas, o que indica que a tela de projeção deve ser utilizada com a estereoscopia ativada, para que a visualização seja adequada.

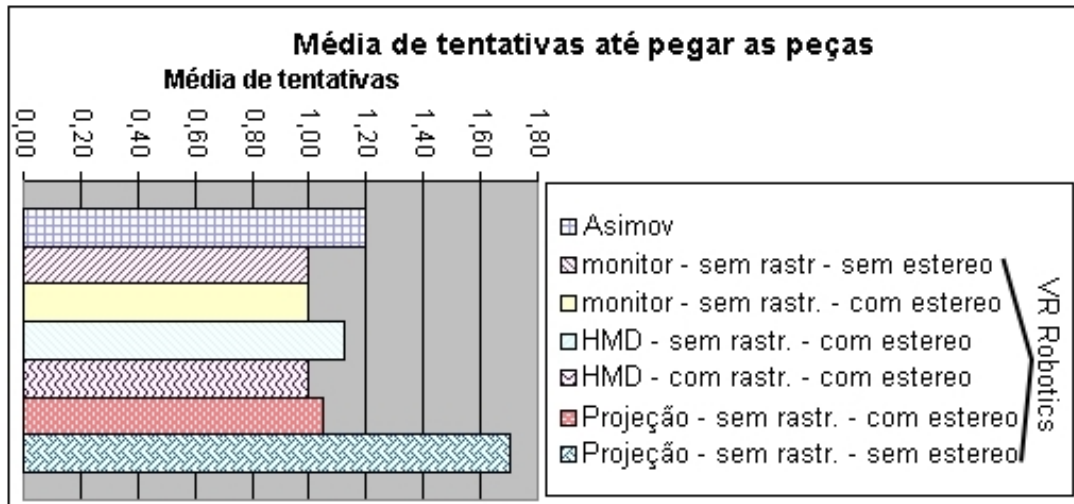


Figura 37: Gráfico da média de tentativas até pegar as peças

8.6.4 Conclusões dos testes

O tempo de realização da tarefa foi reduzido consideravelmente após a mudança na forma de interação do *joystick* para o *mouse*, mas ainda assim alguns usuários comentaram nos questionários que a navegação com o mouse ficou um pouco confusa no início, porém o tempo de aprendizado da navegação com o mouse foi menor.

O monitor com estereoscopia ativada foi o dispositivo que apresentou melhores resultados, obtendo menor tempo para realização da tarefa e também para pegar as peças, necessitando do menor número de tentativas para esta tarefa e obtendo boa precisão na posição final das peças. Acredita-se que este dispositivo é o que melhor se adapta à interação com o mouse, e por isto obteve este resultado. Além disto, o monitor é o dispositivo que exige menor gasto para implantação, exigindo apenas uma placa de vídeo simples e um óculos do tipo *shutter*, sendo que a aquisição de ambos pode ser feita a partir de R\$670, como pode ser visto nas Tabelas 4 e 5.

Acredita-se que a tela de projeção seja uma opção interessante, apesar dos resultados apresentados, pois o grupo de usuários que realizaram o teste com este dispositivo foi pequeno, e apenas um destes usuários conhecia o TP, exigindo um tempo para conhecimento das funções deste dispositivo. Quanto ao custo para implantação, deve ser considerado que a maioria das escolas possui dois projetores iguais, e o custo do restante de equipamentos a serem adquiridos é praticamente igual ao obtido na configuração mais barata.

O HMD foi o dispositivo que apresentou a menor precisão na posição final das peças. Este dispositivo foi considerado desconfortável por praticamente todos usuários, o que prejudicou a realização da tarefa e pode ter sido responsável pelo resultado. O uso de rastreamento com este dispositivo também não ajudou no processo interativo, o que indica que não é necessário rastreamento quando for utilizado o *mouse* como dispositivo de interação.

Tabela 4: Lista de equipamentos para todas configurações do VR Robotics

Cód	Equipamento	Preço	Loja
PV	Placa de vídeo G-Force FX5500 (com 2 saídas VGA)	R\$170	On Line BR http://www.onlinebr.com.br
SGN	Shutter glasses Nuvision 60GX-NSR	R\$975 (FOB Miami) R\$1.808 (ICMS 12%) R\$1.917 (ICMS 17%)	Absolut Technologies http://www.abs-tech.com
SGS	Shutter glasses Stereo Eyes Wireless	R\$254 (FOB Miami) R\$472 (ICMS 12%) R\$500 (ICMS 17%)	Absolut Technologies
HMD	HMD I-Glasses PC-3D Pro	R\$2.600 (FOB Miami) R\$4.346 (ICMS 12%) R\$4.608 (ICMS 17%)	Absolut Technologies
RAC	Rastreador Cyber Track II	R\$2.000 (ICMS 17%)	Absolut Technologies
RAP	Rastreador Polhemus Isotrack II	R\$10.000 (ICMS 17%)	Absolut Technologies
OP	Óculos polarizado de papel	R\$2,05 un. (ICMS 17%)	Absolut Technologies
PR	Proj. SHARP XR10S 2000 lumens	R\$2.630	On Line BR
TPB	Tela de projeção retrátil 1,50m x 1,50m branca	R\$259	Submarino http://www.submarino.com.br
TPA	Tela de projeção aluminizada (modelo Frame)	R\$700	Projetelas http://www.projetelas.com.br
SUP	Suporte com 2 lentes polarizadoras	R\$270	3D RV

Analisando-se a Tabela 5 é possível notar que quando se utilizam vários computadores as configurações com HMD se tornam praticamente inviáveis, pois o custo sem rastreamento equivale a 207% do custo do RoboCell, chegando a 293% quando for utilizado rastreamento.

As configurações com tela de projeção mantém praticamente o mesmo valor, pois foi considerado o uso de apenas uma tela, aumentando-se apenas os equipamentos necessários para possibilitar um número maior de expectadores, dessa forma, para vinte alunos temos um custo de implantação equivalente a 14% do custo para o RoboCell.

Nota-se ainda que a configuração do monitor com estereoscopia, que foi a que obteve melhores resultados, possui um custo de implantação equivalente a 30% do custo para o RoboCell, sendo considerada a melhor alternativa.

Tabela 5: Estimativa de custos para implantação do VR Robotics nas escolas

Configuração	Equipamento	Custo	Custo 1 PC	Custo 20 PCs
Monitor	1 PV	R\$170	R\$170	R\$3.400
Monitor com estereoscopia	1 PV	R\$170	de R\$670 à R\$2.087	de R\$13.400 à R\$41.740
	1 SGN ou SGS	entre R\$500 (SGS) e R\$1.917 (SGN)		
HMD com estereoscopia	1 PV	R\$170	R\$4.778	R\$95.560
	1 HMD	R\$4.608		
HMD com estereoscopia e rastreamento	1 PV	R\$170	de R\$6.778 à R\$14.778	de R\$135.560 à R\$295.560
	1 HMD	R\$4.608		
	1 RAC ou RAP	entre R\$2.000 (RAC) e R\$10.000 (RAP)		
Projeção	1 PV	R\$170	R\$3.059	R\$3.059
	1 PR	R\$2.630		
	1 TPB	R\$259		
Projeção com estereoscopia	1 PV	R\$170	R\$6.402,05	R\$6.441 (com 20 OP)
	2 PR	R\$5.260		
	1 TPA	R\$700		
	1 SUP	R\$270		
	1 OP	R\$2,05		
RoboCell (p/ comparar)	1 PV	R\$170	R\$2.310	R\$46.200
	Licença	R\$2.140		

9 Conclusões

Após a realização deste trabalho, constatou-se que a melhor opção para ensinar robótica nas escolas, utilizando recursos de RV, é através do uso de um monitor com estereoscopia, pois esta opção apresentou melhora na realização da tarefa, aumentando a precisão e diminuindo o tempo total.

Em escolas que possuem turmas com muitos alunos, pode ser interessante utilizar a opção de projeção com estereoscopia, pois apenas um equipamento pode ser utilizado por todo grupo de alunos.

Constatou-se que não é necessário utilizar rastreamento com o VR Robotics, devido à forma de interação adotada, que utiliza principalmente o mouse como dispositivo de interação. Com isto conclui-se também que o HMD não oferece ganho na execução da tarefa, sendo mais apropriado para outros tipos de ambientes virtuais.

Para que sejam obtidos resultados mais concretos, seria necessária a realização de novos testes, com um grupo maior de usuários, e principalmente, utilizando mais usuários ligados à robótica, pois notou-se que o desempenho de alguns usuários não foi melhor pelo desconhecimento do uso de um TP para comandar um robô, inclusive alguns usuários sugeriram nos questionários que antes da realização do teste fosse apresentado um manual explicando as funções do TP.

Além disto, notou-se que será necessário realizar um trabalho de conscientização de professores de robótica e diretores nas escolas, para a adoção de um simulador com RV, para que não aconteça como ocorreu com o Asimov, que acabou não sendo utilizado nas escolas por resistência dos próprios professores. Sabe-se que a adoção do Asimov não ocorreu também por conta de problemas do simulador, mas algumas escolas do SENAI sequer chegaram a conhecer este aplicativo.

Como trabalhos futuros, sugere-se a implementação de um ambiente para criação e compilação de programas em linguagem ACL, para que o simulador VR Robotics se torne completo. Também seria interessante a implementação da comunicação com o robô real, permitindo o envio de programas e até mesmo uma simulação sincronizada com a execução no robô real.

Conforme os resultados apresentados acredita-se objetivo de desenvolver uma aplicação que melhore a tarefa de obter os pontos de referência foi alcançado.

Referências

- [1] Artoolkit. Site: <http://www.hitl.washington.edu/artoolkit/>. acessado em 18/07/2006.
- [2] Denford. Site: <http://www.denford.co.uk/product.asp?link=119>. acessado em 15/09/2005.
- [3] Fanuc. Site: <http://www.fanucrobotics.com>. acessado em 15/09/2005.
- [4] Intelitek. Site: <http://www.intelitek.com>. acessado em 15/09/2005.
- [5] Laboratório CIM - Computer Integrated Manufacturing. Site: <http://www.em.pucrs.br/labcim>. acessado em 15/06/2006.
- [6] Vraniml. Site: <http://www.greathill.com/>. acessado em 16/11/2006.
- [7] Acl - linguagem de controle avançada - guia de referência e ats - terminal avançado de software. Eshed Robotec Limited, 1992.
- [8] D. D. Ardayfio. *Fundamentals of Robotics*, volume 57 of *Mechanical Engineering*. CRC Press, Marcel Dekker edition, January 1987.
- [9] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997.
- [10] J. T. Bell and H. S. Fogler. Vicher: A virtual reality based educational module for chemical reaction engineering. *Computer Applications in Engineering Education*, 4(4):285–296, 1996.
- [11] R. Bukowski and C. Séquin. Interactive simulation of fire in virtual building environments. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 35–44, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [12] A. A. A. Coelho, H. A. O. e Silva, F. Schirmbeck, M. Cohen, F. H. Cardoso, M. S. Pinho, R. M. da Silva, M. S. dos Santos, and A. Mazzorani. Programação e simulação de robôs usando técnicas de realidade virtual. WRV'99 - II Workshop de Realidade Virtual, 1999. Marília, SP.

- [13] J. K. Crosier, S. V. G. Cobb, and J. R. Wilson. Experimental comparison of virtual reality with traditional teaching methods for teaching radioactivity. *Education and Information Technologies*, 5(4):329–343, 2000.
- [14] M. Mujtaba et al. Stanford's al robot programming language. *ASME Computers in Mechanical Engineering*, 2:50–57, 1982.
- [15] J. Geissler, P. J. Knott, M. R. Vazquez, and J. R. Wright Jr. Virtual reality robotic programming software in the technology classroom. *The technology teacher*, 63:6–12, March 2004.
- [16] VRML Architecture Group. The virtual reality modeling language specification version 2.0. Site: <http://vag.vrml.org>. acessado em 17/10/2006.
- [17] T. C. Hudson, M. C. Lin, J. Cohen, S. Gottschalk, and D. Manocha. V-collide: accelerated collision detection for vrml. In *VRML '97: Proceedings of the second symposium on Virtual reality modeling language*, pages 117–ff., New York, NY, USA, 1997. ACM Press.
- [18] Intelitek Inc. *RoboCell User Manual*, November 2003. Catalog #100346 Rev. D.
- [19] Intelitek Inc. *Scorbase User Manual*, November 2003. Catalog #100342 Rev. E.
- [20] T. U. St. Julien and C. D. Shaw. Firefighter command training virtual environment. In *TAPIA '03: Proceedings of the 2003 conference on Diversity in computing*, pages 30–33, New York, NY, USA, 2003. ACM Press.
- [21] H. Kaufmann. Construct3d: an augmented reality application for mathematics and geometry education. In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 656–657, New York, NY, USA, 2002. ACM Press.
- [22] G. D. Kessler, D. A. Bowman, and L. F. Hodges. The simple virtual environment library: An extensible framework for building ve applications. *Presence: Teleoperators & Virtual Environments*, 9(2):187–208, 2000.
- [23] L. I. Lieberman and M. A. Wesley. Autopass: An automatic programming system for computer controlled mechanical assembly. *IBM J. Res. Develop.*, 21:321–333, 1977.
- [24] B. F. Lu, K. T. Lim, J. M. Zheng, and Y. Y. Cai. Learning molecular biology by vr playing. In *VRCAI '04: Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, pages 168–172, New York, NY, USA, 2004. ACM Press.
- [25] J. W. Masterson, E. C. Poe, and S. W. Fardo. *Robotics*. Reston Publishing Company, Inc, 1985.

- [26] T. L. Perez and P. Winston. Lama: A language for automatic mechanical assembly. In *Proceedings of 5th IJCAI*, pages 710–716, 1977.
- [27] M. S. Pinho. Dispositivos para realidade virtual. Site: <http://grv.inf.pucrs.br/Pagina/TutRV/disp.htm>. acessado em 20/08/2006.
- [28] II R. L. Williams, M. Srivastava, J. N. Howell, Jr. R. R. Conatser, D. C. Eland, J. M. Burns, and A. G. Chila. The virtual haptic back for palpatory training. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*, pages 191–197, New York, NY, USA, 2004. ACM Press.
- [29] M. Rohrmeier. Web based robot simulation using vrm. In *WSC'00: Proceedings of the 2000 Winter Simulation Conference (WSC'00) - Volume 2*, pages 1525–1528, Washington, DC, USA, 2000. IEEE Computer Society.
- [30] R. Safaric, M. Debevc, R. Parkin, and S. Uran. Telerobotics experiments via internet. *Education and Information Technologies*, 48(2):424–431, April 2001.
- [31] A. Sánchez, J. M. Barreiro, and V. Maojo. Design of virtual reality systems for education: A cognitive approach. *Education and Information Technologies*, 5(4):345–362, 2000.
- [32] C. Wagner, M. Schill, and R. Männer. Intraocular surgery on a virtual eye. *Commun. ACM*, 45(7):45–49, 2002.
- [33] A. Yeh. Vrmath: knowledge construction of 3d geometry in virtual reality microworlds. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 1061–1062, New York, NY, USA, 2004. ACM Press.

Apêndice A - Aplicações adicionais do VR Robotics

Além do VR Robotics, foram desenvolvidas algumas aplicações auxiliares, para permitir a configuração do programa principal e também o uso de dispositivos remotamente, como visto nas seções a seguir.

A.1 Config

Para que o VR Robotics funcione corretamente, devem ser informadas algumas configurações, para identificação do tipo de equipamentos utilizados. Para facilitar essa configuração, foi criado o programa “config.exe”, visto na Figura 38, que permite escolher os parâmetros de execução.

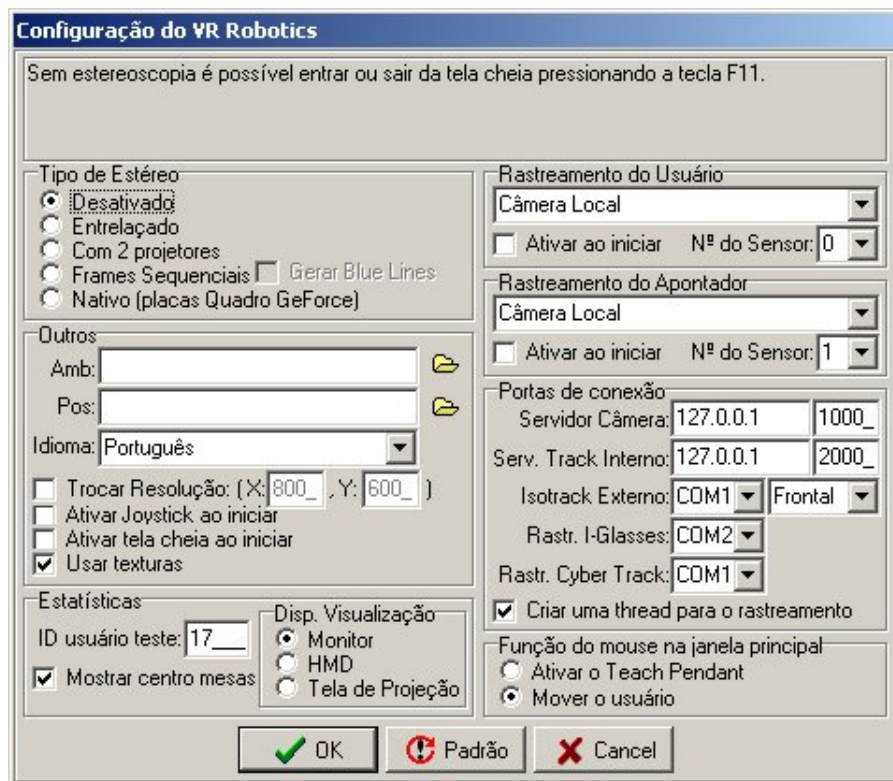


Figura 38: Config - Programa de configuração do VR Robotics

Na parte superior do programa é exibido um texto de ajuda, de acordo com o ítem selecionado na tela, a fim de esclarecer ao usuário a função deste ítem.

O VR Robotics e o config possuem dois idiomas, inglês e português, podendo ser escolhido neste aplicativo.

Neste programa pode ser escolhido o tipo de estereoscopia a ser empregado, que depende dos dispositivos de visualização e da placa de vídeo utilizados. Também podem ser escolhidos os tipos de rastreador para a cabeça e a mão do usuário. Para isto deve ser indicado o tipo de dispositivo, o número do sensor a ser utilizado e a porta de conexão, além de informar se o rastreamento será inicializado junto com o programa principal.

Alguns dispositivos possuem uma resolução de tela específica para funcionar corretamente. É possível informar a resolução a ser utilizada, e se o aplicativo deve trocar a resolução ao iniciar, além de indicar se a tela cheia deve ser ativada e se o joystick será ativado ao iniciar.

Além disto, um ambiente padrão e um arquivo de posições de memória podem ser informados, dispensando a exibição do diálogo de Abrir Ambiente ao iniciar o programa.

No VR Robotics o mouse possui duas funções, podendo a função desejada ser escolhida aqui.

Quando forem coletadas estatísticas no VR Robotics, devem ser indicados na configuração o identificador do usuário e o dispositivo utilizado para visualização. Também pode-se escolher se o centro das mesas deve ser realçado, a fim de auxiliar o usuário a soltar uma peça no centro de uma mesa.

A.2 Servidor de Rastreamento por Câmera

O programa “server_camera.exe” é o servidor utilizado para executar o rastreamento por câmera no modo remoto, e deve ser iniciado antes do programa principal. A função deste aplicativo, visto na Figura 39, é fornecer a posição dos padrões para o programa principal, sendo possível também a visualização da imagem captada, fornecendo retorno visual quanto à identificação dos padrões, marcando na imagem os padrões reconhecidos com um retângulo verde e uma espécie de chave para indicar a orientação (padrão “*Hiro*” na Figura 39), enquanto que regiões pretas, que podem ser padrões, mas não foram identificadas, são marcadas com um retângulo vermelho (padrão “A” na Figura 39).

A configuração é feita no arquivo “bin/camera/config_padroes.dat”, onde deve ser indicado quais os padrões a serem reconhecidos. Foi empregado o padrão “*Hiro*” para o rastreamento da cabeça do usuário e o padrão “A” para o rastreamento da mão, que seleciona um botão do TP.

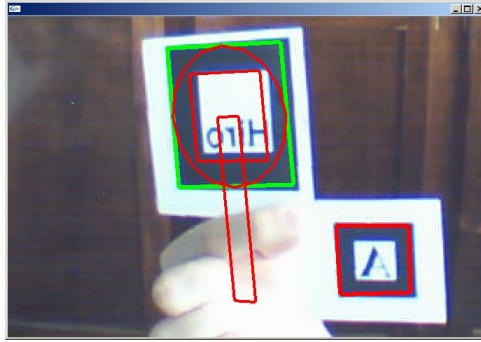


Figura 39: Servidor de Rastreamento por Câmera reconhecendo um padrão e não reconhecendo outro

A.3 Servidor de rastreamento interno

O rastreamento com o Polhemus Isotrack Interno somente pode ser realizado na forma cliente/servidor, pois este equipamento é conectado no computador através do barramento ISA, que não é mais fabricado. Além disso, as máquinas com este barramento, por serem antigas, provavelmente não sejam capazes de rodar o simulador. Então, para utilizar este dispositivo é necessário rodar o programa “server_internal_track.exe” antes do VR Robotics, para fornecer as informações de rastreamento.

Apêndice B - Componente curricular de Robótica - SENAI Gravataí

Aula		Objetivo	Conteúdos	Avaliação	Recursos Didáticos
1	Definição dos robôs	Conhecer as definições envolvendo robótica e suas aplicações	Definição, conceito, comparação de robôs com outras tecnologias. Discussão social Robô x emprego. Aplicações de sistemas robóticos	Reconhecer o grau de interesse da turma em relação à robótica.	Quadro branco
2	Visão de sistemas robóticos	Visualizar as aplicações dos robôs na indústria.	Vídeos e mídia mostrando sistemas robóticos e suas aplicações	*****	Quadro branco TV / vídeo Projetor Computador
3	Juntas	Conhecer os componentes do sistema robótico	Composição do sistema robótico. Malha de controle do sistema robótico (noção de controladores ON-OFF, P, I, D) Juntas (rotacional, esférica e prismática)	Questionamentos teóricos do assunto	Quadro branco TV / vídeo
4	Braço robótico	Saber classificar os robôs quanto ao tipo de juntas	Classificação dos robôs quanto ao tipo de juntas- Punho do robô	Questionamentos teóricos do assunto	Quadro branco
5	Acionamentos e sensoramento	Saber o princípio de funcionamento dos acionamentos e sensoramentos robóticos	Acionamentos (hidráulico, pneumático e elétrico), servomotores e princípio de PWM Sensoramento por encoders e resolver	Questionamentos teóricos do assunto	Quadro branco

6	Órgão Terminal e Parâmetros dos Robôs	Conhecer os tipos de órgão terminais, saber interpretar os parâmetros dos robôs	Função, tipos (garras, ferramentas), trocador de ferramentas, configuração de TCP.Parâmetros importantes (graus de liberdade, precisão, carga útil, área de trabalho, repetibilidade, configuração de juntas)	Pesquisa de parâmetros nos catálogos de robôs	Quadro branco Catálogos dos robôs
7	Pesquisa de Parâmetros dos Robôs	Ser capaz de pesquisar parâmetros de um sistema robótico	Pesquisa de parâmetros do sistema robótico na Internet. Produção de um Folder	Êxito na produção do Folder	Quadro branco Lab. info.
8	Avaliação composição do sistema robótico	Avaliar a compreensão dos conteúdos sobre composição do sistema robótico.	Vídeo do telecurso 2000 como revisão dos conteúdos.Atividade de avaliação teórica	Avaliação teórica	Quadro branco TV / vídeo
9a	Teach Pendant	Compreender a manipulação do sistema robótico	Funções, comandos, sistemas de coordenadas (joint, cartesiana)	Questionamentos teóricos do assunto	Quadro branco Retroprojektor
9b	Teach Pendant (Prática)	Ser capaz de manipular o robô	Prática no robô didático (manipulação do robô)	Êxito na atividade prática	Eshed ER-V Computadores
10a	Marcação de pontos	Conhecer os comandos de marcação de pontos pelo teach pendant no robô	Marcação de pontos, movimentação entre os pontos	Questionamentos teóricos do assunto	Quadro branco
10b	Marcação de pontos (montagem de uma pilha)	Ser capaz de movimentar e marcar pontos pelo teach pendant no robô	Prática no robô didático (montagem de uma pilha)	Êxito na atividade prática	Eshed ER-V Computadores

11a	Programação ON-LINE	Conhecer os comandos ACL, saber as funções	Ambiente ACL, Modo direto, comandos do modo de edição, instruções print, println, move, moved, run, delay, edição	Questionamentos teóricos do assunto	Quadro branco
11b	Programação ON-LINE	Ser capaz de editar um programa na linguagem ACL	Prática no robô didático (Apresentação - fila indiana)	Êxito na atividade prática	Eshed ER-V Computadores
12a	Programação OFF-LINE	Conhecer a forma de programação OFF-LINE	Comandos do modo direto. Elaboração de programas no modo direto	Questionamentos teóricos do assunto	Quadro branco
12b	Programação OFF-LINE (Leão - Couve - Cabrito)	Ser capaz de implementar um programa no modo OFF-LINE	Prática no robô didático (Leão - Couve - Cabrito)	Êxito na atividade prática	Eshed ER-V Computadores
13a	Laço de repetição	Saber interpretar um programa utilizando laço de repetição	Laço de repetição infinito, laço de repetição finito, instrução for	Questionamentos teóricos do assunto	Quadro branco
13b	Laço de repetição (paletização)	Ser capaz de implementar um programa com laço de repetição	Prática no robô didático (paletização)	Êxito na atividade prática	Eshed ER-V Computadores
14a	Teste Condicional	Conhecer as instruções destinadas a implementar um teste condicional	Definição de teste condicional, condições lógicas, instruções if, else, andif e orif	Questionamentos teóricos do assunto	Quadro branco
14b	Teste Condicional (programa dominó)	Ser capaz de implementar tal teste	Prática no robô didático (dominó)	Êxito na atividade prática	Eshed ER-V Computadores

15a	Periféricos de I/O	Conhecer o interfaceamento do robô com o meio, saber monitorar uma entrada e acionar uma saída	Estudo de sistemas de interfaceamento de entrada e saída, instrução in e out	Questionamentos teóricos do assunto	Quadro branco
15b	Periféricos de I/O (transporte automático)	Ser capaz de elaborar programas utilizando periféricos	Prática no robô didático (transporte automático)	Êxito na atividade prática	Eshed ER-V Computadores
16a	Periféricos de I/O	Saber implementar programas utilizando I/O	Estudo de sistemas de interfaceamento de entrada e saída, instrução in e out, enc	Questionamentos teóricos do assunto	Quadro branco
16b	Periféricos de I/O (Seleção de Peças)	Ser capaz de elaborar programas utilizando periféricos	Prática no robô didático (seleção de peças)	Êxito na atividade prática	Eshed ER-V Computadores
17	Avaliação	Avaliar a compreensão dos conteúdos sobre robótica	Atividade de avaliação teórica individual	Avaliação teórica	Quadro branco
18	Oportunidade de Recuperação R1	Oportunizar a recuperação de competências ainda não alcançadas	Atividade de recuperação teórica/prática Aos alunos que atingiram todas as competências, atividades práticas no robô didático	Avaliação teórica / prática	Eshed ER-V Computadores
19	Oportunidade de Recuperação R2	Oportunizar a recuperação de competências ainda não alcançadas para unidades de estudos	Atividade de recuperação teórica/prática Aos alunos que atingiram todas as competências, atividades práticas no robô didático	Avaliação teórica / prática	Eshed ER-V Computadores

Apêndice C - Questionário Pré-Teste

Avaliação do VR Robotics

Questionário Pré-Teste

Data: __ / __ / ____

Idade: _____ anos

Sexo: () Masculino () Feminino

Número do teste: _____

1. Em média, com que frequência você utiliza o computador:

- diariamente
- mais de 3 vezes por semana
- de 1 a 3 vezes por semana
- 1 vez a cada 2 semanas
- 1 vez por mês ou menos

2. Quais sistemas você costuma utilizar no computador:

- Email (e.g. Outlook express)
- Editor de texto (e.g. Word)
- Planilha (e.g. Excel)
- Sistema de apresentação (e.g. Powerpoint)
- Navegador (e.g. Internet Explorer)
- Sistemas de bate-papo
- Sistemas de discussão
- Jogos
- Outros: _____

3. Em geral como você se considera um usuário de computadores:

- Avançado () Intermediário () Principiante

4. Como você avalia a sua disposição em utilizar o computador:

- Gosto muito, e utilizo sempre que posso
- Gosto, mas não tenho muita chance de utilizar
- Não me incomoda utilizar o computador e o faço sempre que necessário
- Não gosto de utilizar e evito sempre que posso

Não gosto e não utilizo nunca o computador

5. Você é familiarizado com o uso de Joystick de videogame?

Sim Não

6. Você é familiarizado com o uso de um Teach Pendant para manipular robôs?

Sim Não

7. Você possui algum problema visual?

Não

Astigmatismo

Miopia

Hipermetropia

Presbicia / Presbiopia

Outro _____

Apêndice D - Questionário Pós-Teste

Avaliação do VR Robotics

Formulário de Acompanhamento e Pós-Teste

Data: __ / __ / ____

Número do teste: _____

1) Como você avalia a forma de interação com o ambiente?

fácil médio difícil normal

Comentários: _____

2) Em comparação com o Asimov, a união de todos os comandos de navegação e manipulação do robô no Joystick tornou a interação fácil e/ou intuitiva?

sim não

Porque? _____

3) Na sua opinião, em comparação com o Asimov, a detecção de colisão ajudou a melhorar a percepção do ambiente?

sim não

Porque? _____

4) Você acha que em relação à versão sem RV, os recursos de RV ajudaram na realização da tarefa?

sim não

Porque? _____

5) Você se sentiu confortável utilizando os equipamentos de RV?

() sim () não

Porque? _____

6) Escreva aqui as observações, críticas, sugestões, etc.

Apêndice E - Documento de descrição dos testes

Testes de Avaliação do VR Robotics Instruções para o Experimento

Introdução

Seja bem vindo! Muito obrigado por concordar em participar de nosso experimento. Neste trabalho estamos explorando o potencial dos Ambientes Virtuais para a manipulação de robôs.

Você irá interagir em dois ambientes tridimensionais onde poderá controlar um robô Scorbot ER VII e fazê-lo "pegar" e mover peças no ambiente.

Através deste experimento você estará nos ajudando a entender e a avaliar o potencial e a facilidade de uso de diferentes tipos de dispositivos de realidade virtual. Durante o experimento você usará um óculos de realidade virtual, através do qual verá o ambiente virtual.

Solicitaremos a você que realize certas tarefas utilizando o controlador do robô, semelhante a um controle remoto, com o qual você poderá manipular objetos no ambiente virtual.

Ao final do experimento solicitaremos que você que responda a algumas questões sobre sua impressão a respeito de sua experiência no ambiente virtual.

Gostaríamos ressaltar que, neste experimento, **não estaremos avaliando você ou seu desempenho**. Estamos interessados, isto sim, em avaliar as técnicas de interação e a qualidade de nosso ambiente.

O Experimento

O experimento em que você irá participar consiste de uma prática inicial onde você poderá familiarizar-se com os equipamentos, com o ambiente virtual e com as técnicas de interação a serem utilizadas. A partir disto você irá iniciar a execução de algumas tarefas.

O experimento deverá ser realizado duas vezes, sendo uma delas no software VR Robotics, e a outra em um programa similar, chamado **Asimov**.

Primeiramente, você terá algum tempo para familiarizar-se com os programas e a seguir a tarefa será iniciada.

A execução destas tarefas deve levar entre 35 e 40 minutos. Procure planejar a execução das tarefas previamente e se, for o caso discuti-las com o pesquisador que o acompanhará durante os testes.

Caso você se sinta enjoado, tonto ou com algum desconforto por causa do uso dos óculos, pode retirá-lo e suspender a execução da tarefa a qualquer momento.

Ao final, solicitaremos que você preencha um questionário com suas impressões sobre a execução das tarefas.

O ambiente para realização dos testes, exibido na Figura 1, possui do lado esquerdo duas mesas giratórias, esteira e a mesa de experiência. Sobre as mesas encontram-se cinco peças que deverão ser movidas pelo usuário.

O robô está no centro do ambiente virtual, sobre uma esteira, que deverá ser utilizada para posicioná-lo ao alcance de todas peças contidas no cenário. Esta esteira está associada ao eixo 6 do robô.

Ainda existem outras cinco mesas giratórias, posicionadas no lado direito do ambiente.



Figura 1: Ambiente de testes

A tarefa a ser executada consiste em pegar cada uma das cinco peças do lado esquerdo, e soltá-las sobre a mesa do lado direito que estiver mais alinhada à posição da qual a peça foi pega.

Para manipular o robô durante a execução desta tarefa, o usuário deverá utilizar o Teach Pendant.

Para que a tarefa seja realizada com a maior precisão possível, aconselha-se que sejam seguidas algumas regras:

- as peças devem ser soltas o mais próximo possível do centro das mesas, realçados com uma marca preta;
- a garra do robô deve ser fechada somente quando se desejar pegar uma peça;
- a garra do robô deve ser aberta somente quando a tentativa de pegar uma peça falhou, ou para soltar uma peça sobre uma mesa do lado direito do cenário;
- uma vez que uma peça foi solta, a mesma não deve ser pega novamente, mesmo que o posicionamento final não tenha sido satisfatório;
- a tarefa deve ser realizada o mais rapidamente possível, porém mantendo a qualidade ao pegar ou soltar as peças.

Após a realização desta tarefa você responderá um questionário, visando a avaliação do ambiente desenvolvido.

Mais uma vez, muito obrigado por participar!

Bom teste!

Apêndice F - Estatísticas dos testes dos usuários

Tabela 7: Estatísticas do VR Robotics - Monitor - Sem rastreamento - Sem Estereoscopia

Info	ID User	Peças não pegou	Médias			Tempo (s)			Tentat. para pegar			Pos Soltou			Tentativas					
			Dist. erro	Temp pegar	Tent. Pegar	Total	Pegar	Soltar	ID Peça	X	Y	Z	N	X	Y	Z	Tempo (s)	Dist. garra	Dist. user	Dist. Erro
13/10/2006 15:19:32 testes_mau ro5.amb	7	0	0,00	25,85	1,00	252,64	25,28	22,14	1	51,13	18,63	7,97	1	-29,45	19,63	8,55	25,28	0,00	28,72	5,31
13/10/2006 15:24:57 testes_mau ro5.amb	9	0	0,00	25,30	1,00	199,59	43,17	22,54	2	45,57	21,04	-22,37	1	-31,07	9,71	-32,59	43,17	0,00	56,76	5,3
13/10/2006 15:53:53 testes_mau ro5.amb	6	0	0,00	66,33	1,00	516,39	18,79	16,34	3	37,16	20,53	-48,26	1	-40,49	19,30	-47,94	18,79	0,00	47,71	9,68
							24,99	25,43	4	47,98	15,75	-89,20	1	-51,42	17,56	-92,88	24,99	0,00	39,92	5,44
							17,04	36,53	5	46,22	28,97	-110,93	1	-22,39	17,00	-110,63	17,04	0,00	59,78	9,1
							17,14	13,35	1	55,38	17,86	4,38	1	-28,29	19,68	11,51	17,14	0,00	83,76	2,75
							36,12	10,24	2	45,28	18,07	-26,64	1	-29,47	9,08	-33,40	36,12	0,00	58,15	2,31
							21,14	6,85	3	37,44	18,00	-57,26	1	-41,65	19,17	-49,66	21,14	0,00	25,89	4,45
							33,60	23,54	4	50,13	17,64	-82,48	1	-50,26	16,89	-97,43	33,60	0,00	29,49	8,44
							18,49	17,99	5	49,18	27,30	-121,84	1	-23,25	17,01	-112,25	18,49	0,00	41,39	6,17
							41,37	26,64	1	56,41	28,53	6,05	1	-28,31	18,31	12,32	41,37	0,00	47,66	4,57
							46,18	25,43	2	45,48	18,76	-25,09	1	-33,02	9,27	-28,42	46,18	0,00	27,58	3,13
							77,29	16,94	3	30,75	29,13	-58,86	1	-42,26	20,41	-50,03	77,29	0,00	36,84	2,92
							74,15	65,27	4	47,39	18,59	-86,07	1	-54,02	16,53	-96,05	74,15	0,00	48,00	4,49
							92,66	49,37	5	49,01	18,47	-117,59	1	-19,64	17,55	-114,28	92,66	0,00	32,80	4,03
Média		0,00	0,00	39,16	1,00	322,87	39,16	25,24		1,00							39,16	0,00	44,30	5,21
Desvio Padrão		0,00	0,00	23,53	0,00	169,68	24,18	15,21		0,00							24,18	0,00	15,76	2,29

Tabela 8: Estatísticas do VR Robotics - Monitor - Sem rastreamento - Com Estereoscopia
 Estatísticas do VR Robotics - Monitor - Sem rastreamento - Com Estereoscopia

Info	ID User	Peças não pegou	Médias			Tempo (s)			ID Peça	Tentat. para pegar			Pos Solto						Tentativas					
			Dist. erro	Temp pegar	Tent. Pegar	Total	Pegar	Soltar		X	Y	Z	N	X	Y	Z	Tempo (s)	Dist. garra	Dist. user	Dist. Erro				
27/10/2006 15:25:35 testes_mau ro5.amb	11	0	0,00	50,63	1,00	404,09	31,95	29,88	1	53,06	21,83	8,96	1	-29,63	19,89	11,12	31,95	0,00	24,10	5,96				
							42,63	23,81	2	48,88	18,47	-28,68	1	32,88	9,43	-29,95	42,63	0,00	45,15	6,12				
							57,16	30,36	3	35,16	22,44	-51,52	1	-43,22	19,15	-48,62	57,16	0,00	28,77	5,89				
							50,81	26,75	4	45,54	17,72	-83,00	1	-53,35	16,63	-93,00	50,81	0,00	37,78	4,74				
							70,62	39,53	5	57,24	19,54	-119,33	1	-21,13	16,92	-112,32	70,62	0,00	22,33	4,84				
27/10/2006 15:44:39 testes_mau ro5.amb	12	0	0,00	77,15	1,00	540,33	90,59	21,08	1	48,29	17,19	2,94	1	-30,59	20,24	7,58	90,59	0,00	33,20	4,71				
							94,66	28,55	2	40,64	16,73	-28,40	1	-35,53	9,81	-32,38	94,66	0,00	66,01	2,74				
							101,63	33,35	3	32,45	17,58	-56,17	1	-43,76	20,34	-50,48	101,63	0,00	64,90	1				
							69,54	36,50	4	49,45	18,67	-87,61	1	-54,87	16,68	-93,25	69,54	0,00	38,16	6,48				
							29,33	34,69	5	44,40	21,12	-109,55	1	-21,73	16,68	-116,19	29,33	0,00	64,61	11,38				
27/10/2006 16:25:00 testes_mau ro5.amb	13	0	0,00	24,83	1,00	194,68	22,92	10,40	1	47,94	17,64	3,21	1	-28,29	19,68	9,91	22,92	0,00	29,60	5,06				
							50,11	12,59	2	43,79	18,07	-28,43	1	-34,13	8,67	-33,81	50,11	0,00	42,13	1,63				
							18,42	15,04	3	36,61	18,23	-52,80	1	-41,05	19,20	-50,38	18,42	0,00	37,77	5,54				
							14,36	16,38	4	49,66	19,14	-85,66	1	-53,30	16,99	-92,91	14,36	0,00	34,99	6,79				
							18,34	15,87	5	56,96	18,53	-108,42	1	-23,72	17,29	-113,15	18,34	0,00	84,55	9,45				
27/10/2006 16:50:21 testes_mau ro5.amb	14	0	0,00	23,57	1,00	199,11	26,37	18,74	1	59,16	17,60	-4,39	1	-30,13	19,23	12,36	26,37	0,00	34,24	9,62				
							25,83	6,93	2	37,60	17,82	-23,93	1	-33,59	9,50	-30,52	25,83	0,00	45,94	6,21				
							33,41	10,51	3	47,08	18,00	-59,24	1	-41,61	19,51	-50,81	33,41	0,00	26,79	14,26				
							14,46	14,44	4	43,43	17,58	-91,41	1	-49,87	16,46	-97,17	14,46	0,00	33,60	4,43				
							17,76	29,96	5	41,61	19,53	-111,56	1	-19,53	16,17	-114,04	17,76	0,00	45,86	12,62				
27/10/2006 20:14:12 testes_mau ro5.amb	17	0	0,00	43,89	1,00	363,62	80,39	28,97	1	54,29	20,39	-0,43	1	-32,33	20,12	10,34	80,39	0,00	51,08	3,66				
							55,81	20,17	2	43,67	17,49	-29,89	1	-32,14	9,08	-33,62	55,81	0,00	59,61	2,97				
							12,21	38,32	3	39,81	18,21	-56,11	1	-42,30	19,51	-48,16	12,21	0,00	91,72	6,87				
							34,05	31,49	4	44,70	18,11	-91,49	1	-55,38	16,72	-92,45	34,05	0,00	45,91	4,8				
							36,97	24,71	5	53,73	19,02	-115,67	1	-21,61	15,51	-113,83	36,97	0,00	31,96	1,52				

27/10/2006 20:19:00 testes_mau ro5.amb	15	0	0,00	22,06	1,00	185,99	26,99	17,48	1	1	58,68	18,67	-3,27	1	-31,46	19,36	9,56	26,99	0,00	32,32	8,46
							13,15	10,01	2	1	44,99	17,46	-24,43	1	-33,23	9,08	-31,19	13,15	0,00	51,17	3,25
							21,93	12,68	3	1	32,12	18,79	-55,96	1	-44,36	19,40	-46,81	21,93	0,00	49,25	1,36
							36,49	13,27	4	1	45,62	16,85	-84,51	1	-53,27	17,28	-92,19	36,49	0,00	42,01	3,61
							11,73	22,01	5	1	56,15	18,50	-116,90	1	-23,36	16,72	-109,89	11,73	0,00	47,83	3,15
27/10/2006 20:24:19 testes_mau ro5.amb	16	0	0,00	28,08	1,00	228,38	19,63	13,41	1	1	48,77	18,65	5,81	1	-32,30	19,34	8,32	19,63	0,00	36,70	5,08
							38,07	14,37	2	1	41,30	18,07	-26,68	1	-31,73	9,09	-33,76	38,07	0,00	49,18	1,73
							20,56	25,91	3	1	26,41	20,36	-55,37	1	-42,04	20,31	-47,44	20,56	0,00	43,96	6,79
							36,87	21,15	4	1	38,84	24,88	-86,12	1	-52,48	16,42	-93,49	36,87	0,00	60,15	4,25
							25,25	12,71	5	1	48,21	18,51	-118,40	1	-20,34	15,39	-114,31	25,25	0,00	27,42	4,99
Média		0,00	0,00	38,60	1,00	302,31	38,60	21,77	1,00							38,60	0,00	44,59	5,48		
Desvio Padrão		0,00	0,00	20,24	0,00	136,63	24,85	9,10	0,00								24,85	0,00	15,97	3,12	

Tabela 9: Estatísticas do VR Robotics - HMD - Com estereoscopia - Com rastreamento

Info	ID User	Peças não pegou	Médias			Tempo (s)		Tentat. para pegar	Pos. Soltou			Tentativas									
			Dist. erro	Tmp pegar	Tent. Pegar	Total	Pegar		Soltar	ID Peça	X	Y	Z	N	X	Y	Z	Tempo (s)	Dist. garra	Dist. user	Dist. Erro
27/10/2006 16:14:08 testes_mau ro5.amb	13	0	0,00	24,82	1,00	219,72	27,89	13,10	1	53,23	18,30	1,73	1	-28,29	19,68	10,31	27,89	0,00	37,89	1,29	
27/10/2006 16:55:39 testes_mau ro5.amb	14	0	0,00	25,68	1,00	189,11	37,44	14,89	2	41,46	17,36	-28,40	1	-32,44	9,65	-28,76	37,44	0,00	53,91	2,08	
24/11/2006 19:32:27 testes_mau ro5.amb	22	0	0,00	101,34	1,00	806,48	27,56	27,00	3	41,35	18,03	-57,83	1	-40,95	19,51	-50,19	27,56	0,00	34,74	8,39	
							18,67	22,95	4	46,73	20,78	-84,40	1	-52,96	17,42	-96,16	18,67	0,00	51,81	4,55	
							12,52	17,25	5	48,64	32,28	-116,81	1	-20,65	16,28	-116,56	12,52	0,00	84,85	4,36	
							25,14	14,23	1	46,55	18,61	9,39	1	-28,29	19,68	9,51	25,14	0,00	32,46	9,08	
							14,73	12,49	2	37,60	17,82	-23,93	1	-32,66	8,86	-31,36	14,73	0,00	40,04	6,21	
							45,62	13,30	3	40,84	18,38	-53,16	1	-43,06	19,03	-47,10	45,62	0,00	32,96	8,73	
							15,32	10,23	4	55,24	17,52	-89,75	1	-49,40	16,60	-96,85	15,32	0,00	44,94	12,55	
							27,61	10,03	5	50,11	19,21	-107,13	1	-18,57	16,48	-111,88	27,61	0,00	33,99	10,28	
							164,37	100,30	1	62,99	19,29	2,89	1	-31,59	18,69	7,27	164,37	0,00	27,92	9,99	
							59,71	27,66	2	44,76	17,13	-20,93	1	-34,27	9,09	-28,44	59,71	0,00	26,99	6,32	
							84,52	66,57	3	36,24	18,00	-59,84	1	-44,56	19,57	-45,03	84,52	0,00	27,04	4,31	
							161,37	33,91	4	45,51	18,01	-81,68	1	-54,06	16,29	-95,53	161,37	0,00	45,97	5,88	
							36,73	70,99	5	55,51	19,68	-115,66	1	-22,19	16,07	-108,35	36,73	0,00	32,68	2,85	
Média		0,00	0,00	50,61	1,00	405,11	50,61	30,33					1,00				50,61	0,00	40,54	6,46	
Desvio Padrão		0,00	0,00	43,93	0,00	347,94	49,33	27,17					0,00				49,33	0,00	14,91	3,29	

Tabela 10: Estatísticas do VR Robotics - HMD - Com estereoscopia - Sem rastreamento

Info	ID User	Peças não pegou	Médias			Tempo (s)			Tentat. para pegar	Pos Solto			Tentativas								
			Dist. erro	Tnp pegar	Tent. Pegar	Total	Pegar	Soltar		ID Peça	X	Y	Z	N	X	Y	Z	Tempo (s)	Dist. garra	Dist. user	Dist. Erro
27/10/2006 15:07:52 testes_mau ro5.amb	11	0	0,00	63,11	1,00	511,58	62,40	31,70	1	51,78	33,25	4,90	-28,31	19,65	12,33	62,40	0,00	41,29	2,26		
							43,54	36,83	2	43,25	19,84	-26,95	-33,24	9,29	-32,70	43,54	0,00	32,56	0,25		
							58,73	30,76	3	38,44	23,16	-58,59	-42,51	18,75	-48,69	58,73	0,00	24,87	5,67		
							85,61	44,94	4	48,75	18,35	-89,25	-52,72	16,97	-96,71	85,61	0,00	17,18	6,17		
							65,25	49,68	5	57,90	28,59	-131,09	-21,29	16,22	-112,01	65,25	0,00	26,26	14,92		
27/10/2006 16:00:26 testes_mau ro5.amb	12	0	0,00	69,43	1,00	513,73	47,46	18,09	1	41,01	20,51	-3,44	-26,89	19,39	7,84	47,46	0,00	57,96	13,61		
							35,87	1,28	2	36,48	17,30	-22,58	-30,36	9,07	-30,23	35,87	0,00	68,08	7,88		
							92,77	47,61	3	34,89	16,41	-62,08	-41,64	20,06	-49,14	92,77	0,00	30,26	5,42		
							39,98	23,53	4	42,87	18,54	-80,79	-51,62	17,46	-94,70	39,98	0,00	42,93	6,21		
							131,09	60,77	5	52,40	21,48	-117,88	-22,11	15,71	-114,27	131,09	0,00	42,75	1,07		
27/10/2006 19:04:36 testes_mau ro5.amb	15	0	11,46	61,14	1,20	478,47	63,38	47,28	1	57,49	22,86	10,56	-28,11	19,07	14,54	63,38	0,00	58,40	8,79		
							61,44	32,95	2	40,74	15,97	-27,95	-33,05	10,01	-35,16	61,44	0,00	42,20	2,45		
							81,78	37,07	3	40,32	37,85	-64,09	-41,82	19,51	-44,76	81,78	0,00	43,27	10,19		
							61,00	31,93	4	39,71	17,64	-83,04	-61,72	16,48	-89,11	23,09	11,46	66,97	5,15		
							38,10	22,12	5	55,40	18,27	-118,73	-24,80	15,70	-111,38	38,10	0,00	58,39	2,96		
27/10/2006 19:17:24 testes_mau ro5.amb	16	0	7,49	43,86	1,40	339,36	66,67	27,68	1	54,46	18,00	-0,60	-30,69	19,26	8,07	66,67	0,00	25,66	3,88		
							28,07	10,20	2	41,05	17,38	-25,54	-30,65	8,69	-31,59	28,07	0,00	47,46	2,44		
							55,80	16,43	3	35,95	19,23	-68,64	-34,62	19,07	-44,06	38,09	8,95	43,87	12,01		
							43,62	32,48	4	43,92	20,89	-86,33	-52,56	16,47	-92,09	43,62	0,00	47,33	1,14		
							25,13	32,16	5	51,22	18,81	-113,66	-19,08	16,05	-114,61	25,13	0,00	35,38	3,78		
27/10/2006 19:33:07 testes_mau ro5.amb	17	0	0,00	53,27	1,00	447,36	75,60	39,15	1	42,93	23,80	10,85	-28,29	19,68	15,11	75,60	0,00	48,27	12,77		
							52,60	31,16	2	44,04	18,06	-24,98	-34,50	8,67	-29,28	52,60	0,00	45,97	2,27		
							47,50	52,58	3	36,11	17,63	-56,80	-43,00	19,40	-46,40	47,50	0,00	47,45	3,12		
							75,90	26,28	4	44,55	18,00	-86,13	-51,51	16,84	-92,44	75,90	0,00	26,28	1,78		
							14,73	27,57	5	50,80	19,69	-112,57	-23,74	16,08	-114,69	14,73	0,00	66,97	4,95		
Média		0,00	3,79	58,16	1,12	458,10	58,16	32,49	1,12						48,67	0,88	43,60	5,65			
Desvio Padrão		0,00	5,38	9,86	0,18	71,74	24,53	13,54	0,44						29,19	2,78	13,01	4,22			

Tabela 11: Estatísticas do VR Robotics - Tela de projeção - Sem rastreamento - Com estereoscopia

Info	ID User	Peças não pegou	Médias			Tempo (s)			ID Peça	Tentat. para pegar	Pos Soltou			Tentativas							
			Dist. erro	Tmp pegar	Tent. Pegar	Total	Pegar	Soltar			Z	Y	X	N	X	Y	Z	Tempo (s)	Dist. garra	Dist. user	Dist. Erro
24/11/2006 16:18:20 testes_ma uro5.amb	18	0	3,74	110,91	1,20	771,79	80,75	25,65	1	1,83	18,65	44,58	-29,09	18,45	9,53	0,00	73,41	8,5			
							171,53	20,93	2	-23,95	18,06	36,55	10,85	-29,86	3,74	88,24	7,13				
							55,79	64,22	3	-59,11	17,36	43,52	19,50	-50,06	0,00	47,49	10,73				
							107,88	30,16	4	-75,53	19,63	45,92	17,50	-95,26	0,00	84,52	11,84				
							138,60	73,90	5	-107,96	17,80	44,61	17,44	-113,92	0,00	49,55	12,33				
24/11/2006 18:05:12 testes_ma uro5.amb	20	0	0,00	46,02	1,00	428,14	38,25	35,95	1	4,08	17,45	53,31	-32,52	19,68	11,10	0,00	19,31	1,12			
							84,40	18,16	2	-23,89	17,28	47,91	9,42	-30,97	0,00	60,28	5,81				
							33,16	21,56	3	-57,45	18,00	33,75	19,85	-50,09	0,00	67,48	0,87				
							21,44	24,91	4	-84,92	19,09	39,38	16,17	-94,24	0,00	45,64	4,18				
							52,86	5,85	5	-110,45	18,00	52,25	16,17	-114,04	0,00	37,82	6,59				
24/11/2006 18:32:18 testes_ma uro5.amb	21	0	0,00	52,31	1,00	543,20	16,65	25,57	1	2,09	18,23	51,85	-29,08	18,89	11,41	0,00	53,29	1,47			
							53,08	24,04	2	-26,30	17,80	45,20	8,69	-27,26	0,00	44,97	2,31				
							52,76	30,73	3	-54,61	17,66	32,13	18,82	-47,70	0,00	30,60	2,54				
							52,35	63,34	4	-87,44	20,08	41,15	16,91	-97,25	0,00	42,03	1,9				
							86,73	137,44	5	-116,13	18,75	50,90	16,66	-114,24	0,00	57,50	2,27				
24/11/2006 17:12:53 testes_ma uro5.amb	19	0	0,00	66,32	1,00	523,27	68,01	52,00	1	5,50	17,95	48,06	-31,46	19,36	9,96	0,00	26,39	5,54			
							50,56	18,17	2	-25,71	18,35	44,61	8,96	-30,17	0,00	33,98	2,06				
							53,48	61,35	3	-55,79	17,22	32,66	19,86	-48,50	0,00	43,63	1,26				
							99,95	36,91	4	-87,07	20,85	39,74	16,29	-96,28	0,00	46,00	3,26				
							59,61	22,60	5	-116,60	23,61	41,49	15,97	-110,47	0,00	42,26	11,52				
Média		0,00	0,93	68,89	1,05	566,60	68,89	39,67	1,05						0,17	52,54	5,16				
Desvio Padrão		0,00	1,87	29,27	0,10	145,71	38,15	29,56	0,22						0,80	19,90	3,95				

24/11/2006								1	52,49	18,95	2,25	1	-28,29	19,68	10,71	64,88	0,00	53,93	0,91
17:54:20								2	40,64	18,34	-24,10	1	-33,30	10,87	-29,08	96,39	4,32	47,71	3,74
testes_ma	20	0	4,32	75,92	1,20	622,69													
uro5.amb																			
24/11/2006																			
18:41:23																			
testes_ma	21	0	0,00	42,16	1,00	397,97													
uro5.amb																			
Média		0,00	5,56	75,69	1,70	622,19		1,70								39,15	6,30	50,51	5,71
Desvio Padrão		0,00	5,48	39,86	1,14	322,53	51,84	23,92								32,15	7,11	14,51	5,56

Tabela 13: Estatísticas do Asimov

Info	ID User	Peças não pegou	Médias		Tempo (s)	Tentat.			Pos Soltou			Tentativas							
			Dist. erro	Temp pegar		Temp Total	ID Peça	para pegar	X	Y	Z	N	X	Y	Z	Tempo (s)	Dist. garra	Dist. user	Dist. Erro
13/10/2006 14:16:47 testes_ma uro5.amb		6	0,00	83,41	686,84	1,00	1	49,56	18,99	-3,68	1	-30,48	20,07	-8,62	65,11	0,00	104,62	4,56	
							2	40,88	19,41	31,01	1	-31,08	9,75	32,43	98,44	0,00	110,90	4,54	
							3	40,15	22,32	54,11	1	-43,14	20,25	50,14	55,63	0,00	62,76	7,71	
							4	49,82	20,56	83,14	1	-51,31	16,13	96,29	125,28	0,00	65,99	7,84	
							5	49,05	22,50	114,12	1	-19,39	16,87	115,04	72,59	0,00	214,75	4,89	
							1	54,44	22,06	-11,49	2	-27,17	17,96	-10,06	63,22	1,26	60,88	3,33	
							2	40,93	21,94	23,32	1	-33,52	7,39	31,85	75,61	0,00	55,38	4,22	
							3	31,47	17,96	52,36	1	-42,30	20,20	51,69	40,86	0,00	84,24	4,89	
							4	49,65	19,82	78,61	2	-52,42	15,87	96,32	94,22	0,00	54,04		
							3	49,76	19,05	110,99	3	49,76	19,05	110,99	3,24	0,00	136,54	10,71	
							5	49,42	26,82	113,23	1	-18,19	15,23	115,60	67,17	0,00	179,45	5,20	
							1	50,72	21,71	-6,50	1	-29,47	18,53	-10,22	32,84	0,00	90,27	3,77	
							2	44,01	17,88	22,20	1	-31,75	9,78	31,97	69,08	0,00	117,62	4,91	
							3	38,89	23,16	55,53	1	-41,94	18,51	50,47	47,69	0,00	100,98	6,07	
							4	44,70	23,72	81,80	1	-54,19	17,01	96,48	85,95	3,32	147,16	5,47	
							2	58,87	25,12	113,10	2	-52,27	16,39	98,57	7,02	0,00	147,16		
							5	58,87	25,12	113,10	1	-19,50	15,41	110,74	27,97	2,75	66,78	7,05	
							2	58,87	25,12	113,10	2	-19,69	15,34	112,85	2,73	0,00	66,78		
Média		0,00	1,88	68,99	563,16	1,20	1,20								54,60	0,56	103,18	5,68	
Desvio Padrão		0,00	1,64	14,38	125,04	0,20	0,41								34,34	1,24	45,37	1,93	