

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

VALDINEI LUÍS BELINI

**Manipulação de objetos orientada pela aplicação: abordagem assistida por
uma rede neural artificial**

São Carlos

2010

VALDINEI LUÍS BELINI

**Manipulação de objetos orientada pela aplicação: abordagem assistida por
uma rede neural artificial**

Tese apresentada à Escola de Engenharia
de São Carlos da Universidade de São
Paulo, como parte dos requisitos para a
obtenção do título de Doutor em Engenharia
Mecânica.

Área de Concentração:
Engenharia Mecatrônica

Orientador:

Prof. Dr. Glauco Augusto de Paula Caurin

São Carlos

2010

Nome: BELINI, V. L.

Título: Manipulação de objetos orientada pela aplicação: abordagem assistida por uma rede neural artificial

Tese apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para a obtenção do título de Doutor em Engenharia Mecânica.

Aprovado em:

Banca Examinadora

Profa. Dra. Roseli Aparecida Francelin Romero Instituição: ICMC-USP, São Carlos

Julgamento: _____ Assinatura: _____

Prof. Dr. Denis Fernando Wolf Instituição: ICMC-USP, São Carlos

Julgamento: _____ Assinatura: _____

Prof. Dr. Marcelo Becker Instituição: EESC-USP, São Carlos

Julgamento: _____ Assinatura: _____

Prof. Dr. Edilson Reis Rodrigues Kato Instituição: UFSCAR, São Carlos

Julgamento: _____ Assinatura: _____

Prof. Dr. Glauco Augusto de Paula Caurin Instituição: EESC-USP, São Carlos

Julgamento: _____ Assinatura: _____

PUBLICAÇÕES

BELINI, V. L.; MATSUO, D. T.; PEDRO, L. M.; STÜCHELI, M.; CAURIN, G. A. P.; PURQUERIO, B. M. **Integrated System for the development and test of robot hands and grippers**. Proceedings of the XIII International Symposium on Dynamic Problems of Mechanics (DINAME 2009), ABCM, Angra dos Reis-RJ, Brazil, March 2nd - March 6th, 2009.

BENANTE, R. C.; PEDRO, L. M.; MASSARO, L. C.; BELINI, V. L.; ARAÚJO, A. F. R.; CAURIN, G. A. P. **A self-organizing state trajectory planner applied to an anthropomorphic robot hand**. Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems San Diego, CA, USA, Oct 29 - Nov 2, pp. 3082 – 3087, 2007.

FOLHA DE JULGAMENTO

Candidato: Engenheiro VALDINEI LUIS BELINI.

Tese defendida e julgada em 07.10.2010 perante a Comissão Julgadora:

VIDEO CONFERENCIA APROVADO
Prof. Associado **GLAUCO AUGUSTO DE PAULA CAURIN** – (Orientador)
(Escola de Engenharia de São Carlos/USP)

marcelo becker A PROVA DO
Prof. Dr. **MARCELO BECKER**
(Escola de Engenharia de São Carlos/USP)

Roseli Ap. Francelin Romero APROVADO
Prof^a. Associada **ROSELI APARECIDA FRANCELIN ROMERO**
(Instituto de Ciências Matemáticas e de Computação/USP)

Denis Fernando Wolf APROVADO
Prof. Dr. **DENIS FERNANDO WOLF**
(Instituto de Ciências Matemáticas e de Computação/USP)

Edilson Reis Rodrigues Kato Aprovado
Prof. Dr. **EDÍLSON REIS RODRIGUES KATO**
(Universidade Federal de São Carlos/UFSCar)

Jonas de Carvalho
Prof. Associado **JONAS DE CARVALHO**
Coordenador do Programa de Pós-Graduação em
Engenharia Mecânica

Geraldo Roberto Martins da Costa
Prof. Titular **GERALDO ROBERTO MARTINS DA COSTA**
Presidente da Comissão da Pós-Graduação da EESC

*Dedico este trabalho aos meus pais
Claúdio Belini (In memoriam) e Rita Aparecida Rezende,
exemplos de humildade, honestidade e perseverança.*

AGRADECIMENTOS

Ao Professor Glauco pela orientação e aceitação no programa de doutorado durante o meu período de exílio da Engenharia Elétrica.

Aos Professores Dra. Roseli Francelin, Dr. Denis Wolf, Dr. Edilson Kato e Dr. Marcelo Becker pela paciência na leitura do texto e participação no exame de defesa.

Ao Leandro Fernandes (Führer) pela paciência e sabedoria de um mestre e disponibilidade digna de um amigo em investir um sábado inteiro, de sua comprometida agenda, dedicado a me ajudar na configuração da implementação computacional do método de minimização Levenberg-Marquardt.

Ao compreensível e sereno técnico da informática, Leandro, pelas inúmeras atenções dedicadas à compilação, instalação e atualização das ferramentas computacionais fundamentais para o desenvolvimento deste projeto.

À professora Daniela Favarão, a qual eu não tenho idéia do tempo que ela investiu na leitura das minhas tentativas em registrar no papel o que acontecia diante de uma tela de computador. Sem as suas inteligentes, ponderadas e criativas sugestões, talvez esse trabalho descrevesse outro assunto.

Ao amigo Kleber pela atenção dedicada durante as minhas insistências em aceitar os paradigmas da linguagem de programação C/C++.

Ao grande Xamã Jorge Felix, profundo conhecedor da linguagem universal dos bits, por sua dedicação em compilar o simulador GraspIt!, estudado no início do programa de doutorado.

E por último, mas não menos importante, à amiga e companheira Andréia Mayumi Oshiro, detentora da compreensão, paciência, irradiante senso de humor, dignos da estimável sabedoria oriental.

E, para não cometer a injustiça de esquecer alguns amigos e colegas do laboratório, me reservo ao direito de omitir os nomes daqueles que de modo direto ou indireto contribuíram para a realização deste trabalho.

Muito obrigado a todos vocês!

Os problemas procuram por aqueles que têm capacidade para resolvê-los.

(Provérbio Chinês – Autor desconhecido)

RESUMO

BELINI, V. L. **Manipulação de objetos orientada pela aplicação: abordagem assistida por uma rede neural artificial.** 185 p. Tese (Doutorado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2010.

Motivado pelo fato de que a escolha correta da forma preênsil para mãos robóticas antropomórficas representa aspectos relevantes no planejamento de estratégias de manipulação de objetos, este trabalho: desenvolve um algoritmo de pegadas de objetos compatíveis com a tarefa a ser realizada e um método computacional de ajuste de superelipsóides a superfícies de objetos sintéticos 3D para modelar as primitivas resultantes do estágio de segmentação manual do objeto. Essa representação volumétrica contempla as atrativas propriedades de um modelo compacto, controlável e intuitivo, capaz de modelar uma ampla variedade de geometrias simétricas básicas a partir de 11 parâmetros. Tendo em vista que a acomodação de superelipsóides a nuvens de pontos caracteriza desafios típicos de mínimos quadrados, um breve estudo sobre o método numérico de Levenberg-Marquardt, utilizado na minimização da função objetivo adotada, se mostrou necessário para maximizar a sua eficiência. O desempenho e a eficiência do programa computacional desenvolvido foram demonstrados na modelagem experimental de cinco primitivas organizadas em diferentes configurações. As pequenas distorções, consideradas aceitáveis para a proposta do trabalho, evidenciaram a robustez da representação de formas elementares empregando superelipsóides sem deformações. Encerrada a etapa de modelagem das primitivas, um método de auto-aprendizagem supervisionada

interpreta as informações de forma e tamanho de tais formas básicas para sugerir a pega de uma das primitivas que compõem o objeto e a forma preênsil compatível com a tarefa desejada.

Palavras-chave: Manipulação; robótica; tarefa.

ABSTRACT

BELINI, V. L. **Object grasping oriented by the application: an approach assisted by an artificial neural network.** 185 p. Thesis (Doctoral) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2010.

Motivated by the fact that the correct choice of prehensile form for anthropomorphic robotic hand describes relevant aspects in planning strategies of object manipulation, this work develops an algorithm of grasping objects to match the task requirements and a computational method capable of fitting superellipsoids to surfaces of synthetic 3D objects to model the resulting primitives from the manual segmentation stage of the object. This volumetric representation describes the attractive properties of a compact, controllable and intuitive model, which is capable of modeling a wide variety of standard geometric solids by using eleven parameters. Since the superellipsoid fitting process characterizes typical least squares problems, a brief review about Levenberg-Marquardt numerical method, employed for the adopted objective function minimization, was necessary to maximize its efficiency. The performance and efficiency of the developed program were demonstrated in the experimental modeling of five symmetrical primitives shapes organized in several configurations. The small distortions, considered acceptable to the work proposal, confirm the robustness of representing elementary shapes employing superellipsoids without deformations. After finishing the primitive modeling stage, a supervised learning method interprets the size and shape information of such basic shapes to suggest the grasp of one constituent single parts and the task-compatible prehensile form.

Keywords: Manipulation; robot; task.

SUMÁRIO

Lista de Figuras	xvi
Lista de Tabelas	xxi
Lista de Abreviações	xxiii
Lista de Símbolos	xxv
Capítulo I – Introdução	1
1.1 Motivação	6
1.2 Relevância	8
1.3 Objetivo	10
1.4 Organização	10
Capítulo II - Introdução ao estado da arte na modelagem e síntese de objetos 3D baseada em superelipsóides	13
Parte I - Modelagem automática de nuvem de pontos empregando superelipsóides	23
Capítulo III - Representação de objetos por superelipsóides	25
3.1 Introdução às <i>Superquadrics</i>	25
3.2 Posições genéricas no espaço tridimensional	33
3.4 Definição e interpretação geométrica da função objetivo	36

3.5. Cálculo das estimativas iniciais dos parâmetros da superelipsóide	44
3.5.1 Cálculo do baricentro da nuvem de pontos	44
3.5.2 Estimativa inicial da orientação da nuvem de pontos	45
3.5.3 Determinação do eixo de inércia principal da superelipsóide	47
3.5.4 Estimativa do tamanho da superelipsóide candidata	48
3.5.5 Resultados numéricos	50
3.6 Método numérico de Levenberg-Marquardt	51

Capítulo IV - Programa de ajuste de superelipsóides **59**

4.1 Fluxograma de ajuste de superelipsóides	59
4.1.1 Nuvem de pontos	61
4.1.2 Cálculo da posição global da nuvem de pontos	61
4.1.3 Cálculo da orientação da nuvem de pontos	63
4.1.4 Mudança do sistema de coordenadas da nuvem de pontos	64
4.1.5 Estimativa inicial do tamanho da nuvem de pontos	65
4.1.6 Identificação dos eixos de inércia	66
4.1.7 Atribuição dos parâmetros de tamanho aos eixos coordenados.	69
4.1.8 Ajuste dos quadrantes trigonométricos para os ângulos de Euler.	69
4.1.9 Declaração das condições iniciais dos parâmetros da função objetivo	73
4.1.10 Função objetivo a ser minimizada	73
4.1.11 Rotina de minimização Levenberg-Marquardt	74
4.1.12 Verificação do erro do ajuste	77
4.2 Programa gráfico	78
4.2.1 Declaração da expressão paramétrica	78
4.2.2 Parametrização regular da superfície da superelipsóide	80

Capítulo V – Modelagem automática de nuvem de pontos	83
5.1 Exemplo numérico	83
5.1.1 Estimativa da posição da nuvem de pontos	84
5.1.2 Estimativa da orientação da nuvem de pontos	84
5.1.3 Determinação do eixo de inércia principal da superelipsóide	85
5.1.4 Estimativa do tamanho da nuvem de pontos	86
5.2. Minimização da função objetivo empregando o método de Levenberg-Marquardt	87
5.3 Coerência na interpretação dos modelos ajustados	100
Parte II - Pega de objetos de uso cotidiano assistida por uma rede neural artificial	105
Capítulo VI – Introdução ao estado da arte do planejamento de estratégias para a pega de objetos	107
6.1 Métodos analíticos de pegadas de objeto orientadas pela tarefa	108
6.2 Métodos empíricos de pegadas de objeto orientadas pela tarefa	111
Capítulo VII – Estratégia de pega de objetos orientada à tarefa	117
7.1 Pegadas de objetos compatíveis com a tarefa	119
7.2 Resultados experimentais	127
Capítulo VIII - Considerações finais	137
8.1 Contribuições do trabalho	138
8.2 Sugestões para futuros trabalhos	139
Referências	141

Apêndice A - Conjuntos de dados não normalizados utilizados no treinamento e validação da RNA	151
Apêndice B - Conjuntos de dados normalizados utilizados no treinamento e validação da RNA	159
Apêndice C - Conjuntos de dados normalizados e aleatoriamente distribuídos para o processo de aprendizagem da RNA	167
C1: Base de dados para o treinamento (95 objetos)	168
C2: Base de dados para a validação (41 objetos)	173
Apêndice D - Conjuntos de dados utilizados para o teste da RNA envolvendo objetos pertencentes à mesma categoria daqueles utilizados no processo de aprendizagem	175
D1: Dados não normalizados	176
D2: Dados normalizados	177
Apêndice E – Arquivo de resultados para o teste envolvendo objetos pertencentes à mesma categoria daqueles adotados na base de aprendizagem da RNA	179
Apêndice F – Arquivo de saída dos resultados para o teste da RNA envolvendo itens de categorias diferentes daquelas utilizadas no processo de aprendizagem	181

Lista de Figuras

Figura 1.1- Protótipos de mãos robóticas antropomórficas	2
Figura 1.2 – A escolha pela parte manipulável de um objeto é influenciada pela forma dos volumes parciais que compõem este item, independentemente das suas orientações: (a) uma xícara; (b) um balde e (c) uma maleta. Todos esses objetos são pegos pelas respectivas partes curvadas para ações reveladas pelo ato de “pegar para servir”, enquanto atividades relacionadas às “mudanças de posições” podem sugerir os volumes cilíndricos para (a) e (b) e cúbico para (c)	5
Figura 1.3 – Elenco de objetos que podem ser simulados por superelipsóides. Tanto as partes claras quanto as escuras caracterizam volumes possíveis de serem modeladas e expõem as regiões tipicamente escolhidas por agentes humanos na realização de determinadas tarefas	6
Figura 1.4 – Visão geral da proposta: (a) objeto de interesse; (b) representação do modelo geométrico das partes que compõem o objeto original e modelagem das nuvens de pontos empregando superelipsóides; (c) vetores dos parâmetros das primitivas; (d) escolha da primitiva adequada para atender as tarefas demandas por um agente humano	9
Figura 2.1 – Exemplos de superelipsóides. (Chevalier, 2004)	13
Figura 2.2 – (a) e (b) primitivas cônicas produzidas pela deformação <i>tapering</i> , (c) e (d) primitivas dobradas resultantes da deformação <i>bending</i>	13
Figura 2.3 – (a) Fonte de água no formato de superelipse em frente ao edifício Hötorget; (b) Praça em desnível e interior de um centro de compras (c) na Avenida Sergels Torg (Estocolmo - Suécia)	14
Figura 2.4 – (a) nuvem de pontos do objeto; (b) estágio intermediário do processo <i>recovery-and-select</i> , (c) modelo final recuperado. (Leonardis <i>et al.</i> , 1997)	15

Figura 2.5 – Síntese de objetos reais empregando <i>superquadrics</i> estendidas: (a) travesseiro; (b) fatia de pão; (c) cogumelo. (Zhou e Kambhamettu, 2000)	16
Figura 2.6 – Modelagem de um boneco de neve: (a) nuvem de pontos do objeto; (b) Segmentação dos dados; (c) aproximação do objeto por duas <i>superquadrics</i> . (Bhabhrawala e Krovi, 2005)	17
Figura 2.7 – Reconhecimento de objetos: (a) cenário original; (b) modelagem dos objetos escolhidos; (c) cenário reconstruído com <i>superquadrics</i> ; (d) objetos reconhecidos. Krivic (2006)	18
Figura 2.8 – Processo de ajuste utilizando <i>superquadrics</i> estendidas: (a) nuvem de pontos do objeto; (b) ajuste inicial por uma superfície simétrica; (c) modelo representativo do objeto. (Bhabhrawala <i>et al.</i> 2007)	19
Figura 2.9 – (a) Escolha manual de um ser humano a partir de uma cena principal. (b) homem segmentado; (c) representação composta por <i>superquadrics</i> . (Strand e Dillmann, 2009)	19
Figura 3.1 – Exemplos de superelipses. (Chevalier, 2004)	22
Figura 3.2 – (a) Esfera unitária obtida pelo produto esférico entre um círculo contido no plano (x, y) e um semicírculo projetado num plano ortogonal; (b) cilindro resultante do produto esférico entre um círculo e um retângulo	24
Figura 3.3 - Exemplos de superelipsóides de acordo com a variação dos parâmetros de forma $\varepsilon_1, \varepsilon_2$ quando $a_1 = a_2 = a_3 = 1,0$: as formas geométricas modelam um cubo para $\varepsilon_1 = \varepsilon_2 = 0,1$; um cilindro para $\varepsilon_1 = 0,1$ e $\varepsilon_2 = 1,0$; um paralelepípedo para $\varepsilon_1 = 0,1$ e $\varepsilon_2 = 1,9$ e uma esfera para $\varepsilon_1 = \varepsilon_2 = 1,0$, entre outros, como pode ser observado	25
Figura 3.4 – A família de <i>superquadrics</i> inclui as formas geométricas de uma superelipsóide (a), superhiperbolóides de uma folha (b) e duas folhas (c) e <i>supertoróides</i> (d). (Barr, 1981) ..	28
Figura 3.5 - Representação de uma superelipsóide em uma posição qualquer no espaço. Os eixos do sistema de coordenadas local são denotados por x, y e z , enquanto os	

correspondentes no sistema de coordenadas global são caracterizados por X , Y e Z . A matriz de transformação T descreve a mudança do sistema de coordenadas local para o sistema global. Em contrapartida, a matriz inversa T^{-1} promove a transformação do sistema de coordenadas global para o sistema de coordenadas local	29
Figura 3.6 – Comparação entre a distância euclidiana $\ PM\ $ de um ponto à superfície da superelipsóide e a distância euclidiana radial $\ PP_0\ $ considerada como aproximação do cálculo da distância exata. O ponto M corresponde à projeção ortogonal do ponto P , enquanto o ponto P_0 descreve a intersecção entre a reta definida pela origem O e o ponto P à superfície da superelipsóide S	34
Figura 3.7 – Interpretação geométrica da distância euclidiana radial. O vetor r_0 representa a separação entre a origem da superelipsóide e o ponto (X, Y, Z) , enquanto o vetor r_S indica a distância da origem à superfície mencionada. O módulo da diferença desses dois vetores caracteriza a distância euclidiana radial, d	36
Figura 3.8 - Superelipsóide localizada na origem do sistema de coordenadas e a relação geométrica com o auto-sistema definido pela matriz de covariância. Os autovetores correspondentes v_1 , v_2 e v_3 indicam as direções dos eixos de inércia ao longo dos eixos coordenados X , Y e Z , respectivamente	42
Figura 3.9 – Interpretação geométrica do critério utilizado para definir a direção do eixo de inércia principal: (a) e (b) linha tracejada indica a direção do eixo de menor inércia para os objetos alongados; (c) e (d) orientação do eixo de maior inércia para itens de perfis baixos	44
Figura 3.10 - Conjunto de entrada e de saída da função de minimização <code>dlevmar_bc_dif</code> do método de Levenberg-Marquardt	51
Figura 4.1 - Fluxograma computacional desenvolvido para o processo de ajuste das superelipsóides	55
Figura 4.2 - Captura de tela de um <i>arquivo.txt</i> utilizado para registrar as coordenadas dos pontos	56

- Figura 4.3 – Captura de tela após a execução do programa principal para ajustar uma superelipsóide a uma caixa irregular 70
- Figura 4.4 - Parametrização da malha da superfície de modelagem das superelipsóides. (a) grande concentração de linhas ocorre próximo às bordas do modelo. (b) parametrização regular das linhas e (c) distribuição uniforme representada por pontos 76
- Figura 5.1 – (a) paralelepípedo e o sistema de coordenadas original; (b) nuvem de pontos correspondente ao objeto 78
- Figura 5.2 - Interpretação geométrica da direção do eixo de inércia principal da nuvem de pontos: (a) objeto com sistema de coordenadas original; (b) direção do eixo de inércia indicada pelo eixo z do novo sistema de coordenadas 80
- Figura 6.1 – Tarefa Peg-in-hole (Li e Sastry, 1988) 108
- Figura 6.3 – Ilustração das diferentes distribuições das forças e torques que podem atuar sobre o objeto. A soma da intensidade normalizada de todos os vetores de força resulta em um valor unitário. 109
- Figura 6.4 – A proposta baseada no índice de qualidade da pega maximiza o valor de α de uma determinada combinação de forças e torques aplicados na execução de uma determinada tarefa contida no interior do espaço W permitido pelas *wrenches* da pega (Haschke *et al.*, 2005) 110
- Figura 6.5 – Formas preêenseis orientadas pela tarefa. Pratz *et al.* (2007) 110
- Figura 6.6 – Comparação entre pegas humanas e versões sintéticas aleatórias. A barra vertical da imagem (b) indica o índice de qualidade de uma pega humana para a caixa ilustrada no gráfico. 111
- Figura 6.7 – Classes de objetos utilizadas no estágio de treinamento da rede neural artificial. A linha superior da figura ilustra os objetos originais, enquanto a corresponde linha inferior destaca a identificação das regiões favoráveis à pega (Saxena *et al.*, 2008) 113
- Figura 6.8 – Síntese de pegas baseada na similaridade das formas. O sistema de comparação de formas procura na base de dados a melhor pose da mão que atende ao

objeto. As poses representativas 1, 2 e 3 são, nesta ordem, as configurações dos dedos para agarrar um mouse, uma jarra e uma lâmpada, Li e Pollard (2007) 114

Figura 6.9 – Objetos utilizados no teste do sistema de auto-aprendizagem: (a) objeto original; (b) segmentação em formas básicas; (c) modelagem por superelipsóides; (d) pontos de contato sobre as superfícies dos objetos. El-Khoury *et al.* (2009) 115

Figura 7.1: Visão geral da seleção da primitiva compatível com a tarefa desejada. 120

Figura 7.2 – Comparação de desempenho dos melhores resultados para os algoritmos de aprendizagem: (a) *Backpropagation* e (b) *Resilient Propagation*, para a mesma topologia de rede. 127

Figura 7.3 – Fragmento do arquivo de saída utilizado na interpretação do comportamento da rede neural artificial. 129

Lista de Tabelas

Tabela 3.1 – Parâmetros exatos dos objetos utilizados nas simulações	46
Tabela 3.2 – Parâmetros estimados para demonstrar a precisão das simulações	47
Tabela 5.1 – Autovalores e autovetores para o paralelepípedo do exemplo numérico	79
Tabela 5.2 – Resumo dos resultados obtidos para o paralelepípedo do exemplo numérico	81
Tabela 5.3 - Estimativa da orientação do eixo de inércia principal das primitivas	83
Tabela 5.4 – Superelipsóide ajustada a uma esfera	86
Tabela 5.5 – Superelipsóide ajustada a um cubo	87
Tabela 5.6 – Superelipsóide ajustada a um cilindro alto	88
Tabela 5.7 – Superelipsóide ajustada a um paralelepípedo	89
Tabela 5.8 – Superelipsóide ajustada a um paralelepípedo tombado	90
Tabela 5.9 – Superelipsóide ajustada a um cilindro tombado	91
Tabela 5.10 – Superelipsóide ajustada a um cilindro baixo	92
Tabelas 5.11 – Modelagem de um paralelepípedo organizado em seis configurações	95
Tabela 7.1: Padronização adotada nos arquivos de treinamento, validação e teste da RNA na descrição das primitivas, tarefa e saídas desejadas.	122
Tabela 7.2: Conjunto dos oito padrões utilizados no processo de aprendizagem da RNA. A seta indica a primitiva que deve ser escolhida pela RNA, enquanto as linhas tracejadas denotam a seleção de um ser humano para a mesma tarefa.	124
Tabela 7.3: Sequência de preparação dos padrões de entrada para o processo de aprendizagem e fases de testes da RNA.	125

Tabela 7.4 – Objetos utilizados para demonstrar a capacidade da RNA em trabalhar com itens similares aqueles adotados no processo de treinamento. As linhas tracejadas indicam as regiões tipicamente escolhidas pelos seres humanos, enquanto a seta aponta a seleção da RNA para a mesma tarefa. 131

Tabela 7.5 – Objetos inéditos utilizados para testar a capacidade de generalização da RNA. A seta indica a escolha da rede neural, enquanto as linhas tracejadas denotam a região tipicamente selecionada pelos seres humanos para a mesma tarefa. 132

Lista de Símbolos

a_i ($i = 1, 2, 3$): Parâmetro de tamanho das primitivas

\hat{a}_1 : Estimativa do tamanho da nuvem de pontos na direção do eixo x

\hat{a}_2 : Estimativa do tamanho da nuvem de pontos na direção do eixo y

\hat{a}_3 : Estimativa do tamanho da nuvem de pontos na direção do eixo z

C : Matriz de covariância

d : Distância

F : Função implícita

f : Relação funcional

i : Variável da quantidade de pontos

I : Matriz de inércia

J : Matriz de Jacobi

J : Variável da quantidade de pontos para a estimativa de tamanho

K : Escalar de proporcionalidade

S : Superfície da superelipsóide

x : Coordenada do ponto em relação ao eixo x do sistema de coordenadas local

y : Coordenada do ponto em relação ao eixo y do sistema de coordenadas local

z : Coordenada do ponto em relação ao eixo z do sistema de coordenadas local

X : Coordenada do ponto em relação ao eixo x do sistema de coordenadas global

Y : Coordenada do ponto em relação ao eixo y do sistema de coordenadas global

Z : Coordenada do ponto em relação ao eixo z do sistema de coordenadas global

\bar{X} : Coordenada do baricentro em relação ao eixo x do sistema de coordenadas global

\bar{Y} : Coordenada do baricentro em relação ao eixo y do sistema de coordenadas global

\bar{Z} : Coordenada do baricentro em relação ao eixo z do sistema de coordenadas global

X_i : Coordenadas dos pontos em relação ao eixo x

Y_i : Coordenadas dos pontos em relação ao eixo y

Z_i : Coordenadas dos pontos em relação ao eixo z

∂ : Derivada parcial da função implícita

δ_p : Fator de atualização do algoritmo de Levenberg-Marquardt

ϵ : Erro entre a solução desejada e aquela apresentada pelo método numérico

ε : Parâmetro de forma das primitivas

κ_1 : Limite da magnitude do gradiente

κ_2 : Patamar de mudança relativa da magnitude do fator de atualização

κ_3 : Limiar inferior para o erro de ajuste

K_{\max} : Quantidade máxima de iterações

μ : Termo de amortecimento do método de Levenberg-Marquardt

η : Ângulo de geração das superfícies no plano xz

ω : Ângulo de geração das superfícies no plano xy

ϕ : Ângulo de Euler em relação ao eixo z

θ : Ângulo de Euler em relação ao eixo y

ψ : Ângulo de Euler em relação ao novo eixo z

$\hat{\phi}$: Estimativa do ângulo de Euler em relação ao eixo z

$\hat{\theta}$: Estimativa do ângulo de Euler em relação ao eixo y

$\hat{\psi}$: Estimativa do ângulo de Euler em relação ao novo eixo z

\otimes : Produto esférico

N : Quantidade de pontos

O : Origem do sistema de coordenadas local

P : Ponto externo à superfície

ρ : Vetor de parâmetros

ρ_0 : Estimativa do vetor de parâmetros da superelipsóide

P_0 : Ponto pertencente à superfície para o cálculo da aproximação da distância

M : Ponto pertencente à superfície para o cálculo da distância euclidiana

\bar{P} : Vetor de translação

P_j : Vetor das coordenadas dos pontos do conjunto de dados

ρ_x : Coordenada do vetor translação na direção do eixo x

ρ_y : Coordenada do vetor translação na direção do eixo y

ρ_z : Coordenada do vetor translação na direção do eixo z

$\hat{\rho}_x$: Estimativa da coordenada do baricentro em relação ao eixo x

$\hat{\rho}_y$: Estimativa da coordenada do baricentro em relação ao eixo y

$\hat{\rho}_z$: Estimativa da coordenada do baricentro em relação ao eixo z

R : Matriz de rotação

R^{-1} : Matriz inversa de rotação

R^T : Matriz transposta de rotação

r_0 : Vetor entre a superelipsóide e um ponto externo

r_S : Vetor entre a origem do sistema de coordenadas global e a superelipsóide

T : Matriz de transformação de coordenadas

T^{-1} : Matriz inversa de transformação de coordenadas

v : Autovetor da matriz de covariância

λ : Autovalor da matriz de covariância

$v_{i,j}$: J -ésimo componente do i -ésimo autovetor

\mathcal{A} : Vetor de parâmetros das superelipsóides

2D: Bidimensional

3D: Tridimensional

Lista de Abreviações

EOF: *Error of Fitting*

GB: Giga Byte

GHZ: Giga Hertz

PMC: Perceptron Multicamadas

RAM: Random Access Memory

RNA: Rede Neural Artificial

WWW: World Wide Web

Introdução

A capacidade de um manipulador robótico de agarrar um objeto e promover uma mudança em sua orientação geométrica é um dos aspectos primários que potencializa a habilidade de um robô em interagir e modificar o ambiente no qual ele está inserido. Cenários perfeitamente controlados, tais como aqueles existentes em aplicações industriais, permitem que garras robóticas dedicadas desempenhem suas atividades de modo muito eficiente.

De fato, manipuladores industriais trabalham em ambientes altamente controlados, uma vez que são programados para executarem uma seqüência de tarefas pré-estabelecidas. Desse modo, os sistemas robóticos são extremamente eficientes em aplicações que envolvem o deslocamento de uma grande quantidade de carga. No entanto, eles não apresentam a destreza dos movimentos precisos demandados pela manipulação de objetos desconhecidos.

O elenco restrito de objetos que podem ser pegos de modo suficientemente estável para permitir a execução de determinadas tarefas mostra-se uma limitação de manipuladores robóticos. Conseqüentemente, há um crescente e desafiante desejo por manipuladores robóticos genéricos o suficiente para oferecer reais possibilidades de re-configurações mecânicas, no sentido de ampliar substancialmente a variedade de objetos que podem ser efetivamente manipulados.

A possibilidade de inserção de assistentes robóticos móveis em ambientes essencialmente não controlados descreve um dos interesses mais promissores da robótica atual. Para isso, esses sistemas devem apresentar uma capacidade mínima para desenvolverem um elenco limitado de decisões, mas suficientemente elaboradas para superar as circunstâncias impostas pela dinâmica do ambiente.

Para concluir esses objetivos é necessário desenvolver uma série de funcionalidades integradas a essas sofisticadas plataformas robóticas. Além da

mobilidade, a habilidade de um braço robótico em configurar, de forma totalmente autônoma, os dedos de uma mão para modificar a pose de um objeto genérico caracteriza um aspecto decisivo para permitir que tais robôs interajam e modifiquem o ambiente não estruturado no qual estejam inseridos. No entanto, para que esse comportamento possa ser explorado de forma aproveitável, tais habilidades de manipulação devem ser compatíveis, ou, pelo menos muito próximas, daquelas exibidas pela dinâmica da mão humana.

Estratégias que tem permitido o desenvolvimento de tal grau de precisão têm adotado mãos robóticas antropomórficas com dedos independentemente articuláveis, tais como a alemã DLR-HIT (Butterfaß *et al.*, 2004) da figura 1.1(a); os protótipos japoneses GIFU-III (Mouri *et al.*, 2002) (ver figura 1.1(b)) e KH Hand type S (Mouri *et al.*, 2005) (ver figura 1.1(c)); a italiana *Cyberhand* (Carrozza *et al.*, 2003) (ver figura 1.1(d)); a Americana *Robonaut* (Martin *et al.*, 2004) (ver figura 1.1(e)); e o modelo nacional *kangüera* (Belini *et al.*, 2009) da figura 1.1(f).

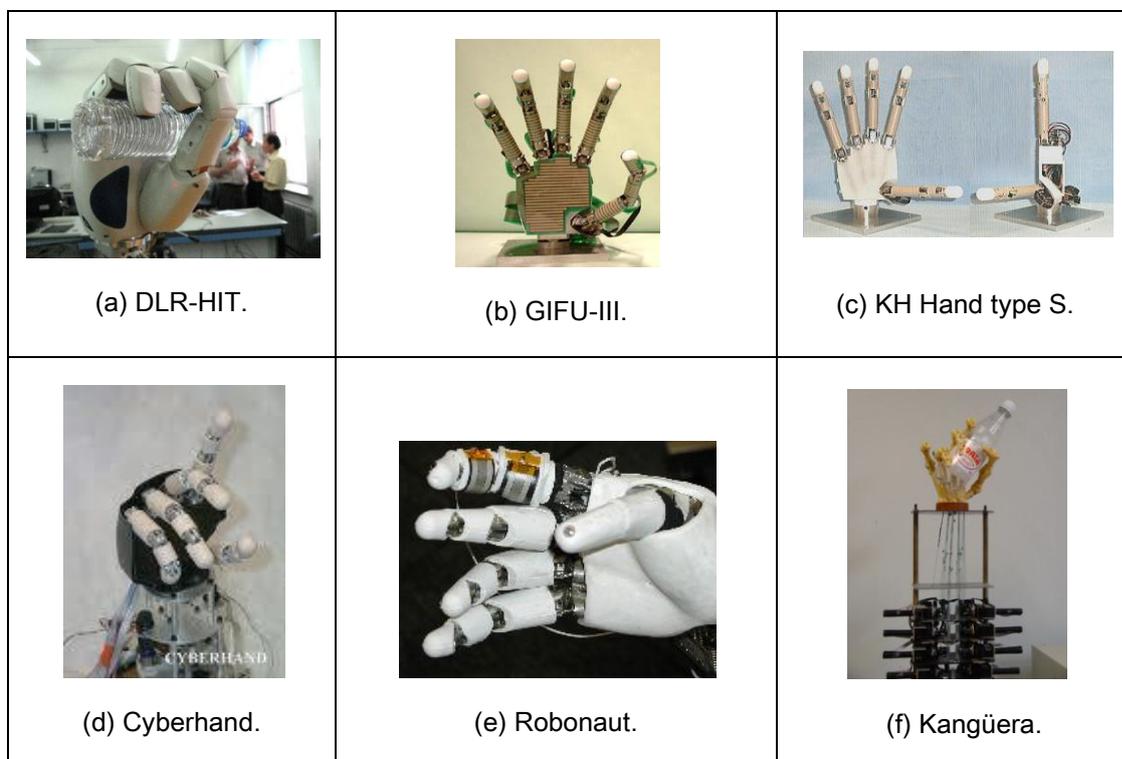


Figura 1.1- Protótipos de mãos robóticas antropomórficas.

Ao mesmo tempo em que tais elaborados mecanismos são projetados como potenciais soluções, eles contribuem para o aumento da complexidade da análise e síntese da cinemática e dinâmica de todo o sistema. Além dos desafios tecnológicos inerentes à pesquisa, desenvolvimento e construção de tais protótipos, a manipulação habilidosa contribui para um nível de dificuldade adicional, denotado por planejamentos computacionalmente onerosos, geralmente exigidos para pegar determinados objetos.

Diversos métodos para a síntese e análise de configurações para os dedos de uma mão robótica têm sido propostos. Algumas dessas abordagens consideram a cinemática e a dinâmica desses mecanismos. Desse modo, essas propostas planejam as posturas dos dedos necessárias para garantir uma pega estável e adaptada às operações de *pick and place*, supostamente independente da tarefa a ser realizada. De fato, poucas propostas (Pollard, 1997; Borst, 2004; e Pratz, 2007) incluem a dependência da tarefa no planejamento da configuração dos dedos. No entanto, a elevada complexidade computacional envolvida na descrição analítica de determinadas tarefas revela a restrição dessas técnicas. Desse modo, enquanto a seleção de uma pega suficientemente ótima, orientada à tarefa, é muito simples para uma mão humana, ela ainda é um processo complicado para uma versão robótica (Sahbani e El-Khoury, 2008). Em contrapartida, métodos empíricos evitam tais complexidades procurando reproduzir as estratégias humanas. Em particular, essas técnicas podem ser classificadas em:

- sistemas baseados na observação do objeto a ser pego (Pelosof *et al.*, 2004; Li *et al.*, 2007; e Saxena *et al.*, 2008); e
- sistemas conduzidos pela observação de um ser humano realizando uma determinada tarefa (Fischer *et al.*, 1998; Oztop e Arbib, 2002; Ekvall e Kragic, 2004; Kyota *et al.*, 2005; Hueser *et al.*, 2006).

A primeira abordagem geralmente aprende a associar às características dos objetos a formas preênsais para os dedos da mão, enquanto na segunda, o robô observa o comportamento de um humano pegando um objeto e tenta reproduzi-lo.

Sistemas baseados na observação de objetos (Pelosof *et al.*, 2004; Li *et al.*, 2007; e Saxena *et al.*, 2008) são adaptados para novos itens, mas produzem uma grande quantidade de possibilidades para as posições dos dedos da mão robótica e falham na seleção da melhor configuração para atender a uma dada tarefa. Desse modo, tais técnicas demonstram dificuldades similares àquelas inerentes aos métodos analíticos para modelar uma tarefa de maneira totalmente autônoma. Portanto, esses sistemas não representam uma abordagem totalmente independente para objetos desconhecidos (Sahbani e El-Khoury, 2008). Conseqüentemente, metodologias capazes de aprender a associar a pega de um objeto desconhecido a uma determinada tarefa ainda se revelam um dos maiores desafios da robótica atual.

Inspirado na proposta de aprendizagem de Sahbani e El-Khoury (2008), o presente trabalho teve seu foco direcionado ao desenvolvimento de uma estratégia de manipulação capaz de aprender a combinar ações demandadas por agentes humanos a objetos desconhecidos pelo sistema robótico. Como os itens de uso cotidiano têm a sua forma normalmente construída para atender às necessidades dos seres humanos, algumas de suas partes são projetadas para permitirem maior destreza de manipulação. Embora sejam particularmente bem concebidas para proporcionarem interações com os humanos, tais partes não podem ser facilmente identificadas por agentes robóticos.

Associado à dificuldade do planejamento da configuração adequada para a mão robótica desempenhar uma determinada tarefa, o desafio em identificar componentes geométricos específicos de um objeto constitui uma preocupação adicional, inclusive para itens conhecidos do sistema de manipulação.

Com a finalidade de descrever um objeto por geometrias mais simples e, hipoteticamente adequadas para o manuseio desses exemplares, modelos compactos de representação volumétrica – denominados superelipsóides - serão adotados. Além da reduzida quantidade de parâmetros necessários para descrevê-los, a possibilidade de controlar sua forma geométrica de modo altamente intuitivo justifica tal escolha. Todavia, técnicas de segmentação que normalmente precedem o estágio de

modelagem não serão discutidas neste estudo, mas somente os seus resultados. Diante disso, a proposta considera somente formas básicas isoladas.

Os escalares que caracterizam as superelipsóides de cada região volumétrica do objeto a ser modelado são interpretados por um sistema de auto-aprendizagem responsável pela escolha da parte do objeto a ser pega pela mão robótica. Essa decisão é fundamentada na ação previamente definida que o sistema robótico deve desempenhar, garantindo, desse modo, compatibilidade com a tarefa a ser realizada ao mesmo tempo em que evita a complexidade matemática envolvida na modelagem de tais escolhas.

Em particular, a seleção da parte manipulável é influenciada pelo tamanho e forma geométrica dos volumes parciais que compõem o objeto, independentemente das suas orientações espaciais, como mostra a figura 1.2.



Figura 1.2 – A escolha pela parte manipulável de um objeto é influenciada pela forma dos volumes parciais que o compõem, independentemente das suas orientações: (a) uma xícara; (b) um balde e (c) uma maleta. Todos esses objetos são pegos pelas respectivas partes curvadas para ações reveladas pelo ato de “pegar para usar”, enquanto atividades relacionadas às “mudanças de posições” podem sugerir os volumes cilíndricos para (a) e (b) e cúbico para (c).

A escolha deve ser suficientemente adequada à tarefa a ser realizada, para caracterizar real eficiência da estratégia envolvida no planejamento das configurações dos dedos da mão robótica.

A partir de um reduzido elenco de escalares a modelagem volumétrica empregando superelipsóides permite a simulação de uma ampla variedade de geometrias padrões localizadas em poses genéricas no espaço. Os conceitos e propriedades necessários para modelar objetos 3D utilizando tais superfícies são

estudados, enquanto os parâmetros que descrevem essas superelipsóides serão interpretados pelo algoritmo de manipulação a ser desenvolvido para o encerramento deste projeto.

1.1 Motivação

A inspiração para a realização deste trabalho foi baseada na atrativa possibilidade de modelar uma ampla variedade de formas geométricas básicas empregando um pequeno elenco de parâmetros. A figura 1.3 ilustra alguns objetos de uso cotidiano que podem ser representados pela combinação de primitivas simuladas por superelipsóides.

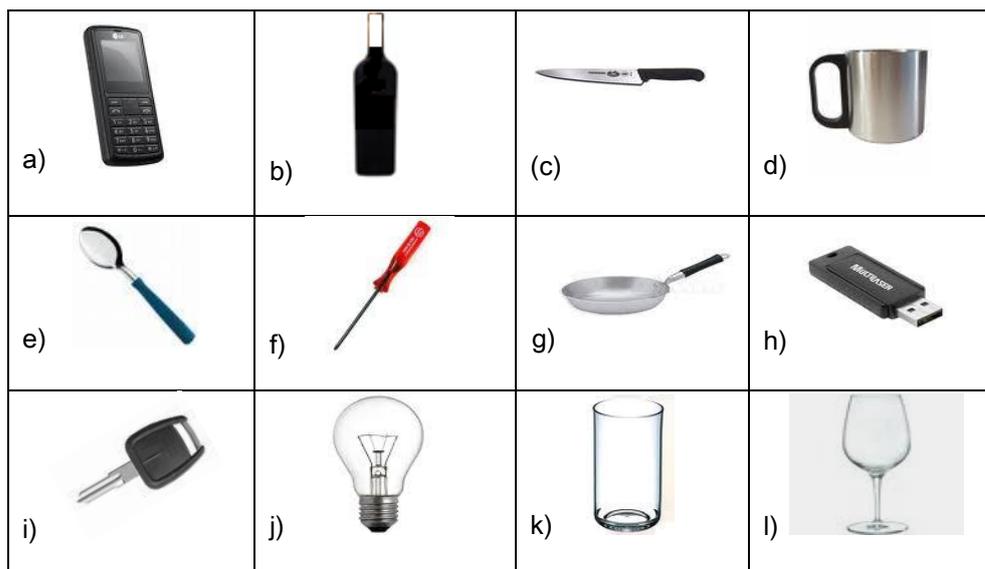


Figura 1.3 – Elenco de objetos que podem ser simulados por superelipsóides. Tanto as partes claras quanto as escuras caracterizam volumes possíveis de serem modelados e expõem as regiões tipicamente escolhidas por agentes humanos na realização de determinadas tarefas.

A estrutura do celular da figura 1.3(a) pode ser representada por um paralelepípedo de bordas ligeiramente arredondadas, enquanto o corpo e o gargalo da garrafa da figura 1.3(b), bem como as duas partes da chave de fenda da figura 1.2(f) e a panela da figura 1.3(g), podem ser corretamente modelados por cilindros de diferentes tamanhos. Em contrapartida, um cilindro e um paralelepípedo, ambos nas

proporções adequadas, podem simular o corpo e a alça da caneca da figura 1.2(d), respectivamente. Além disso, a faca da figura 1.3(c), a colher da figura 1.3(e), a chave de ignição da figura 1.3(i) e a *pen drive* da figura 1.3(h) podem ser representadas por duas primitivas no formato de paralelepípedos. Diferentemente dos demais itens, a lâmpada da figura 1.3 (j) exige a combinação de uma esfera com um pequeno cilindro, ao passo que o copo da figura 1.3(k) pode ser simulado por essa última primitiva. Embora a taça de vinho da figura 1.3(l) seja composta por três partes, a modelagem deste objeto é formada exclusivamente por cilindros.

Uma correspondência estratégica de tarefas demandadas por agentes humanos a determinadas primitivas pode revelar potenciais contribuições na elaboração de metodologias de manipulação de itens de uso cotidiano. Além disso, as limitadas capacidades de memória e processamento dos atuais assistentes robóticos, associadas às restritas disponibilidades de potência das fontes de energia, incentivam o estudo de técnicas alternativas de modelagem para os objetos a serem manipulados.

Entre outras vantagens, alguns dos aspectos atrativos que motivaram o estudo de tal técnica de simulação de objetos podem ser resumidos como:

(i) necessidade de poucos parâmetros para descrever um amplo elenco de formas geométricas quando comparado à grande capacidade de representação desses modelos, e,

(ii) a possibilidade de combinar geometrias primitivas para representar objetos mais complexos.

Com a finalidade de evitar interpretações incompatíveis com aquelas envolvidas no desenvolvimento deste trabalho, os seguintes critérios foram adotados:

- O objetivo do ajuste dos parâmetros das superelipsóides consiste em determinar o melhor modelo geométrico capaz de representar uma determinada nuvem de pontos;
- a distribuição regular dos pontos da nuvem;
- o uso de somente uma primitiva durante cada execução do programa computacional.

Além disso,

- todos os pontos do conjunto pertenciam ao mesmo objeto;
- todos os objetos apresentavam simetria rotacional em torno dos eixos coordenados;
- nenhuma consideração foi feita sobre a obtenção dos pontos que descrevem as primitivas. De fato, os pontos foram obtidos pelo uso de programas específicos de computação gráfica.

1.2 Relevância

Dependendo da tarefa a ser conduzida por um agente humano, a escolha pela parte escura da garrafa da figura 1.3(b) pode ser a mais adequada para alterar a posição deste objeto. Em contrapartida, para citar alguns exemplos, a parte clara deste item deve ser escolhida para o caso em que se tenha o objetivo de guardar este recipiente em um espaço delimitado por um engradado. Tal observação também se torna pertinente no caso da caneca da figura 1.3(d), ou seja, se o interesse consistisse em mudar a sua localização, seria suficiente pegá-la pela região clara, enquanto a atitude de servir a um usuário deveria contemplar a alça deste item.

A proposta de El-Khoury e Sahbani (2008) para identificar as partes tipicamente manipuláveis de objetos de uso geral permitiu a um método de aprendizagem supervisionada gerar pegadas estáveis desses itens. De fato, a partir de um algoritmo de segmentação, os parâmetros das superelipsóides resultantes do processo de modelagem foram interpretados por uma rede neural, responsável pela escolha da região a ser pega. Em particular, tal método tem a capacidade de generalizar a seleção dessas partes a partir do tamanho e forma das primitivas empregadas na modelagem. Todavia, a estratégia de seleção não considerou a tarefa a ser conduzida pelo agente robótico, sugerindo, simplesmente, os volumes parciais geralmente escolhidos pelos seres humanos para propósitos gerais.

Baseado nesta observação, a contribuição deste trabalho se caracteriza pela utilização de um sistema de auto-aprendizagem, suficientemente capaz de associar

tarefas demandadas por agentes humanos às informações geométricas de determinados objetos, como mostra a figura 1.4.

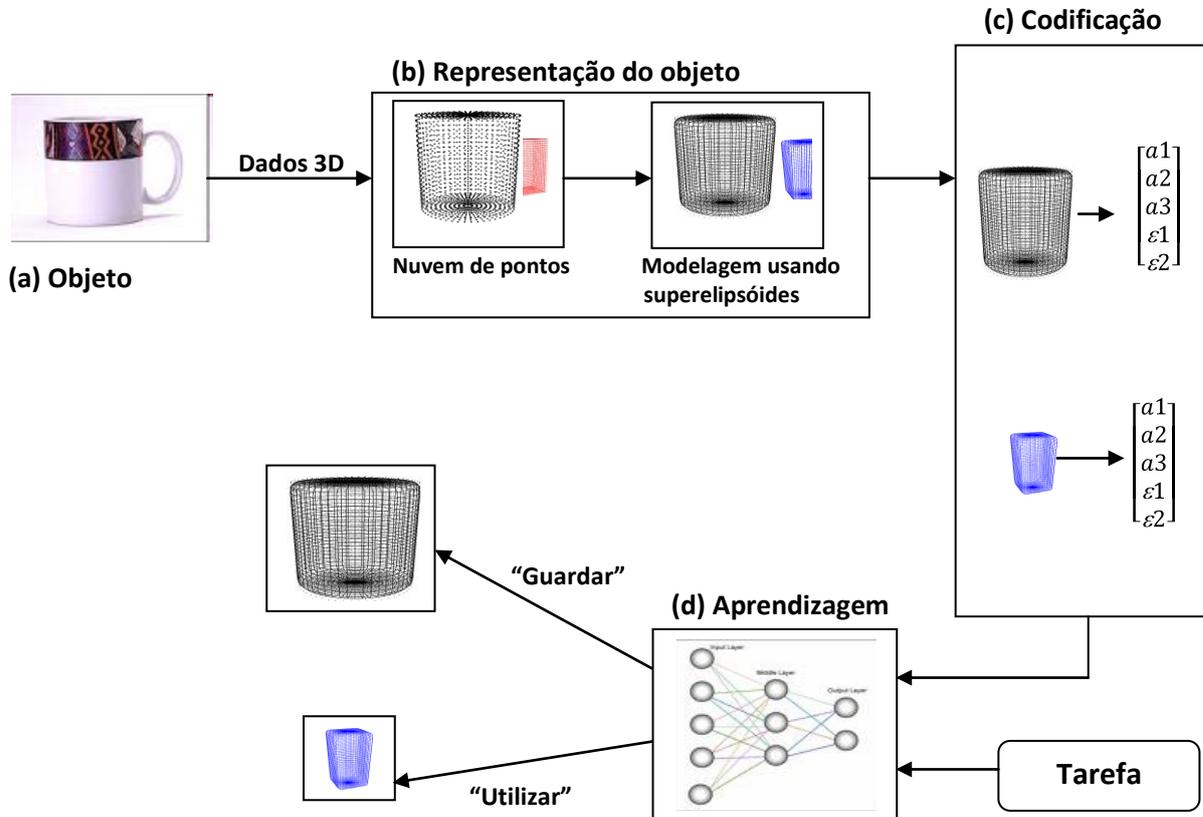


Figura 1.4 – Visão geral da proposta: (a) objeto de interesse; (b) representação do modelo geométrico das partes que compõem o objeto original e modelagem das nuvens de pontos empregando superelipsóides; (c) vetores dos parâmetros das primitivas; (d) escolha da primitiva adequada para atender as tarefas demandas por um agente humano.

A partir de uma descrição por nuvem de pontos das partes que compõem o modelo geométrico do objeto, a modelagem automática desses volumes é obtida pelo uso de superelipsóides.

O vetor que representa os parâmetros do conjunto de superelipsóides é interpretado por um método de aprendizagem supervisionado para sugerir a primitiva consistente com a tarefa. Diante da abrangência de possibilidades, esta proposta se reserva à interpretação convencional de duas ações: "Utilizar" e "Guardar".

O uso de uma caneca para servir a alguém, por exemplo, pode ser obtido com a primeira instrução, enquanto a alteração da posição de uma garrafa ou panela é produzida pela segunda ação. Todavia, a aplicação desses comandos pode ser estendida para os demais objetos da figura 1.3, sem perda de generalidade.

1.3 Objetivo

Com a finalidade de decompor objetos complexos em formas geométricas elementares, um método de modelagem automática de objetos sintéticos 3D baseada em superelipsóides é desenvolvido neste trabalho.

A partir da interpretação dos parâmetros que descrevem o conjunto de tais volumes básicos, uma rede neural artificial capaz de aprender a associar tais primitivas a tarefas demandadas por agentes humanos, encerrará a contribuição acadêmica deste projeto.

1.4 Organização

Uma introdução ao estado da arte das propostas empregadas na modelagem, síntese, reconhecimento e reconstrução de objetos 3D utilizando superelipsóides é conduzida no próximo capítulo.

Posteriormente, o capítulo III apresenta o conceito e aplicações das superelipsóides como modelos volumétricos compactos, controláveis e intuitivos na representação de formas geométricas primitivas. As principais propriedades dessa concepção de modelagem, fundamentais para o desenvolvimento deste trabalho, também serão discutidas. Transformações matriciais homogêneas, matriz de rotação envolvendo ângulos de Euler, cálculo do baricentro e mudança de sistemas de coordenadas, são igualmente apresentados para permitir a simulação de primitivas em poses genéricas no espaço. Além disso, uma breve revisão sobre os procedimentos algébricos propostos para o ajuste de superelipsóide a nuvens de pontos está incluída neste estudo.

Na seqüência, o programa computacional desenvolvido para o processo automático de ajuste de superelipsóides, bem como o método numérico utilizado, compõem o capítulo IV.

Para demonstrar o desempenho e a eficiência do algoritmo de ajuste, o capítulo V apresenta as simulações numéricas de um amplo elenco de primitivas sintéticas organizadas em diferentes configurações.

O capítulo VI apresenta uma breve discussão do estado da arte das principais estratégias de planejamento de pegadas orientadas à tarefa, enquanto o capítulo subsequente desenvolve a contribuição inovadora deste trabalho.

Finalmente, o capítulo VIII encerra esta tese com considerações e discussões sobre os resultados demonstrados.

Introdução ao estado da arte na modelagem e síntese de objetos 3D baseada em superelipsóides

Não é recente a busca pela modelagem geométrica de objetos tridimensionais por meio de computação gráfica e métodos numéricos. Barr (1981), numa introdução de um dos mais importantes artigos sobre o tema, já enfatizava a procura e o interesse acadêmicos num tema que tem ganhado cada vez mais relevância, sempre numa crescente, à mesma medida da evolução tecnológica. Um fragmento de seu texto explicita o então estado da arte em modelagem geométrica de objetos 3D quando da época de sua publicação.

“Over the past 20 years, a great deal of interest has developed in the use of computer graphics and numerical methods for three-dimensional design. Significant progress in geometric modeling is being made, predominantly for objects best represented by lists of edges, faces, and vertices. One long-term goal of this work is a unified mathematical formalism, to form the basis of an interactive and intuitive design environment in which designers can simulate three-dimensional scenes with shading and texture, produce usable design images, verify numerical machining-control commands, and set up finite-element meshwork for structural and dynamic analysis.”

(Barr, 1981)

Com esta introdução, o estudante de doutorado do Rensselaer Polytechnic Institute, Alan H. Barr, estendeu o uso das formas geométricas básicas, tipicamente representadas por superfícies quadráticas bidimensionais, com a introdução de modelos 3D denominados *superquadrics*. Como mostra a figura 2.1, este estudo

revelou um novo espectro de representações volumétricas controladas por um reduzido conjunto de parâmetros.

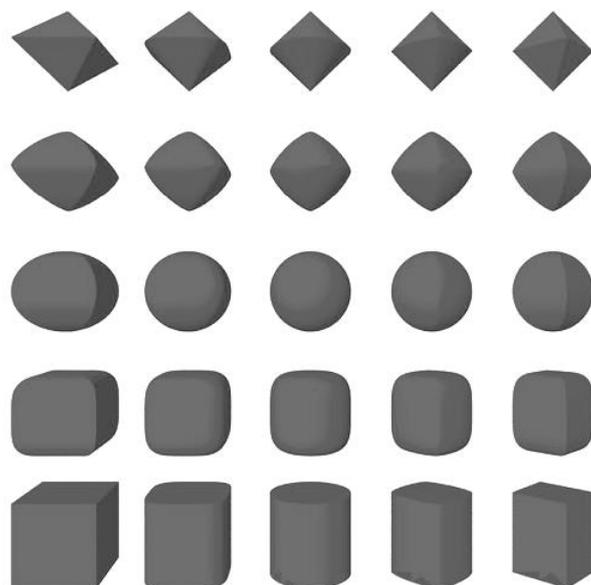


Figura 2.1 – Exemplos de superelipsóides.

Deformações globais descritas por *tapering* e *bending* (Barr, 1984) foram acrescentadas na formulação original de tais superfícies, ampliando a capacidade de representação desses modelos, como mostra a figura 2.2.

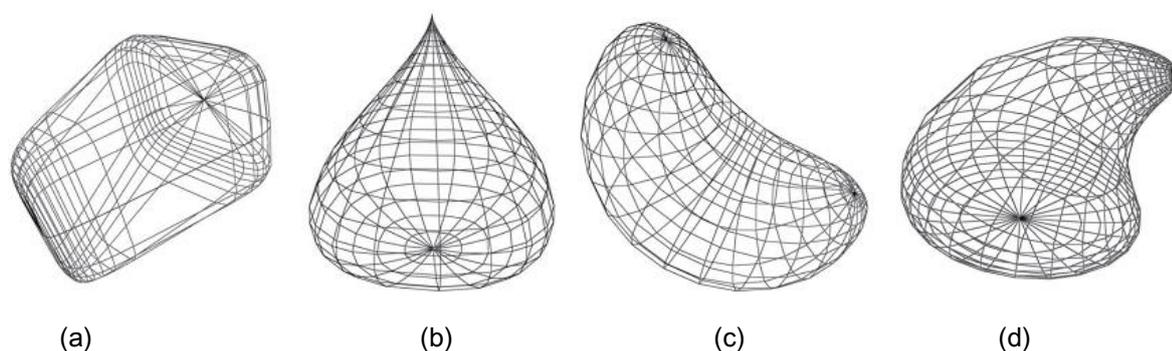


Figura 2.2 – (a) e (b) primitivas cônicas produzidas pela deformação *tapering*; (c) e (d) primitivas dobradas resultantes da deformação *bending*.

Tais estudos foram motivados pela popularidade do uso não acadêmico, na década de 1960, das chamadas superelipses, utilizadas pelo escritor e designer

dinamarquês Piet Hein (Gardner, 1977). Em particular, essas geometrias foram aplicadas nos projetos de pavimentação e centros comerciais da Avenida Sergels Torg em Estocolmo – Suécia, como ilustra a figura 2.3.

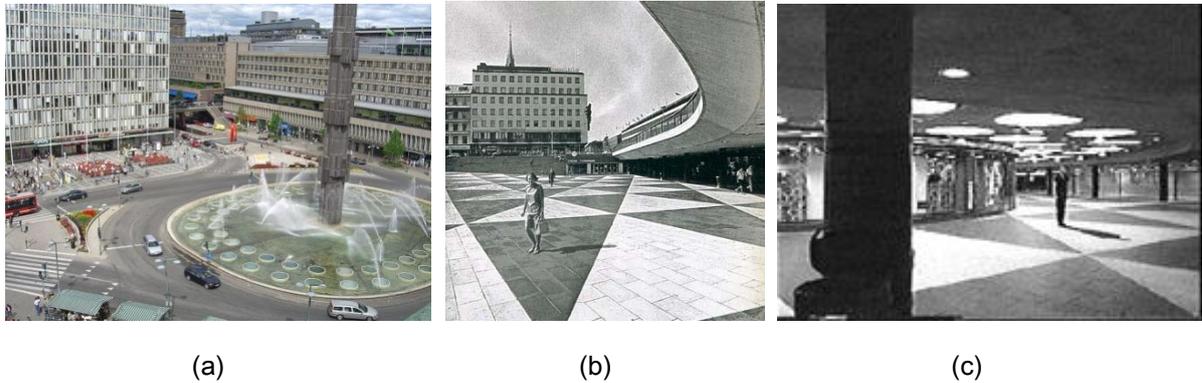


Figura 2.3 – (a) Fonte de água no formato de superelipse em frente ao edifício Hötorget; (b) Praça em desnível; (c) interior de um centro de compras na Avenida Sergels Torg (Estocolmo - Suécia).

No entanto, o termo *superquadrics* ganhou popularidade nas aplicações envolvendo visão computacional com os trabalhos de Pentland (1986), Solina e Bajcsy (1990), Leonardis *et al.* (1997) e Jaklic *et al.* (2000). Mas, o método de Solina e Bajcsy (1990) praticamente se tornou o procedimento padrão para ajustar *superquadrics* individuais a conjuntos de pontos tridimensionais.

Gupta *et al.* (1989) demonstrou argumentos qualitativos e quantitativos sobre os critérios necessários para a análise experimental da qualidade das *superquadrics* ajustadas às imagens de objetos reais. Todavia, os métodos de Pentland (1991), Solina e Bajcsy (1990) e Terzopoulos e Metaxas (1991) são adequados para recuperar formas volumétricas isoladas, não sendo capazes de modelar objetos mais elaborados.

Embora as técnicas de Pentland (1987), Pentland (1990), Ferrie *et al.* (1993) e Gupta e Bajcsy (1993) segmentem e modelem volumes complexos, elas envolvem vários procedimentos, alguns deles aplicados de modo hierárquico, abrangendo desde a estimativa de propriedades de superfícies, curvaturas, entre outros, até estágios mais difíceis, como aqueles definidos pela busca por simetria rotacional, com o propósito de segmentar os dados em partes que podem, supostamente, serem representadas por um único modelo volumétrico (Leonardis *et al.*, 1997).

De fato, essas propostas isolam o estágio de segmentação daquele da representação volumétrica e significativos esforços são necessários para combiná-los em um único modelo. A habilidade para identificar um conjunto de superfícies pertencentes a um volume comum se mostra uma tarefa não trivial sem o conhecimento, pelo menos, da conectividade dessas superfícies. Além disso, as descrições dessas superfícies podem não ser consistentes com o volume original (Leonardis *et al.*, 1997).

Motivados pelas dificuldades inerentes a tais métodos, o paradigma *recover-and-select*, de Leonardis *et al.* (1997), combinou os procedimentos de segmentação e recuperação em um método iterativo de crescimento de modelos intermediários capazes de modelar nuvem de pontos que melhor descreve o volume completo do objeto. Como resultado, tal técnica evitava qualquer estágio preliminar de pré-segmentação e, desse modo, caracterizou um contraste para os procedimentos tradicionais. A seqüência mostrada na figura 2.4 expõe os resultados experimentais para objetos reais.

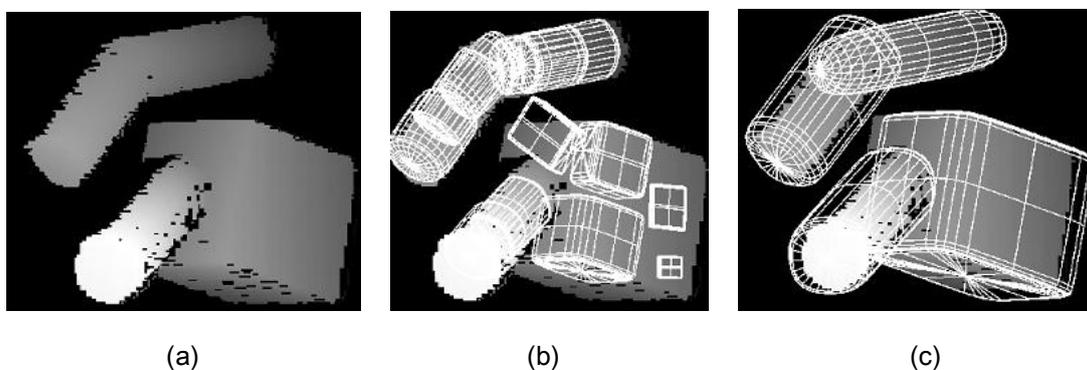


Figura 2.4 – (a) nuvem de pontos do objeto; (b) estágio intermediário do processo *recovery-and-select*; (c) modelo final recuperado por quatro superelipsóides. (Leonardis *et al.*, 1997)

Embora as técnicas mencionadas acima e aquelas propostas por Barr (1981, 1984), Boulton e Gross (1987), Ferrie e White (1993), Gupta (1992) e Solina (1990) modelem um amplo elenco de objetos, elas não são adequadas para representar volumes não simétricos em torno dos eixos coordenados. Para esses casos, as superfícies paramétricas deformáveis para o modelo proposto por Bardinnet *et al.*

(1995) descreveram uma aplicação na representação de imagens médicas com a modelagem do ventrículo esquerdo de um coração humano.

Inspirados nestes resultados, a abordagem das superelipsóides estendidas de Zhou e Kambhamettu (2000) ampliou a capacidade de representação desses modelos. Especificamente, tal proposta substituiu os expoentes constantes da notação original que controlavam a forma dos modelos por funções dependentes da latitude e longitude do objeto quando centrado na origem de um sistema de coordenadas esférico. Como consequência, foi possível modificar superelipsóides em direções genéricas, permitindo a modelagem de uma variedade muito maior de formas geométricas quando comparada àquelas tipicamente obtidas para os modelos padrões. Além disso, as desejáveis propriedades caracterizadas por um modelo compacto, controlável e com significado intuitivo foram preservadas, as quais descrevem aspectos atrativos na reconstrução, reconhecimento e modelagem de formas. A habilidade dessa forma de representação para sintetizar objetos assimétricos de uso cotidiano é ilustrada na figura 2.5.

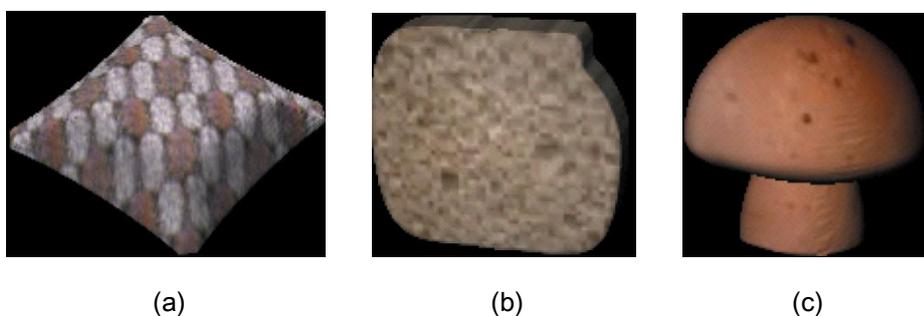


Figura 2.5 – Síntese de objetos reais empregando superelipsóides estendidas: (a) traveseiro; (b) fatia de pão; (c) cogumelo. (Zhou e Kambhamettu, 2000)

Em contrapartida, um método de codificação geométrico proposto por Chevalier *et al.* (2001) demonstrou aplicações na visualização e transmissão de imagens de objetos reais e virtuais com o emprego de uma técnica de *constructive solid geometry*. Ainda com o foco da aplicação direcionado à compreensão de dados para transmissão e armazenamento, Chevalier *et al.* (2003) apresentaram uma nova abordagem com o emprego de dois métodos de segmentação e modelagem capazes de representar pontos tridimensionais não organizados.

Como as funções objetivo empregadas para indicar a distância da nuvem de pontos às superelipsóides descrevem problemas típicos de mínimos quadrados, uma comparação experimental das abordagens de Solina e Bajcsy (1990) e Gross e Boulton (1988), freqüentemente propostas para tais critérios de ajustes, foi conduzida por Yan Zhang (2003). Neste estudo, o foco de tais comparações foi direcionado a objetos sintéticos e reais quanto à exatidão dos parâmetros estimados, erros do ajuste, robustez contra ruídos, sensibilidade às faces não visíveis de objetos reais, bem como a velocidade de convergência do método numérico.

Baseado na análise da curvatura de superfícies, a estratégia de representação de Yan Zhang (2003) modelou peças automotivas com superelipsóides. Posteriormente, na aplicação de um procedimento iterativo de segmentação e recuperação, Bhabhrawala e Krovi (2005) conduziram discussões sobre o uso de superelipsóides estendidas na biomedicina, cujas irregularidades das formas dos itens impunham dificuldades para os modelos de representação tradicionais. A modelagem de formas geométricas com concavidades pronunciadas é ilustrada na figura 2.6.

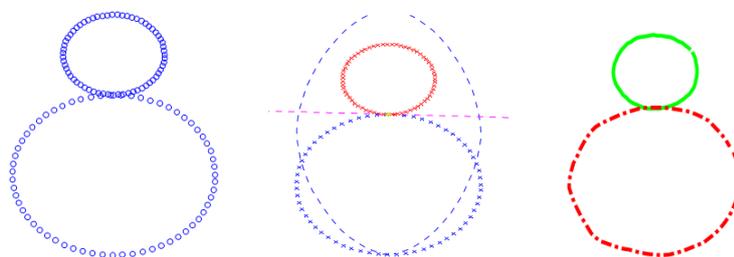


Figura 2.6 – Modelagem de um boneco de neve: (a) nuvem de pontos do objeto; (b) Segmentação dos dados; (c) aproximação do objeto por duas superelipsóides. (Bhabhrawala e Krovi, 2005)

A partir da reconstrução de uma imagem genérica por um conjunto de superelipsóides, o método de Krivic (2006) permitiu reconhecer objetos com juntas articuladas, os quais, supostamente, compõem o cenário. Os objetos selecionados da figura 2.7(a), modelados manualmente (ver figura 2.7(b)), formam a entrada para o sistema proposto reconhecer esses itens no cenário da figura 2.7(c) também reconstruído por superelipsóides (ver figura 2.7(d)).

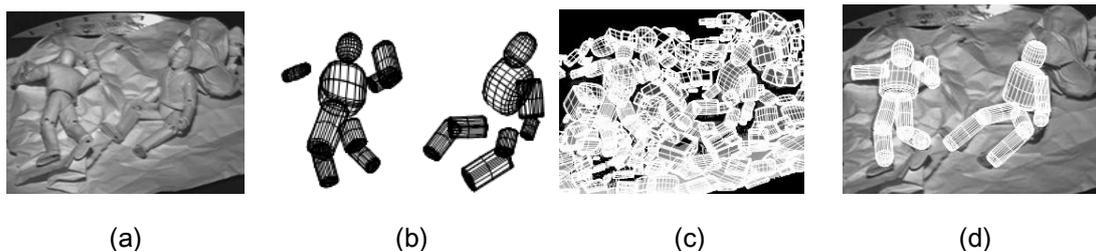


Figura 2.7 – Reconhecimento de objetos: (a) cenário original; (b) modelagem dos objetos selecionados; (c) cenário reconstruído com superelipsóides; (d) objetos reconhecidos. Krivic (2006)

Os esforços de Bhabhrawala *et al.* (2007) resultaram no desenvolvimento de uma classe de modelos híbridos, na qual a estrutura de dados geométricos e computacionais combinou as representações de superfícies implícitas, explícitas e parametrizadas com representações volumétricas de sólidos adequadas para o processo de transição da análise, ao mesmo tempo em que possuía uma parametrização de ordem reduzida. Em particular, aquele trabalho propôs uma nova configuração de parametrização para objetos ligeiramente deformados, tais como aqueles descritos em imagens biomédicas, nos quais a diversidade e irregularidade apresentadas expõem uma dificuldade adicional para representá-los em função de características fixas ou modelos de alta fidelidade capazes de serem submetidos a análises. Tal estratégia é capaz de representar imagens médicas 3D descritas por nuvens de pontos obtidas com o uso de dispositivos de captura de imagens, como ilustra a figura 2.8.

Atualmente esses conhecimentos são usados para a decomposição de uma cena em um conjunto de objetos mais simples. No entanto, a simulação de uma cena como um conjunto de primitivas é mais difícil do que uma representação genérica, tal como aquela descrita por triangulação, mas apresenta algumas vantagens. No caso particular de um robô industrial, por exemplo, é muito mais simples determinar como pegar um objeto representado por um grupo de formas básicas. Além disso, a armazenagem de uma cena composta de primitivas na forma binária é obviamente o modo mais eficiente quando se considera o esforço computacional demandado (Strand e Dillmann, 2009).

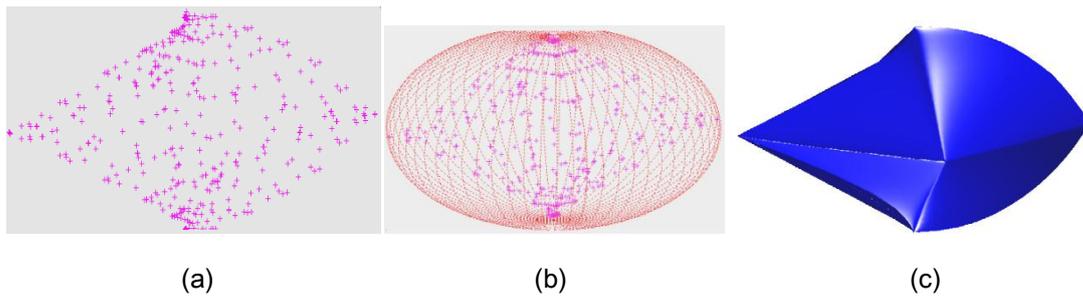


Figura 2.8 – Processo de ajuste utilizando superelipsóides estendidas: (a) nuvem de pontos do objeto; (b) ajuste inicial por uma superfície simétrica; (c) modelo representativo do objeto. (Bhabhrawala *et al.* 2007)

A técnica de (Strand e Dillmann, 2009) emprega a abordagem clássica de modelagem de objetos complexos, ou seja, uma etapa de segmentação manual isola os pontos que representam o objeto de interesse, enquanto a escolha da melhor superelipsóides que simula grupos de pontos é obtida pela minimização de uma função distância. A cena da figura 2.9(a) mostra um ser humano em uma sala, enquanto a figura 2.9(b) ilustra este homem isolado da cena principal e a figura 2.9(c) exibe a aproximação utilizando superelipsóides.

A aquisição dos dados a partir da cena real empregou um robô móvel de exploração autônomo equipado com um scanner laser rotativo capaz de gerar, aproximadamente, 180000 coordenadas para cada nuvem de pontos. Todavia, os autores reconhecem as deficiências do método na ambigüidade das descrições das formas.

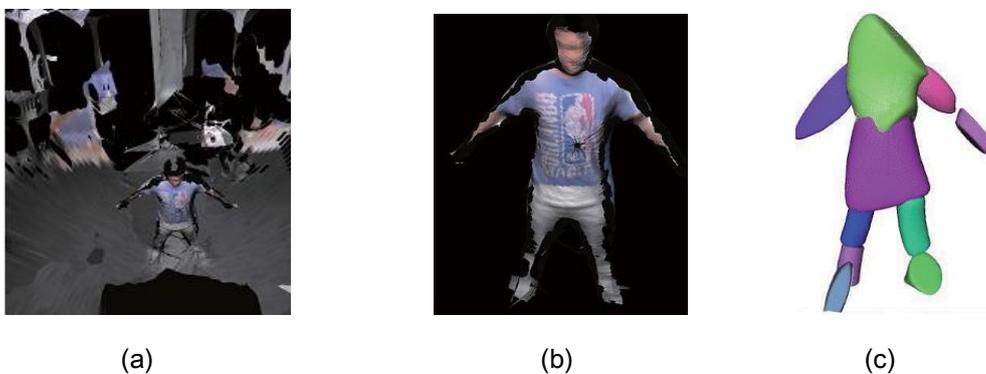


Figura 2.9 – (a) Escolha manual de um ser humano a partir de uma cena principal. (b) homem segmentado; (c) representação composta por superelipsóides. (Strand e Dillmann, 2009)

O capítulo subsequente introduz os principais conceitos e propriedades geométricas do uso das superelipsóides.

Parte I

Modelagem automática de nuvem de pontos empregando superelipsóides

Capítulo III

Representação de objetos por superelipsóides

Os conceitos envolvidos na síntese de superelipsóides dedicadas à representação de nuvens de pontos tridimensionais são estudados no presente capítulo.

As propriedades geométricas desses modelos volumétricos, bem como os procedimentos envolvidos para ajustar os parâmetros de superelipsóides sem deformações que melhor modelam determinadas nuvens de pontos em uma pose genérica no espaço também serão discutidos. Como este processo define um problema típico de ajuste por mínimos quadrados, este trabalho fez uso de um método numérico de minimização baseado em uma técnica de regressão não-linear.

3.1 Introdução às *Superquadrics*

O termo *superquadrics* foi inicialmente apresentado por Barr (1981) para definir uma família de formas geométricas a qual pode ser interpretada como uma generalização de superfícies quadráticas básicas, tais como aquelas descritas por elipsóides. Essas formas têm sido consideradas primitivas volumétricas na representação de modelos geométricos em computação gráfica (Barr, 1981), (Barr, 1984), (Solina e Bajcsy, 1990) e visão computacional (Pentland, 1987).

Tais superfícies são definidas por intermédio das suas entidades análogas bidimensionais descritas pela notação explícita ou paramétrica das superelipses da equação (3.1).

$$S(\eta) = \begin{cases} x(\eta) = a_1 \cdot \cos^\varepsilon(\eta) \\ y(\eta) = a_2 \cdot \sen^\varepsilon(\eta) \end{cases}, \quad -\pi \leq \eta \leq \pi \quad (3.1)$$

Nesta expressão, a_1 , a_2 e ε são os parâmetros da superelipse. Especificamente, a_1 e a_2 expressam o tamanho dessas formas geométricas ao longo das direções dos eixos coordenados x e y , respectivamente, enquanto o termo ε determina a forma da superfície resultante. Esta expressão é empregada para representar as coordenadas dos pontos pertencentes à superfície da superelipse por meio da atribuição uniformemente distribuída para o coeficiente η no intervalo de $-\pi$ até π . A Figura 3.1 ilustra os pontos sobre as superfícies de quatro superelipses obtidas pelo uso da equação (3.1) para $0,1 \leq \varepsilon \leq 2,0$.

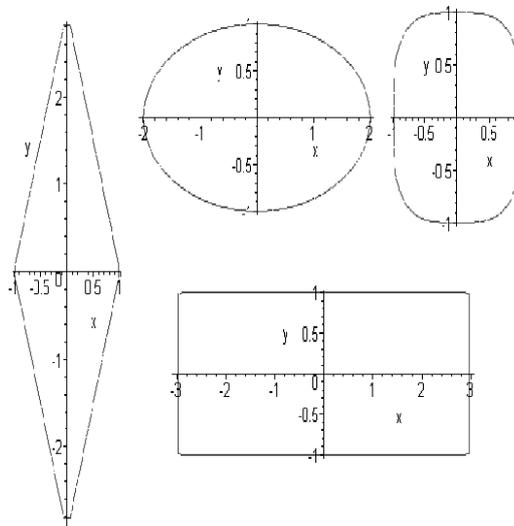


Figura 3.1 – Exemplos de superelipses. (Chevalier, 2004)

A expressão implícita para essas superelipses pode ser derivada a partir da equação (3.1) isolando as funções trigonométricas e elevando os termos resultantes ao quadrado. A consequência desta operação é mostrada nas equações (3.2) e (3.3).

$$\left(\frac{x(\eta)}{a_1}\right)^2 = \cos^{2.\varepsilon}(\eta) \quad (3.2)$$

$$\left(\frac{y(\eta)}{a_2}\right)^2 = \sen^{2.\varepsilon}(\eta) \quad (3.3)$$

Por intermédio do expoente $1/\varepsilon$ aplicado a ambos os termos dessas expressões obtém-se as equações (3.4) e (3.5).

$$\left(\frac{x(\eta)}{a_1}\right)^{\frac{2}{\varepsilon}} = \cos^2(\eta) \quad (3.4)$$

$$\left(\frac{y(\eta)}{a_2}\right)^{\frac{2}{\varepsilon}} = \text{sen}^2(\eta) \quad (3.5)$$

Finalmente, a soma dessas expressões e o emprego da identidade trigonométrica $\cos^2(\eta) + \text{sen}^2(\eta) = 1$ resultam na função implícita da equação (3.6).

$$\left(\frac{x}{a_1}\right)^{\frac{2}{\varepsilon}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon}} = 1 \quad (3.6)$$

Além disso, uma extensão do uso da equação (3.1) pode ser aproveitada para descrever a função paramétrica de uma superfície tridimensional, como foi apresentado por Barr (1981). No entanto, para obter a expressão que representa essa superfície ele introduziu a definição do produto esférico de duas curvas bidimensionais. Especificamente, o modelo de uma superelipsóide $S(\eta, \mu)$ é obtido pelo produto das duas superelipses, $S_1(\eta)$ e $S_2(\mu)$, das equações (3.7) e (3.8), respectivamente.

$$S_1 = \begin{bmatrix} \cos^{\varepsilon_1}(\eta) \\ a_3 \cdot \text{sen}^{\varepsilon_1}(\eta) \end{bmatrix}, \quad -\pi/2 \leq \eta \leq \pi/2 \quad (3.7)$$

$$S_2 = \begin{bmatrix} a_1 \cdot \cos^{\varepsilon_2}(\mu) \\ a_2 \cdot \text{sen}^{\varepsilon_2}(\mu) \end{bmatrix}, \quad -\pi \leq \mu \leq \pi \quad (3.8)$$

O resultado é mostrado na equação (3.9) na qual o símbolo \otimes representa o produto esférico.

$$S(\eta, \mu) = S_1 \otimes S_2 = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_1 \cdot \cos^{\varepsilon_1}(\eta) \cdot \cos^{\varepsilon_2}(\mu) \\ a_2 \cdot \cos^{\varepsilon_1}(\eta) \cdot \text{sen}^{\varepsilon_2}(\mu) \\ a_3 \cdot \text{sen}^{\varepsilon_1}(\eta) \end{bmatrix}, \quad \begin{matrix} -\pi/2 \leq \eta \leq \pi/2 \\ -\pi \leq \mu < \pi \end{matrix} \quad (3.9)$$

Novamente, os parâmetros a_1 , a_2 e a_3 são os fatores de escala ao longo dos eixos coordenados x , y e z , respectivamente. Em contrapartida, o termo ε_1 controla a forma da superelipsóide ao longo do eixo z , enquanto ε_2 define a forma desta superfície ao longo do plano (x, y) .

A figura 3.2 (a) representa uma interpretação geométrica para o caso particular de uma esfera unitária resultante do produto esférico do círculo contido no plano (x, y) pelo semicírculo localizado no plano ortogonal. Essas curvas correspondem àquelas descritas pelas equações (3.7) e (3.8) para $a_1 = a_2 = a_3 = 1,0$ e $\varepsilon_1 = \varepsilon_2 = 1,0$, enquanto a figura 3.2(b) relaciona os parâmetros ε_1 e ε_2 ao controle da forma de um cilindro obtido quando $a_1 = a_2 < a_3$.

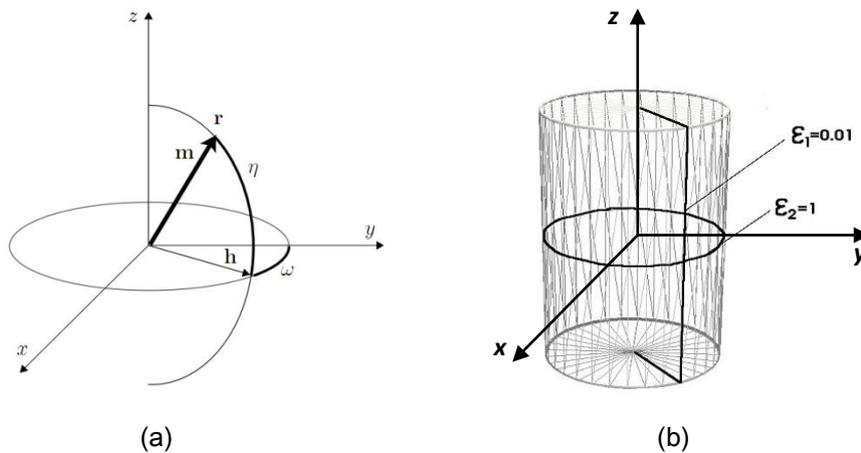


Figura 3.2 – (a) Esfera unitária obtida pelo produto esférico entre um círculo contido no plano (x, y) e um semicírculo projetado num plano ortogonal; (b) cilindro resultante do produto esférico entre um círculo e um retângulo.

Genericamente, essas superfícies caracterizam um modelo muito compacto de aproximação de objetos a partir de poucos parâmetros quando comparado à grande capacidade de representação proporcionada por essas geometrias. De fato, a partir de uma análise da equação (3.9), pode-se afirmar que, com apenas cinco parâmetros, é possível obter uma ampla variedade de formas geométricas, como pode ser observado na figura 3.3.

Em particular, as aplicações demonstradas neste trabalho restringiram os valores dos parâmetros de forma ε_1 e ε_2 a máximos unitários, em decorrência das geometrias convexas exibidas na figura 3.3.

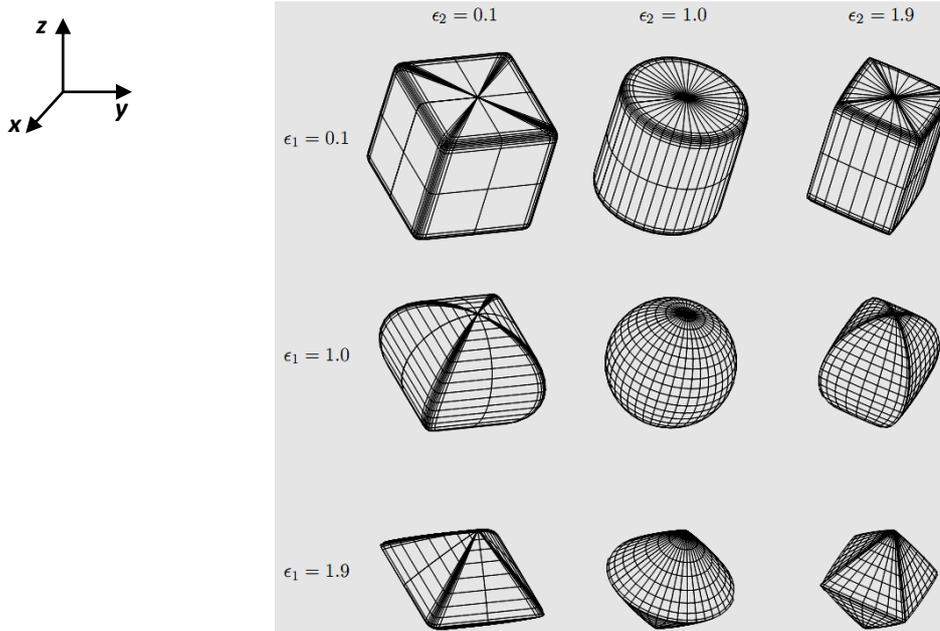


Figura 3.3 - Exemplos de superelipsóides de acordo com a variação dos parâmetros de forma ϵ_1 , ϵ_2 quando $a_1 = a_2 = a_3 = 1,0$: as formas geométricas modelam um cubo para $\epsilon_1 = \epsilon_2 = 0,1$; um cilindro para $\epsilon_1 = 0,1$ e $\epsilon_2 = 1,0$; um paralelepípedo para $\epsilon_1 = 0,1$ e $\epsilon_2 = 1,9$ e uma esfera para $\epsilon_1 = \epsilon_2 = 1,0$, entre outros, como pode ser observado.

Análogo ao procedimento envolvido na obtenção da expressão da superelipse da equação (3.6), a versão implícita das superelipsóides também pode ser derivada a partir da notação paramétrica dessas superfícies.

Isolando as funções trigonométricas da expressão (3.9) as equações (3.10), (3.11) e (3.12) mostram as notações obtidas.

$$\left(\frac{x}{a_1}\right)^2 = \cos^{2 \cdot \epsilon_1}(\eta) \cdot \cos^{2 \cdot \epsilon_2}(\mu), \quad (3.10)$$

$$\left(\frac{y}{a_2}\right)^2 = \cos^{2 \cdot \epsilon_1}(\eta) \cdot \sin^{2 \cdot \epsilon_2}(\mu), \quad (3.11)$$

$$\left(\frac{z}{a_3}\right)^2 = \sin^{2 \cdot \epsilon_1}(\eta) \quad (3.12)$$

Elevando as equações (3.10) e (3.11) ao expoente $1/\epsilon_2$ e, em seguida, somando-as, obtém-se a equação (3.13), na qual foi utilizada a identidade trigonométrica $\cos^2(\eta) + \sin^2(\eta) = 1$.

$$\left(\frac{x}{a_1}\right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon_2}} = \cos^{\frac{2\varepsilon_1}{\varepsilon_2}}(\eta) \quad (3.13)$$

Elevando a equação (3.12) ao expoente $1/\varepsilon_1$ e a equação (3.13) ao expoente $\varepsilon_2/\varepsilon_1$, obtêm-se as expressões (3.14) e (3.15), respectivamente.

$$\left(\frac{z}{a_3}\right)^{\frac{2}{\varepsilon_1}} = \text{sen}^2(\eta) \quad (3.14)$$

$$\left[\left(\frac{x}{a_1}\right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon_2}}\right]^{\frac{\varepsilon_2}{\varepsilon_1}} = \cos^2(\eta) \quad (3.15)$$

Finalmente, a forma implícita de uma superelipsóide, obtida pela soma dessas expressões e uso da identidade trigonométrica, é mostrada na equação (3.16).

$$\left[\left(\frac{x}{a_1}\right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon_2}}\right]^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z}{a_3}\right)^{\frac{2}{\varepsilon_1}} = 1 \quad (3.16)$$

A partir de uma análise desta expressão, pode-se afirmar que todos os pontos com coordenadas (x, y, z) pertencem, por definição, à superfície da superelipsóide. Uma extensão deste resultado permite definir a equação (3.17) como função interior-exterior, ao fato de que ela fornece um teste simples para verificar a posição de um ponto em relação à superelipsóide.

$$F(x, y, z) = \left[\left(\frac{x}{a_1}\right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon_2}}\right]^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z}{a_3}\right)^{\frac{2}{\varepsilon_1}} \quad (3.17)$$

Em particular, se $F(x, y, z) = 1$, o ponto pertence à superfície da superelipsóide, enquanto para $F(x, y, z) < 1$ o ponto é interno à superelipsóide e, finalmente, para $F(x, y, z) > 1$ o ponto é externo à superelipsóide.

A generalização da forma padrão para superfícies tridimensionais foi demonstrada por Barr (1981) para obter as equações implícitas e paramétricas, nesta ordem, das seguintes *superquadrics*:

A. Superhiperbolóide de uma folha:

$$\begin{aligned}
 F(x, y, z) &= \left(\frac{x}{a_1}\right)^2 + \left(\frac{y}{a_2}\right)^2 - \left(\frac{z}{a_3}\right)^2 \\
 S(\eta, \mu) &= \begin{bmatrix} \sec(\eta) \\ a_3 \cdot \tan(\eta) \end{bmatrix} \otimes \begin{bmatrix} a_1 \cdot \cos(\mu) \\ a_2 \cdot \sen(\mu) \end{bmatrix} \\
 &= \begin{bmatrix} a_1 \cdot \sec(\eta) \cdot \cos(\mu) \\ a_2 \cdot \sec(\eta) \cdot \sen(\mu) \\ a_3 \cdot \tan(\eta) \end{bmatrix}, \quad \begin{array}{l} -\pi/2 < \eta < \pi/2 \\ -\pi \leq \mu < \pi \end{array} \quad (3.18)
 \end{aligned}$$

B. Superhiperbolóide de duas folhas:

$$\begin{aligned}
 F(x, y, z) &= \left(\frac{x}{a_1}\right)^2 - \left(\frac{y}{a_2}\right)^2 - \left(\frac{z}{a_3}\right)^2 \\
 S(\eta, \mu) &= \begin{bmatrix} \sec(\eta) \\ a_3 \cdot \tan(\eta) \end{bmatrix} \otimes \begin{bmatrix} a_1 \cdot \sec(\mu) \\ a_2 \cdot \tan(\mu) \end{bmatrix} \\
 &= \begin{bmatrix} a_1 \cdot \sec(\eta) \cdot \sec(\mu) \\ a_2 \cdot \sec(\eta) \cdot \tan(\mu) \\ a_3 \cdot \tan(\eta) \end{bmatrix}, \quad \begin{array}{l} -\pi/2 < \eta < \pi/2 \\ -\pi/2 < \mu < \pi/2 \text{ (folha 1)} \\ \pi/2 < \mu < 3\pi/2 \text{ (folha 2)} \end{array} \quad (3.19)
 \end{aligned}$$

C. Supertoróides:

$$F(x, y, z) = \left\{ \left[\left(\frac{x}{a_1}\right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon_2}} \right]^{\frac{\varepsilon_2}{2}} - a_4 \right\}^{\frac{2}{\varepsilon_1}} + \left(\frac{z}{a_3}\right)^{\frac{2}{\varepsilon_1}} \quad (3.20)$$

$$S(\eta, \mu) = \begin{bmatrix} a_1 \cdot [a_4 + \cos^{\varepsilon_1}(\eta)] \cdot \cos^{\varepsilon_2}(\mu) \\ a_2 \cdot [a_4 + \cos^{\varepsilon_1}(\eta)] \cdot \sen^{\varepsilon_2}(\mu) \\ a_3 \cdot \sen^{\varepsilon_1}(\eta) \end{bmatrix}, \quad \begin{array}{l} -\pi/2 \leq \eta \leq \pi/2 \\ -\pi \leq \mu \leq \pi \end{array} \quad (3.21)$$

A Figura 3.4 (a) mostra a superfície descrita pelas superelipsóides definidas nesta seção, enquanto as imagens (b) e (c) exibem as *superhiperbolóides* de uma e duas folhas, respectivamente, e a imagem (d) ilustra uma *supertoróide*.

Uma vez que as superelipsóides descrevem a única representação que modela superfícies fechadas sem aberturas, ela foi majoritariamente adotada na robótica (Dop e Regtien, 1996; Leonardis, 1997; Chevalier *et al.*, 2001; Chevalier *et al.*, 2003; Pelossof, 2004; Goldfeder, 2007; El-khoury e Sahbani, 2008; El-khoury e Sahbani, 2009; e Strand e Dillmann, 2009) para modelar objetos envolvidos nos processos de manipulação. Desse modo, este trabalho tem seu foco na construção geométrica dessas superfícies, bem como nas suas propriedades, as quais serão estudadas na próxima seção.

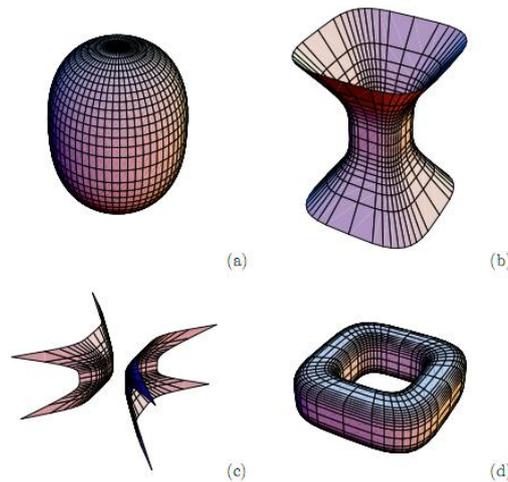


Figura 3.4 – A família de *superquadrics* inclui as formas geométricas de uma superelipsóide (a), superhiperbolóides de uma folha (b) e duas folhas (c) e *supertoróides* (d). (Barr, 1981)

Todavia, independente da notação que descreve as superelipsóides, as equações (3.9) e (3.17) envolvem a mesma quantidade de parâmetros e, de acordo com a exigência da aplicação, cada uma delas, a seu tempo, foi utilizada no trabalho. Em particular, o cálculo da distância de um ponto à superfície da superelipsóide fez uso da função implícita, enquanto o processo de exibição visual dessas superfícies empregou a expressão da forma paramétrica. Uma generalização do uso de tais expressões para a representação de uma nuvem de pontos localizada em uma posição genérica no espaço é estudada na próxima seção.

3.2 Posições genéricas no espaço tridimensional

As equações da seção anterior são válidas somente para representar superelipsóides localizadas na origem do sistema de coordenadas local orientadas na direção dos eixos coordenados x , y e z . Como os pontos do conjunto de dados podem estar contidos no sistema de coordenadas global, estas notações não são adequadas para descrever objetos em posições genéricas no espaço, como aquela ilustrada na Figura 3.5.

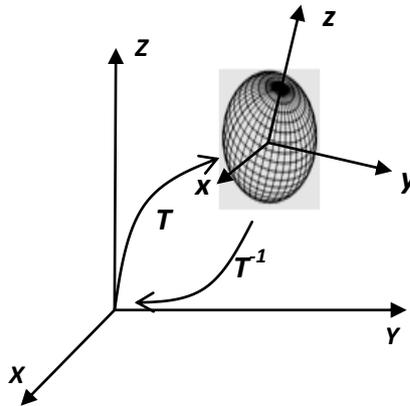


Figura 3.5 - Representação de uma superelipsóide em uma posição qualquer no espaço. Os eixos do sistema de coordenadas local são denotados por x , y e z , enquanto os correspondentes no sistema de coordenadas global são caracterizados por X , Y e Z . A matriz de transformação T descreve a mudança do sistema de coordenadas local para o sistema global. Em contrapartida, a matriz inversa T^{-1} promove a transformação do sistema de coordenadas global para o sistema de coordenadas local.

Com o objetivo de adequar essas expressões para superelipsóides localizadas em uma posição genérica no espaço são necessárias uma matriz de rotação R e um vetor de translação \bar{P} , aplicados às coordenadas originais dos pontos do modelo. A notação empregada para essa transformação seguiu aquela apresentada na equação (3.22),

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & p_x \\ \cdot & R & \cdot & p_y \\ \cdot & \cdot & \cdot & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad (3.22)$$

na qual (X, Y, Z) e (x, y, z) são as coordenadas dos pontos nos sistemas de coordenadas global e local, respectivamente, enquanto as coordenadas (p_x, p_y, p_z) representam o vetor de translação do baricentro da nuvem de pontos à origem do sistema de coordenadas global.

Em particular, neste trabalho adotou-se a definição da equação (3.23) para a matriz de rotação R com seus elementos expressos em função dos ângulos de Euler ϕ , θ e ψ .

$$R = \begin{pmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \cdot \begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.23)$$

Como conseqüência, esta matriz apresenta uma importante propriedade revelada pelo fato de ela ser ortogonal, ou seja, $R^{-1} = R^T$. Pode-se observar que a seqüência de operações adotadas é descrita por uma rotação de ϕ graus em relação ao eixo- z , seguida por uma rotação em relação ao eixo- y por um ângulo θ e, finalmente, outra rotação em relação ao eixo- z por um ângulo ψ . Todavia, outras combinações podem ser utilizadas, tais como xyz , yxy , entre outras, tendo em vista que o produto final independe da ordem dessas operações.

Em resumo, a representação das coordenadas de um ponto do sistema de coordenadas local para o sistema global sugere que as coordenadas desse ponto devem ser submetidas a uma rotação seguida por um deslocamento definido pelo vetor de translação, como exibido na equação (3.22).

No entanto, como as superelipsóides são definidas no sistema de coordenadas local, é necessário aplicar uma transformação complementar dada pela equação (3.24).

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & 1 \\ \cdot & R^{-1} & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} (X - p_x) \\ (Y - p_y) \\ (Z - p_z) \\ 1 \end{bmatrix} \quad (3.24)$$

A inversa da equação (3.23) é exibida na equação (3.25).

$$R^{-1} = \begin{bmatrix} [\cos(\phi). \cos(\theta). \cos(\psi) - \text{sen}(\phi). \text{sen}(\psi)] \\ [-\cos(\phi). \cos(\theta). \text{sen}(\psi) - \text{sen}(\phi). \cos(\psi)] \\ [\cos(\phi). \text{sen}(\theta)] \end{bmatrix}$$

$$\begin{bmatrix} [\text{sen}(\phi). \cos(\theta). \cos(\psi) + \cos(\phi). \text{sen}(\phi)] & - [\text{sen}(\phi). \cos(\psi)] \\ [-\text{sen}(\phi). \cos(\theta). \text{sen}(\psi) + \cos(\phi). \cos(\phi)] & [\text{sen}(\theta). \text{sen}(\psi)] \\ [\text{sen}(\phi). \text{sen}(\theta)] & [\cos(\theta)] \end{bmatrix}$$

(3.25)

Como resultado, as coordenadas globais (X, Y, Z) de um ponto são transformadas nas correspondentes x, y e z do sistema de coordenadas da superelipsóide, como mostram as equações (3.26), (3.27) e (3.28).

$$x = [\cos(\phi). \cos(\theta). \cos(\psi) - \text{sen}(\phi). \text{sen}(\psi)]. (X - p_x) + [\text{sen}(\phi). \cos(\theta). \cos(\psi) + \cos(\phi). \text{sen}(\phi). Y - p_y - \text{sen}(\phi). \cos(\psi). Z - p_z]$$

(3.26)

$$y = [-\cos(\phi). \cos(\theta). \text{sen}(\psi) - \text{sen}(\phi). \cos(\psi)]. (X - p_x) + [-\text{sen}(\phi). \cos(\theta). \text{sen}(\psi) + \cos(\phi). \cos(\phi)]. (Y - p_y) + [\text{sen}(\theta). \text{sen}(\psi)]. (Z - p_z)$$

(3.27)

$$z = [\cos(\phi). \text{sen}(\theta)]. (X - p_x) + [\text{sen}(\phi). \text{sen}(\theta)]. (Y - p_y) + \cos(\theta). (Z - p_z)$$

(3.28)

Finalmente, a combinação desses resultados na função implícita da equação (3.17) permite descrever superelipsóides em poses genéricas no espaço como mostra a equação (3.29)¹. Por conveniência, seus termos serão representados pelos onze parâmetros do vetor A .

$$A = (a_1, a_2, a_3, \varepsilon_1, \varepsilon_2, \phi, \theta, \psi, p_x, p_y, p_z),$$

(3.30)

na qual os termos representam, nesta ordem, os parâmetros de tamanho; os fatores de forma; a orientação; e as coordenadas do vetor de translação.

3.3 Ambigüidade das superelipsóides na modelagem de formas geométricas

Além da simetria em relação à rotação em torno dos eixos coordenados, a combinação das funções trigonométricas da expressão (3.29) com os parâmetros de forma e tamanho podem produzir geometrias idênticas para diferentes valores desses termos. Um simples exemplo pode ser verificado pelo fato de um paralelepípedo com bordas ligeiramente arredondadas ($a_1 = a_2 = 1$, $\varepsilon_1 = \varepsilon_2 = 0,1$) poder ser representado, após uma rotação de $\pi/4$ em torno do eixo z , pelo objeto obtido com os parâmetros $\varepsilon_1 = 0,1$, $\varepsilon_2 = 1,9$ e $a_1 = a_2 = \sqrt{2}$ (Solina e Bajcsy, 1990).

Contudo, o critério adotado neste trabalho para evitar tais casos de ambigüidade consistiu na restrição dos valores desses parâmetros durante o processo de ajuste para forçar a convergência do método de minimização a mínimos globais. O procedimento utilizado é descrito no final da próxima seção.

3.4 Definição e interpretação geométrica da função objetivo

A superfície que melhor descreve um conjunto de pontos corresponde ao ajuste dos parâmetros da superelipsóide expressos na equação (3.30). Para determinar os valores dos termos que minimizam a distância desta superfície àqueles pontos é necessário definir uma função objetivo para o processo de minimização, a qual atuará como um indicador da qualidade do modelo resultante.

Como consequência, este ajuste e o desempenho do algoritmo de minimização dependem da escolha adequada desta função.

Todavia, devido à ampla variedade de formas geométricas, não existe um conjunto de parâmetros $(a_1, a_2, a_3, \varepsilon_1, \varepsilon_2, \theta, \phi, \psi, p_x, p_y, p_z)$ para o qual o modelo se ajuste perfeitamente a um determinado conjunto de pontos. Além do mais, uma vez que o cálculo da distância euclidiana dos pontos à superfície da superelipsóide é computacionalmente oneroso, ver Zhang (2003) e Chevalier (2003) para detalhes, uma breve apresentação de abordagens alternativas se mostra necessária.

Para a conveniência das discussões apresentadas nesta seção, a figura 3.6 ilustra uma comparação entre a distância Euclidiana, $\|PM\|$, e a base das aproximações numéricas que serão descritas, $\|PP_o\|$, baseada na função implícita da superelipsóide.

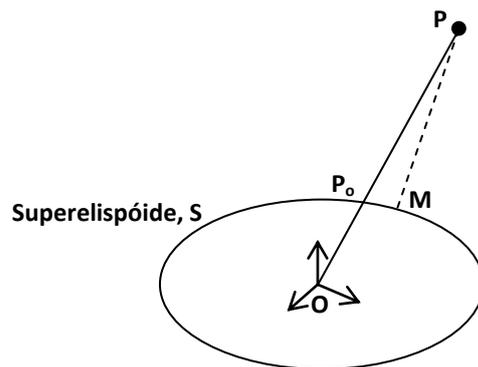


Figura 3.6 – Comparação entre a distância euclidiana $\|PM\|$ de um ponto à superfície da superelipsóide e a distância euclidiana radial $\|PP_o\|$ considerada como aproximação do cálculo da distância exata. O ponto M corresponde à projeção ortogonal do ponto P, enquanto o ponto P_o descreve a intersecção entre a reta definida pela origem O e o ponto P à superfície da superelipsóide S.

Considerando uma nuvem de N pontos posicionados no sistema de coordenadas global, o interesse consiste em variar os onze parâmetros da equação (3.30) para obter a menor distância entre esses pontos e a superfície da superelipsóide candidata ao resultado esperado. A partir deste enunciado o processo de ajuste corresponde à minimização da soma do quadrado da distância dos pontos ao modelo, como descreve a equação (3.31).

$$d_0(\mathcal{A}) = \sum_{i=1}^N [F(\mathcal{A}; X, Y, Z) - 1]^2 \quad (3.31)$$

Entretanto, Solina e Bajcsy (1990) e Jaklic (2000) observaram que esta abordagem direta apresenta duas preocupantes desvantagens. A primeira delas se concentra no fato da minimização desta equação favorecer modelos de grandes volumes, problema que ocorre nos casos particulares de informações insuficientes para estimar o tamanho do modelo, sobretudo quando nem todos os lados do objeto estão visíveis ao mesmo tempo. O segundo aspecto se refere ao fato desta expressão favorecer modelos de elevada curvatura descritas predominantemente por esferas.

Uma tentativa para resolver o primeiro problema foi proposta por Solina e Bajcsy (1990) e Jaklic (2000), a qual consistiu na multiplicação da função implícita da equação (3.17) por $\sqrt{a_1 \cdot a_2 \cdot a_3}$. Esse resultado tornou a busca pela mínima distância proporcional aos modelos de menores volumes. Em contrapartida, o segundo problema foi solucionado elevando a função implícita ao expoente ε_1 . Isto tornou a função objetivo independente da forma da superfície controlada por este termo (Katsoulas, 2004), além de ter combatido o rápido crescimento numérico da função para pequenos valores de ε_1 . Combinando essas modificações, a modelagem de um conjunto de pontos por uma superelipsóide pode ser obtida pela minimização da equação (3.32).

$$d_1(\mathcal{A}) = \sum_{i=1}^N \left\{ \sqrt{a_1 \cdot a_2 \cdot a_3} \cdot [F^{\varepsilon_1}(\mathcal{A}; X, Y, Z) - 1] \right\}^2 \quad (3.32)$$

A principal vantagem no uso desta proposta é revelada pela eficiência computacional do processo de ajuste, uma vez que ela descreve uma função analítica para um método de minimização. Todavia, desconsiderando a melhora na robustez numérica proporcionada pela nova notação, os problemas relacionados à utilização desta expressão ainda não foram totalmente eliminados (Katsoulas, 2004).

Uma análise desta expressão permite afirmar que os resultados obtidos podem divergir daqueles esperados quando dois modelos em escalas diferentes são comparados, sobretudo para objetos alongados (Chevalier *et al.*, 2003).

De fato, Jaklic (2000) observou que esta representação não produz valores uniformemente distribuídos para a distância em todas as direções do modelo, bem

como dependência para com os parâmetros de tamanho. Em particular, na direção da maior dimensão de um modelo o aumento numérico da função se torna mais lento. Além disso, a minimização do escalar $(F^{\varepsilon_1} - 1)$ ainda favorece modelos de maiores volumes, como foi inicialmente demonstrado por Whaite e Ferrie (1991) e Katsoulas (2004). Tal comportamento pode ser verificado pela análise da equação (3.33), proposta por Whaite e Ferrie (1991).

$$F^{\varepsilon_1}(\Lambda; X, Y, Z) - 1 = \frac{d}{|r_s|} \cdot \left(\frac{d}{|r_s|} + 2 \right). \quad (3.33)$$

Nesta equação, dado o ponto (X, Y, Z) e considerando o vetor OP que conecta o centro de gravidade do modelo ao ponto (ver figura 3.7) o ponto P_o representa a intersecção deste vetor à superfície do modelo e $r_s = OP_o$. Uma vez que $|r_s|$ aumenta para volumes crescentes, a função (3.32) ainda favorece modelos de maiores volumes (Katsoulas, 2004).

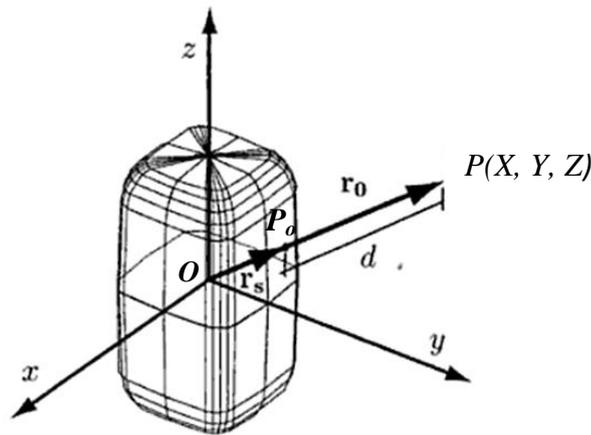


Figura 3.7 – Interpretação geométrica da distância euclidiana radial. O vetor r_o representa a separação entre a origem da superelipsóide e o ponto (X, Y, Z) , enquanto o vetor r_s indica a distância da origem à superfície mencionada. O módulo da diferença desses dois vetores caracteriza a distância euclidiana radial, d .

Para melhorar esta aproximação, especialmente quando a estimativa difere muito do valor exato (Chevalier *et al.*, 2003) a abordagem denominada “distância euclidiana radial”, apresentada por Whaite e Ferrie (1991), descreve a terceira proposta discutida nesta seção. Ela é definida como a distância de um ponto à

superfície da superelipsóide ao longo da linha que conecta este ponto à origem do sistema de coordenadas do modelo, como sugere a interpretação geométrica da figura 3.7.

Como os pontos O , P e P_o pertencem ao mesmo segmento de reta, pode-se afirmar que $\overrightarrow{OP_o} = k \cdot \overrightarrow{OP}$. Além disso, a equação (3.34) relaciona a distância euclidiana radial $|\overrightarrow{PP_o}|$ e o escalar de proporcionalidade k .

$$\overrightarrow{PP_o} = \overrightarrow{OP} - \overrightarrow{OP_o} = \overrightarrow{OP} - k \cdot \overrightarrow{OP} = (1 - k) \cdot \overrightarrow{OP} \quad (3.34)$$

Uma vez que as coordenadas dos pontos O e P são conhecidas, é possível determinar a distância euclidiana radial $|\overrightarrow{PP_o}|$ em função do escalar k usando a equação (3.35).

$$\begin{aligned} F(P) = F(K \cdot P_o) &= \left[\left(\frac{K \cdot x}{a_1} \right)^{\frac{2}{\varepsilon_2}} + \left(\frac{K \cdot y}{a_2} \right)^{\frac{2}{\varepsilon_2}} \right]^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{K \cdot z}{a_3} \right)^{\frac{2}{\varepsilon_1}} \\ &= \left[K^{\frac{2}{\varepsilon_2}} \cdot \left(\frac{x}{a_1} \right)^{\frac{2}{\varepsilon_2}} + K^{\frac{2}{\varepsilon_2}} \cdot \left(\frac{y}{a_2} \right)^{\frac{2}{\varepsilon_2}} \right]^{\frac{\varepsilon_2}{\varepsilon_1}} + K^{\frac{2}{\varepsilon_1}} \cdot \left(\frac{z}{a_3} \right)^{\frac{2}{\varepsilon_1}} \\ &= K^{\frac{2}{\varepsilon_1}} \cdot F(X, Y, Z) = K^{\frac{2}{\varepsilon_1}} \cdot F(P_o) \end{aligned} \quad (3.35)$$

Elevando a equação (3.35) ao expoente $\varepsilon_1/2$, obtém-se a igualdade mostrada na equação (3.36).

$$F^{\frac{\varepsilon_1}{2}}(P) = K \cdot F(P_o) \quad (3.36)$$

Esta expressão descreve as mesmas superfícies anunciadas por Solina e Bajcsy (1990), $F^{\frac{\varepsilon_1}{2}}(P)$, a menos do fator de escala k (Chevalier *et al.*, 2003). Além disso, o fato do ponto P_o pertencer à superfície da superelipsóide, ou seja, $F(P_o) = 1$, permite definir este fator da maneira mostrada na equação (3.37).

$$k = F^{\frac{\varepsilon_1}{2}}(P) \quad (3.37)$$

A substituição deste resultado na equação (3.34) conduz à obtenção da distância euclidiana radial:

$$\overrightarrow{PP_o} = \left(1 - F^{\frac{\varepsilon_1}{2}}(P)\right) \cdot \overrightarrow{OP} \quad (3.38)$$

Reescrevendo esta expressão em função dos parâmetros da figura 3.7, tem-se:

$$d_2(A) = |r_o| \cdot \left|1 - F^{\frac{\varepsilon_1}{2}}(X, Y, Z)\right|. \quad (3.39)$$

A resolução da equação (3.39) não é computacionalmente mais onerosa que aquela proposta por Solina e Bajcsy (1990), dado o fato de que as coordenadas dos pontos O e P são conhecidas. Além disso, ela oferece a vantagem de omitir o efeito da expansão daquela abordagem e apresentar crescimento numérico uniforme, independente da direção do vetor OP (Chevalier, 2004). No entanto, a qualidade dos resultados obtidos usando a distância euclidiana radial não compensa os custos computacionais adicionais introduzidos no processo de ajuste (Katsoulas, 2004).

Desse modo, o processo de ajuste utilizado neste trabalho empregou uma versão modificada da equação (3.31), dada pela expressão (3.40).

$$d(A) = \sum_{i=1}^N \left[F^{\frac{\varepsilon_1}{2}}(A; X, Y, Z) - 1 \right]^2 \quad (3.40)$$

A escolha por tal expressão foi motivada pela eficiência computacional, uma vez que foi possível executar o processo de ajuste de aproximadamente 2400 pontos em uma média de cinco segundos empregando um *Pentium Dual Core* de 2,80GHz (32 bits), com 1,50 GB de memória RAM. Todavia, essa duração ainda pode ser reduzida com a reestruturação do código C/C++ original; substituição do processador ou, ainda, com a escolha de outra linguagem de programação.

Contudo, a qualidade aceitável dos modelos comparados experimentalmente àqueles produzidos pelas demais propostas justificaram a escolha por essa função. Além disso, as imprecisões obtidas não comprometeram o processo de ajuste para os propósitos deste trabalho. Isso se justifica pelo emprego de informações adicionais que restringiram a faixa de valores permitidos para os parâmetros das superelipsóides candidatas. As avaliações quantitativas do processo combinado às restrições mencionadas serão demonstradas no capítulo V.

Como a equação (3.40) é uma função não-linear de onze parâmetros, o processo de ajuste demanda um algoritmo de minimização iterativo. Em particular, para um conjunto de parâmetros candidatos Λ , esta equação é calculada para todos os N pontos e deve empregar um critério matemático para melhorar o resultado. O procedimento se repete para novos valores candidatos enquanto a distorção apresentada por esta expressão estiver acima de uma determinada tolerância previamente escolhida.

Este trabalho adotou a abordagem convencional descrita pelo método numérico de Marquardt (1963), denominado Levenberg-Marquardt, para o algoritmo de minimização. Ele tem sido extensivamente usado para minimizar funções objetivas (Solina e Bajcsy, 1990), (Dop e Regtien 1996), (Leonardis *et al.*, 1997), (Jaklic *et al.*, 2000), (Jaklic e Solina, 2003), (Chevalier *et al.*, 2003 e 2004), (El-Khoury e Sahbani, 2008) devido à sua estabilidade e eficiência computacionais (Zhang, 2003).

A maximização da eficiência do algoritmo foi obtida com a restrição dos valores para os parâmetros da superelipsóide, como será descrito na seção 3.5. Além disso, o confinamento desses valores a certos intervalos contribuem para a estabilidade numérica do método, uma vez que a função implícita da equação (3.17) se torna singular quando $a_1 = a_2 = a_3 = \varepsilon_1 = \varepsilon_2 = 0$ (Solina e Bajcsy, 1990). Para tanto, foram adotadas as seguintes restrições: a_1, a_2 e $a_3 > 0$ e $0,1 \leq \{\varepsilon_1, \varepsilon_2\} \leq 1,0$, pois, quando ε_1 e ε_2 são menores que 0,1, esta função pode se tornar numericamente instável, embora a forma da superelipsóide permaneça virtualmente a mesma. Em contrapartida, quando ε_1 e ε_2 são maiores que 2,0, as superfícies côncavas resultantes não representam interesse para os propósitos deste trabalho. Em particular, foram adotados os valores iniciais $\varepsilon_1 = \varepsilon_2 = 1,0$ para os parâmetros de forma da superelipsóide, indicando que uma elipsóide é o primeiro modelo candidato.

Desse modo, as próximas seções e subseções são dedicadas aos procedimentos empregados na estimativa da posição (seção 3.5.1), orientação (seção 3.5.2) e tamanho (seção 3.5.4) da nuvem de pontos a ser ajustada pela superelipsóide.

3.5 Cálculo das estimativas iniciais dos parâmetros da superelipsóide

Esta etapa é de fundamental importância para a eficiência do processo de ajuste. Como as técnicas numéricas por mínimos quadrados podem apresentar resultados diferentes daqueles esperados, é necessário determinar para qual mínimo local o método deve convergir. Todavia, durante os ensaios experimentais do algoritmo de Levenberg-Marquardt foi possível concluir que uma aproximação da posição, orientação e tamanho da nuvem de pontos era suficiente para garantir a convergência para um mínimo local que correspondia à forma original do objeto. Essa observação fornece uma informação valiosa diante da abrangência de possibilidades para os parâmetros que representam a superelipsóide e esta seção é dedicada ao cálculo das estimativas iniciais desses parâmetros.

3.5.1 Cálculo do baricentro da nuvem de pontos

A posição do centro geométrico de uma nuvem de N pontos, descrita no sistema de coordenadas global, é estimada pelo cálculo do baricentro.

$$\begin{aligned}\bar{X} &= \frac{1}{N} \cdot \sum_{i=1}^N X_i \\ \bar{Y} &= \frac{1}{N} \cdot \sum_{i=1}^N Y_i \\ \bar{Z} &= \frac{1}{N} \cdot \sum_{i=1}^N Z_i\end{aligned}\tag{3.41}$$

sendo que \bar{X} , \bar{Y} , \bar{Z} representam as coordenadas do baricentro em relação à origem do sistema de coordenadas global; e X_i , Y_i e Z_i indicam as coordenadas dos pontos nos eixos locais x , y e z , respectivamente.

Especificamente, o baricentro representa as coordenadas do vetor de translação discutido na seção 3.2 e foi aproveitado para deslocar a nuvem de pontos para a origem do sistema de coordenadas da superelipsóide candidata.

3.5.2 Estimativa inicial da orientação da nuvem de pontos

A orientação da nuvem de pontos fornece informações essenciais para a obtenção da matriz de rotação R da seção 3.2. Para a estimativa da orientação este trabalho adotou a abordagem da matriz de covariância, C , empregada por Strand e Dillmann (2009).

$$C = \frac{1}{N} \cdot \sum_{i=1}^N \begin{bmatrix} (X_i - \bar{X}) \cdot (X_i - \bar{X}) & (Y_i - \bar{Y}) \cdot (X_i - \bar{X}) & (Z_i - \bar{Z}) \cdot (X_i - \bar{X}) \\ (Y_i - \bar{Y}) \cdot (X_i - \bar{X}) & (Y_i - \bar{Y}) \cdot (Y_i - \bar{Y}) & (Z_i - \bar{Z}) \cdot (Y_i - \bar{Y}) \\ (Z_i - \bar{Z}) \cdot (X_i - \bar{X}) & (Z_i - \bar{Z}) \cdot (Y_i - \bar{Y}) & (Z_i - \bar{Z}) \cdot (Z_i - \bar{Z}) \end{bmatrix} \quad (3.42)$$

Em contrapartida, as matrizes de inércia propostas por Solina e Bajcsy (1990), Bardinet (1995) e Chevalier (2004) representam métodos alternativos para a obtenção da orientação.

Os autovetores destas matrizes indicam a direção dos eixos de inércia da nuvem de pontos em relação à origem do sistema de coordenadas global, como mostra a interpretação geométrica da figura 3.8.

A solução do auto-sistema definido pela matriz C demanda uma nova representação na forma de uma matriz diagonal, obtida por transformações de similaridades. Como resultado, os autovalores λ_1 , λ_2 e λ_3 são designados pelos elementos da matriz diagonal resultante, enquanto os autovetores correspondentes, v_1 , v_2 e v_3 podem ser obtidos a partir das colunas da matriz de rotação envolvida neste processo. Os detalhes deste procedimento podem ser encontrados em Press *et al.* (1992).

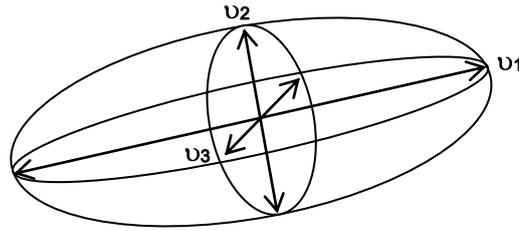


Figura 3.8 - Superelipsóide localizada na origem do sistema de coordenadas e a relação geométrica com o auto-sistema definido pela matriz de covariância. Os autovetores correspondentes u_1 , u_2 e u_3 indicam as direções dos eixos de inércia ao longo dos eixos coordenados X , Y e Z , respectivamente.

O método de Jacobi foi utilizado para calcular este auto-sistema. De forma resumida, o método consiste de uma sequência de transformações de similaridade dedicada a anular em nível computacional os elementos não contidos na diagonal principal da matriz original. Além disso, ele é rápido para matrizes de ordem inferior a 10 e confiável para qualquer matriz simétrica (Press *et al.*, 1992). Uma implementação computacional desse método foi integrada ao programa principal e uma distribuição gratuita da versão utilizada está disponível no endereço eletrônico <http://math.nist.gov/javanumerics/jama>, acessado em 26/06/2010.

A segunda consequência no uso do método de Jacobi é revelada pelo fato da matriz de rotação envolvida na transformação de similaridade descrever os componentes necessários para a obtenção dos ângulos de Euler ϕ , θ e ψ . Inspirado no formalismo apresentado por Strand e Dillmann (2009) para expressar a relação entre esses parâmetros e os autovetores, este trabalho adotou a notação exibida na equação (3.43).

$$\begin{aligned}\phi &= \arctan\left(\frac{v_{2,3}}{v_{1,3}}\right) \\ \theta &= \arccos(v_{3,3}) \\ \psi &= \arctan\left(\frac{v_{3,2}}{-v_{3,1}}\right)\end{aligned}\tag{3.43}$$

sendo que $v_{i,j}$ representa o j -ésimo componente do i -ésimo autovetor.

A partir das informações relacionadas à orientação e posição da nuvem de pontos, a próxima subseção apresenta o procedimento utilizado para definir os eixos de inércia em relação a um novo sistema de coordenadas centrado no baricentro deste conjunto.

3.5.3 Determinação do eixo de inércia principal da superelipsóide

Solina e Bajcsy (1990) demonstraram que diferentes conjuntos de valores para os parâmetros de uma superelipsóide podem simular modelos virtualmente idênticos, porém, inconsistentes com aqueles esperados. Com a finalidade de garantir que o método de minimização apresentasse convergência para resultados particularmente desejados foi necessário restringir o domínio de valores para os parâmetros.

O primeiro procedimento consistiu em alinhar os eixos coordenados àqueles descritos pelas direções dos eixos do momento de inércia para cada item a ser modelado. Tendo em vista que objetos modelados com superelipsóides modificadas pelas deformações propostas por Barr (1984) são descritas em torno do eixo z , este foi adotado como eixo principal para um novo sistema de coordenadas.

Por meio da organização crescente dos autovalores, os três autovetores associados indicaram as direções dos novos eixos coordenados x , y e z , em especial do eixo de inércia principal, de acordo com o critério de Solina e Bajcsy (1990):

Quando $|\lambda_1 - \lambda_2| < |\lambda_2 - \lambda_3|$ a direção do novo eixo z é definida pelas coordenadas do autovetor v_3 . Em caso contrário, a direção deste eixo é indicada pelo autovetor v_1 .

Esta condição orienta o maior lado dos objetos centrados no sistema de coordenadas local ao longo do novo eixo z para objetos alongados (eixo de menor inércia) e ao longo do menor lado para objetos de perfis baixos (eixo de maior inércia), baseado no fato de que as deformações adicionadas aos modelos padrões normalmente afetam os objetos ao longo de seu maior lado (Solina e Bajcsy, 1990). A interpretação geométrica desse critério é mostrada na figura 3.9.

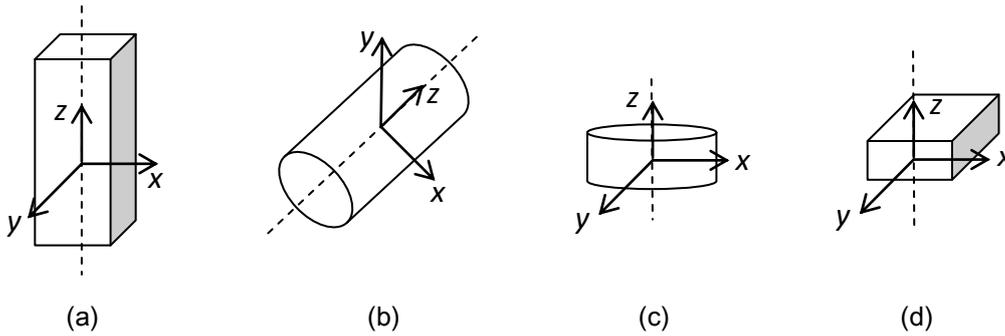


Figura 3.9 – Interpretação geométrica do critério utilizado para definir a direção do eixo de inércia principal: (a) e (b) linha tracejada indica a direção do eixo de menor inércia para os objetos alongados; (c) e (d) orientação do eixo de maior inércia para itens de perfis baixos.

Contudo, as ambigüidades observadas por Solina e Bajcsy (1990) e Strand e Dillmann (2009) não ocorreram nas simulações das primitivas adotadas neste trabalho.

Tendo sido estabelecida a orientação dos novos eixos coordenados da superelipsóide, a subseção a seguir descreve a obtenção da estimativa do tamanho do modelo baseada na distribuição espacial dos pontos.

3.5.4 Estimativa do tamanho da superelipsóide candidata

Em combinação com a orientação e o eixo de inércia principal, a estimativa inicial para o tamanho da superelipsóide candidata finaliza as condições iniciais responsáveis pela maximização da eficiência do método de ajuste numérico.

A abordagem original proposta por Solina e Bajcsy (1990) para estimar o tamanho de uma nuvem de pontos consistiu no cálculo da distância do centro de gravidade aos pontos mais afastados ao longo de cada um dos eixos coordenados. Em contrapartida, Chevalier (2004) empregou o método apresentado por Bardinet (1995) justificando que o primeiro método era extremamente sensível às imprecisões da posição dos pontos. Especificamente, a proposta de Bardinet (1995) utilizou a matriz de inércia, I , de uma nuvem de pontos para estimar o tamanho desta distribuição espacial em função dos autovalores dessa matriz.

$$I = \frac{1}{3} \cdot \begin{pmatrix} a_2^2 + a_3^2 & 0 & 0 \\ 0 & a_1^2 + a_3^2 & 0 \\ 0 & 0 & a_1^2 + a_2^2 \end{pmatrix} \quad (3.44)$$

As diagonais de I permitem o cálculo dos parâmetros de tamanho (Bardinet, 1995).

$$\begin{aligned} a_1^2 &= \frac{3}{2} \cdot (\lambda_2 + \lambda_3 - \lambda_1) \\ a_2^2 &= \frac{3}{2} \cdot (\lambda_1 + \lambda_3 - \lambda_2) \\ a_3^2 &= \frac{3}{2} \cdot (\lambda_1 + \lambda_2 - \lambda_3) \end{aligned} \quad (3.45)$$

Todavia, ensaios experimentais comparando as duas abordagens revelaram que este conceito superestimou as dimensões das nuvens de pontos.

Desse modo, a notação de Strand e Dillmann (2009) para a proposta apresentada por Solina e Bajcsy (1990) foi utilizada neste trabalho para estimar os valores iniciais de a_1 , a_2 e a_3 .

$$a_i = \max_{j=1}^N \{(P_j - \bar{P}) \cdot v_i\}, \quad (3.46)$$

sendo que:

a_i ($i = 1, 2$ e 3) representa os parâmetros de tamanho da nuvem de pontos;

P_j : vetor das coordenadas dos pontos do conjunto de dados para $j = 1$ até N ;

\bar{P} : vetor das coordenadas do baricentro da nuvem de pontos e,

v_i ($i = 1, 2$ e 3): os autovetores da matriz de covariância.

Conforme será discutido no capítulo V, as estimativas obtidas com o uso dessa equação produziram resultados muito próximos dos valores exatos, os quais contribuíram de maneira significativa para a qualidade do processo de ajuste.

3.5.5 Resultados numéricos

A tabela 3.1 apresenta os valores exatos dos parâmetros das nuvens de pontos para seis objetos: uma esfera, um cubo, dois cilindros e dois paralelepípedos.

Tabela 3.1 – Parâmetros exatos dos objetos utilizados nas simulações.

Objeto	Quantidade de pontos	Parâmetros exatos								
		a_1 (cm)	a_2 (cm)	a_3 (cm)	p_x (cm)	p_y (cm)	p_z (cm)	ϕ	θ	ψ
Esfera	2402	20	20	20	0	0	0	180°	0°	-90°
Cubo	2402	25	25	25	0	25	0	0°	0°	0°
Cilindro alto	2402	10	10	20	0	0	20	- 180°	0°	-90°
Cilindro baixo	2402	20	20	5	-8	3	6	0°	90°	180°
Paralelepípedo alto	2402	5	5	15	0	0	100	-90°	0°	90°
Paralelepípedo baixo	2402	15	10	6	-8	-5	9	0°	90°	0°

Os objetos desta tabela são itens sintéticos obtidos com o uso do *software* comercial *Autodesk 3ds Max* amplamente utilizado em computação gráfica. Este mesmo ambiente computacional produziu as 2402 coordenadas de cada uma das nuvens de pontos. A quantidade de pontos necessária para descrever as primitivas foi determinada experimentalmente.

A tabela 3.2 exibe as estimativas obtidas pelo uso das técnicas estudadas nesta seção para os parâmetros dos objetos da tabela 3.1. O baricentro calculado em relação à origem do sistema de coordenadas global apresentou aproximações numéricas satisfatórias quando comparado aos valores exatos dos modelos.

As diferenças obtidas para a orientação, com exceção do cubo e do paralelepípedo baixo, não comprometeram a estimativa precisa dos parâmetros de tamanho e posição de tais itens. Em particular, nem mesmo a grande distorção observada para a orientação da esfera, cilindros alto e baixo interferiram na estimativa para o tamanho de cada nuvem de pontos.

Tabela 3.2 – Parâmetros estimados para demonstrar a precisão das simulações.

Objeto	Quantidade de pontos	Parâmetros estimados								
		\hat{a}_1 (cm)	\hat{a}_2 (cm)	\hat{a}_3 (cm)	\hat{p}_x (cm)	\hat{p}_y (cm)	\hat{p}_z (cm)	$\hat{\phi}$	$\hat{\theta}$	$\hat{\psi}$
Esfera	2402	20	20	20	0	0	0	146°	0°	-124°
Cubo	2402	25	25	25	0	25	0	0°	0°	0°
Cilindro alto	2402	10	10,4	20	0	0	20	- 172°	0°	90°
Cilindro baixo	2402	20,4	20,4	5,1	-8	3	6	0°	90°	174°
Paralelepípedo alto	2402	5	5	15,1	0	0	100	-87°	0°	87°
Paralelepípedo baixo	2402	15,3	10,2	6,1	-8	-5	9	0°	90°	0°

Os escalares de tamanho foram obtidos pelo uso da equação (3.21) após o alinhamento dos lados do objeto aos eixos coordenados x , y e z . Esse procedimento justifica a grande aproximação numérica dos valores estimados para aqueles exatos de cada modelo.

3.6 Método numérico de Levenberg-Marquardt

O algoritmo de Levenberg-Marquardt é um método clássico de regressão não-linear. Especificamente, ele consiste de uma técnica iterativa que calcula o mínimo local de uma função expressa como a soma dos quadrados de funções lineares, tal como aquela da expressão (3.29). Ele tem se tornado uma técnica padrão para problemas de mínimos quadrados não lineares e pode ser descrito como uma combinação dos métodos *Steepest Descent* e Gauss-Newton. Quando o resultado atual está muito longe do valor correto, o algoritmo se comporta como um método *Steepest Descent*: lento, porém garante a convergência. Em contrapartida, quando a solução atual está próxima da exata ele adota a abordagem de Gauss-Newton. Esta seção apresenta uma breve descrição dos aspectos matemáticos da implementação gratuita em linguagem C/C++ do método de Levenberg-Marquardt (Lourakis, 2005).

Seja f uma relação funcional que mapeie os onze parâmetros da superelipsóide representados pelo vetor \mathbf{p} para um vetor de valores estimados $\hat{x} = f(\mathbf{p})$. Uma estimativa \mathbf{p}_0 desses dados é fornecida ao método, observando que o interesse consiste em determinar o vetor \mathbf{p}^+ que melhor satisfaça a relação funcional f , isto é, minimize o quadrado da distância ϵ^T . ϵ , na qual $\epsilon = x - \hat{x}$. A base deste algoritmo é uma aproximação linear para f nas vizinhanças de \mathbf{p} . Para um pequeno valor de $\|\delta_p\|$, uma expansão da série de Taylor conduz à seguinte aproximação:

$$f(\mathbf{p} + \delta_p) \approx f(\mathbf{p}) + \mathbf{J} \cdot \delta_p, \quad (3.47)$$

sendo \mathbf{J} a matriz jacobiana $\frac{\partial f(\mathbf{p})}{\partial \mathbf{p}}$. Assim como ocorre para todos os métodos não lineares de otimização, o algoritmo de Levenberg-Marquardt é iterativo: inicializado em uma condição \mathbf{p}_0 , o método produz uma série de vetores $\mathbf{p}_1, \mathbf{p}_2, \dots$, os quais convergem a um minimizador local \mathbf{p}^+ para f . Desse modo, cada passo exige o cálculo de δ_p que minimiza $\|x - f(\mathbf{p} + \delta_p)\| \approx \|x - f(\mathbf{p}) - \mathbf{J} \cdot \delta_p\| = \|\epsilon - \mathbf{J} \cdot \delta_p\|$. Como resultado, δ_p descreve a solução para o problema de mínimos quadrados. O mínimo é obtido quando $\mathbf{J} \cdot \delta_p - \epsilon$ é ortogonal ao espaço das colunas de \mathbf{J} . Isto conduz a $\mathbf{J}^T \cdot (\mathbf{J} \cdot \delta_p - \epsilon) = 0$, o qual conduz δ_p à solução da equação normal:

$$\mathbf{J}^T \cdot \mathbf{J} \cdot \delta_p = \mathbf{J}^T \cdot \epsilon \quad (3.48)$$

na qual $\mathbf{J}^T \cdot \mathbf{J}$ é uma aproximação para a matriz das derivadas de segunda ordem da função a ser minimizada. A implementação *levmar* oferece uma aproximação para o cálculo dessas derivadas por meio do uso da expressão (3.49).

$$\mathbf{N} \cdot \delta_p = \mathbf{J}^T \cdot \epsilon, \quad (3.49)$$

na qual os elementos fora da diagonal de \mathbf{N} são idênticos aos correspondentes termos da matriz $\mathbf{J}^T \cdot \mathbf{J}$, enquanto os elementos da diagonal são determinados como $\mathbf{N}_{ii} = \mu + [\mathbf{J}^T \cdot \mathbf{J}]_{ii}$ para algum $\mu > 0$. A estratégia de alterar os elementos diagonais de $\mathbf{J}^T \cdot \mathbf{J}$ é

denominada amortecimento e μ descreve o termo de amortecimento. Se uma atualização do vetor de parâmetros $\mathbf{p} + \delta_p$ com δ_p calculado a partir da equação (3.49) conduzir a uma redução no erro ϵ , a atualização é aceita e o processo é repetido com valores decrescentes para o fator de amortecimento. Em contrapartida, este termo é aumentado e a equação (3.49) é calculada novamente e o processo continua até um valor de δ_p que diminua o erro. Este procedimento aplicado a diferentes valores do fator de amortecimento até que uma atualização aceitável para o conjunto de parâmetros seja encontrada corresponde a uma iteração do algoritmo numérico.

O fator de amortecimento é ajustado a cada iteração para assegurar uma redução no erro ϵ . Se o amortecimento é configurado para um valor elevado, a matriz N da equação (3.49) é aproximadamente diagonal e o fator de atualização do algoritmo δ_p está próximo da direção *Steepest Descent*, enquanto a magnitude de δ_p é reduzida neste caso. O amortecimento também manipula situações nas quais o *rank* da matriz jacobiana é deficiente e $\mathbf{J}^T \cdot \mathbf{J}$ se torna singular (Lampton, 1997). Para esses casos, o algoritmo pode defensivamente navegar para uma região de parâmetros na qual o modelo seja altamente não linear. Se o fator de amortecimento for baixo, o método se aproxima de um problema completamente linear. A técnica é adaptativa uma vez que ela controla seu próprio amortecimento: ela aumenta o amortecimento quando uma iteração falha (para reduzir o erro) ou reduz o amortecimento em caso contrário. Desse modo, o algoritmo é capaz de alternar entre uma lenta abordagem descrita pela técnica *Steepest Descent* quando ele é iniciado longe do mínimo e uma rápida convergência quando o método é inicializado próximo ao mínimo (Lampton, 1997). Finalmente, o algoritmo encerra o processo na ocorrência de uma das seguintes condições:

- A magnitude do gradiente de ϵ^T . ϵ , isto é, $\mathbf{J}^T \cdot \epsilon$ da equação (3.43) está abaixo de um limite κ_1 ;
- a mudança relativa na magnitude de δ_p está abaixo de um limite κ_2 ;
- o erro ϵ^T . ϵ está abaixo do limiar κ_3 ;
- a quantidade máxima de iterações K_{\max} for atingida.

O conjunto de valores adotados para esses termos seguiram aqueles sugeridos pela distribuição do algoritmo, a saber:

$$\mu = 1\text{E-}3; \quad \kappa_1 = 1\text{E-}15; \quad \kappa_2 = 1\text{E-}15; \quad \kappa_3 = 1\text{E-}20,$$

enquanto o fator de aproximação para os elementos da matriz de Jacobi foi igual a 1E-6.

A implementação da rotina *levmar* também é capaz de considerar restrições dos parâmetros expressas por equações e inequações lineares, bem como por delimitações definidas por intervalos fechados. Além disso, as derivadas parciais de primeira ordem da função objetivo em relação aos seus parâmetros são estimadas por meio de diferenças finitas.

Dentre as funções disponíveis nesta distribuição, este trabalho utilizou uma função de otimização denominada *dlevmar_bc_dif*(). Esta função aceita restrições do domínio dos parâmetros, além de utilizar precisão numérica *double* e estimar os elementos da matriz de Jacobi a partir da declaração da função objetivo. O conjunto de dados de entrada e de saída, bem como as condições de utilização desta função são exibido na figura 3.10.

Esta função aceita como argumentos de entrada as estimativas iniciais para os parâmetros de tamanho, orientação e posição da nuvem de pontos em relação à origem do sistema de coordenadas global. Essas informações são aproveitadas pelo programa principal para compor a função objetivo da equação (3.40). Além disso, os limites inferiores e superiores dessas grandezas também são apresentados a esta função com a finalidade de reduzir o elenco de resultados que se aproximam daqueles esperados.

Como o método minimiza o escalar representado pela distância *d* da equação (3.40), ele não é capaz de considerar as dimensões das variáveis envolvidas na descrição completa da superelipsóide.

Desse modo, foi necessário formatar a equação (3.40) em uma notação compatível com aquela aceita pela rotina de minimização. Em particular, os onze parâmetros da equação (3.30) formaram o vetor Λ :

$$\Lambda = \{p[0], p[1], p[2], p[3], p[4], p[5], p[6], p[7], p[8], p[9], p[10]\}. \quad (3.50)$$

Os três primeiros termos desta equação representam os parâmetros de tamanho da superelipsóide, seguidos pelos fatores de forma, $p[4]$ e $p[5]$. Na sequência, a orientação é expressa pelos termos $p[5]$, $p[6]$ e $p[7]$, enquanto $p[8]$, $p[9]$ e $p[10]$ descrevem a posição do baricentro da nuvem de pontos em relação à origem do sistema de coordenadas global.

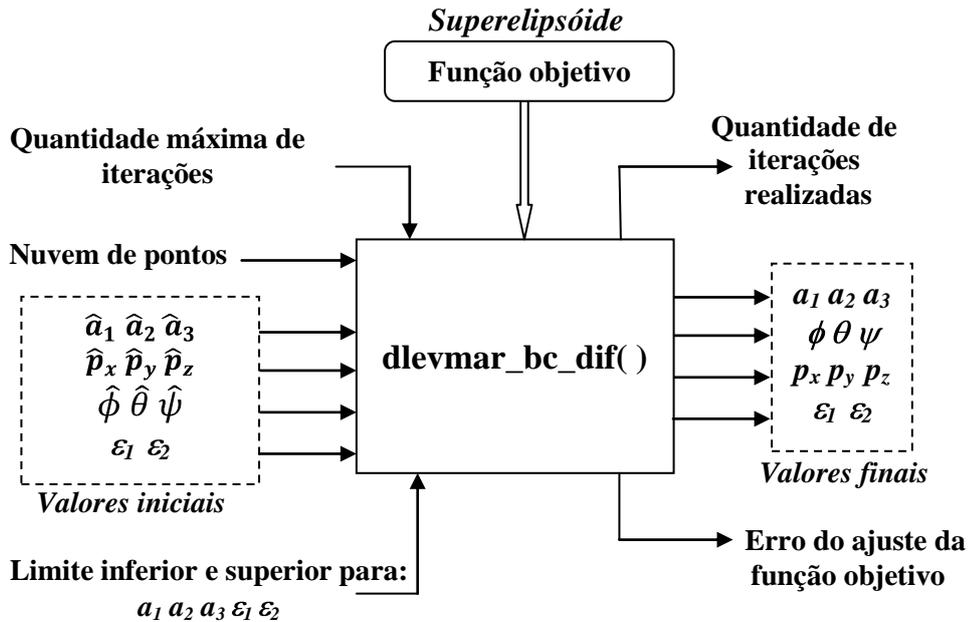


Figura 3.10 - Conjunto de entrada e de saída da função de minimização `dlevmar_bc_dif` do método de Levenberg-Marquardt.

No entanto, para que o método calculasse o mínimo local correto foi necessário delimitar os domínios dessas variáveis. Em particular, comparações empíricas das estimativas iniciais para os parâmetros de tamanho a_1 , a_2 e a_3 calculados a partir da equação (3.41) resultaram na seguinte restrição para essas grandezas:

$$0,95 \cdot \hat{a}_1 \leq p[0] \leq 1,05 \cdot \hat{a}_1$$

$$0,95 \cdot \hat{a}_2 \leq p[1] \leq 1,05 \cdot \hat{a}_2$$

$$0,95 \cdot \hat{a}_3 \leq p[2] \leq 1,05 \cdot \hat{a}_3$$

Em contrapartida, os parâmetros de forma ficaram restritos aos seguintes intervalos:

$$0,1 \leq p[3] \leq 1,0$$

$$0,1 \leq p[4] \leq 1,0$$

A escolha desses valores foi orientada pelas formas geométricas de interesse para esse trabalho, como mostrou a figura 3.3. Todavia, valores iniciais iguais à unidade foram declarados para a rotina de minimização, independente da nuvem de pontos de entrada.

Na sequência, os valores iniciais para a orientação da nuvem de pontos seguiram as restrições a seguir:

- parâmetro $p[5]$ para objetos altos:

$$\begin{aligned} 0 \leq p[5] \leq \pi/2, & \text{ para } 0 \leq \hat{\phi} \leq \pi/2 \\ (\hat{\phi} - \pi/2) \leq p[5] \leq \pi/2, & \text{ para } \pi/2 \leq \hat{\phi} \leq \pi \\ \pi \leq p[5] \leq 3\pi/2, & \text{ para } \pi \leq \hat{\phi} \leq 3\pi/2 \\ (\hat{\phi} + \pi/2) \leq p[5] \leq \pi/2, & \text{ para } 3\pi/2 \leq \hat{\phi} \leq 2\pi \end{aligned}$$

- parâmetro $p[6]$ para objetos altos:

$$\begin{aligned} 0,95 \cdot \hat{\theta} \leq p[6] \leq 1,05 \cdot \hat{\theta}, & \text{ para } 0 \leq \hat{\theta} \leq \pi/2 \\ (\hat{\theta} - \pi/2) \leq p[6] \leq \pi/2, & \text{ para } \pi/2 \leq \hat{\theta} \leq \pi \end{aligned}$$

- parâmetro $p[7]$ para objetos altos:

$$\begin{aligned} \pi/2 \leq p[7] \leq \pi, & \text{ para } 0 \leq \hat{\psi} < \pi/2 \\ (\hat{\psi} - \pi/2) \leq p[7] \leq \pi/2, & \text{ para } \pi/2 \leq \hat{\psi} < \pi \\ (\hat{\psi} + \pi) \leq p[7] \leq \pi/2, & \text{ para } \pi \leq \hat{\psi} < 3\pi/2 \\ 3\pi/2 \leq p[7] \leq 0, & \text{ para } 3\pi/2 \leq \hat{\psi} < 0 \end{aligned}$$

- parâmetro $p[5]$ para objetos baixos:

$$\begin{aligned} 0 \leq p[5] \leq \pi/2, & \text{ para } 0 \leq \hat{\phi} < \pi/2 \\ (\hat{\phi} - \pi/2) \leq p[5] \leq \pi/2, & \text{ para } \pi/2 \leq \hat{\phi} \leq \pi \\ (\hat{\phi} + \pi/2) \leq p[5] \leq \pi, & \text{ para } \pi \leq \hat{\phi} \leq 3\pi/2 \\ \hat{\phi} \leq p[5] \leq \pi, & \text{ para } 3\pi/2 < \hat{\phi} < 0 \end{aligned}$$

- parâmetro $p[6]$ para objetos baixos:

$$\hat{\theta} \leq p[6] \leq 1,05 \cdot \hat{\theta}, \text{ para } 0 \leq \hat{\theta} \leq \pi/2$$

$$\pi/2 \leq p[6] \leq \pi, \text{ para } \pi/2 \leq \hat{\theta} < \pi$$

- parâmetro p[7] para objetos baixos:

$$0,95 \cdot \hat{\psi} \leq p[7] \leq 1,05 \cdot \hat{\psi}, \text{ para } 0 \leq \hat{\psi} < \pi/2$$

$$(\hat{\psi} - \pi/2) \leq p[7] \leq \pi/2, \text{ para } \pi/2 \leq \hat{\psi} < \pi$$

$$(\hat{\psi} - \pi/2) \leq p[7] \leq \hat{\psi}, \text{ para } \pi \leq \hat{\psi} < 3\pi/2$$

$$(\hat{\psi} + \pi/2) \leq p[7] \leq \pi/2, \text{ para } 3\pi/2 \leq \hat{\psi} < 0$$

Uma rotina específica do programa principal era responsável pela adequação dos quadrantes de trabalho para cada um dos ângulos de Euler ϕ , θ e ψ a partir dos valores estimados. Finalmente, os parâmetros de translação foram confinados aos seguintes intervalos:

$$0,95 \cdot \hat{p}_x \leq p[8] \leq \hat{p}_x, \text{ para } \hat{p}_x \geq 0$$

$$1,05 \cdot \hat{p}_x \leq p[8] \leq \hat{p}_x, \text{ para } \hat{p}_x < 0$$

$$0,95 \cdot \hat{p}_y \leq p[9] \leq \hat{p}_y, \text{ para } \hat{p}_y \geq 0$$

$$1,05 \cdot \hat{p}_y \leq p[9] \leq \hat{p}_y, \text{ para } \hat{p}_y < 0$$

$$0,95 \cdot \hat{p}_z \leq p[10] \leq \hat{p}_z, \text{ para } \hat{p}_z \geq 0$$

$$1,05 \cdot \hat{p}_z \leq p[10] \leq \hat{p}_z, \text{ para } \hat{p}_z < 0$$

O capítulo que segue apresenta uma descrição detalhada do fluxograma computacional desenvolvido para o cálculo automático de todas as grandezas discutidas.

$$F(X, Y, Z) =$$

$$\left\{ \left[\left(\frac{[\cos(\phi).\cos \theta.\cos(\psi)-\text{sen}(\phi).\text{sen}(\psi)].(X-p_x)+[\text{sen}(\phi).\cos(\theta).\cos(\psi)+\cos(\phi).\text{sen}(\phi)].(Y-p_y)- [\text{sen}(\phi).\cos(\psi)].(Z-p_z)}{a_1} \right)^{\frac{2}{\varepsilon_2}} + \right. \right.$$

$$\left. \left(\frac{[-\cos(\phi).\cos(\theta).\text{sen}(\psi)-\text{sen}(\phi).\cos(\psi)].(X-p_x)+[-\text{sen}(\phi).\cos(\theta).\text{sen}(\psi)+\cos(\phi).\cos(\phi)].(Y-p_y)+ [\text{sen}(\theta).\text{sen}(\psi)].(Z-p_z)}{a_2} \right)^{\frac{2}{\varepsilon_2}} \right]^{\frac{\varepsilon_2}{\varepsilon_1}} +$$

$$\left. \left(\frac{[\cos(\phi).\text{sen}(\theta)].(X-p_x)+[\text{sen}(\phi).\text{sen}(\theta)].(Y-p_y)+ \cos(\theta)(Z-p_z)}{a_3} \right)^{\frac{2}{\varepsilon_1}} \right\}$$

(3.29)

Programa de ajuste de superelipsóides

Os diversos aspectos da modelagem de nuvens de pontos empregando superelipsóides foram formulados pelas descrições matemáticas do capítulo anterior. O fluxograma desenvolvido para calcular os parâmetros de ajuste a partir de um arquivo de entrada é apresentado nesta etapa do trabalho. Especificamente, o programa computacional foi escrito em linguagem C/C++ para o compilador *Visual Studio* da *Microsoft*.

Com o objetivo de tornar público e demonstrar total transparência no desenvolvimento deste estágio do projeto, as principais linhas de código são exibidas. Além disso, as estratégias utilizadas para superar as dificuldades na elaboração do programa também são discutidas. Esta atitude caracteriza uma contribuição importante, uma vez que não foi possível encontrar literatura pertinente às propostas deste trabalho com o nível de detalhamento apresentado. Todavia, a continuidade de entendimento deste projeto independe da leitura das seções subseqüentes, permitindo, alternativamente, que o leitor redirecione a leitura para o próximo capítulo sem comprometer a compreensão da proposta.

4.1 Fluxograma de ajuste de superelipsóides

A figura 4.1 ilustra o fluxograma dos dois blocos computacionais desenvolvidos: o primeiro estágio deste diagrama é formado pelo programa numérico responsável pelo ajuste das superelipsóides, enquanto o segundo se concentra na apresentação gráfica da nuvem de pontos e da superfície que modela este conjunto. Os itens sintéticos a serem modelados por essas superfícies foram produzidos no *software* comercial denominado *Autodesk 3ds Max*. Este ambiente gráfico também foi o responsável pela geração dos arquivos de pontos correspondentes.

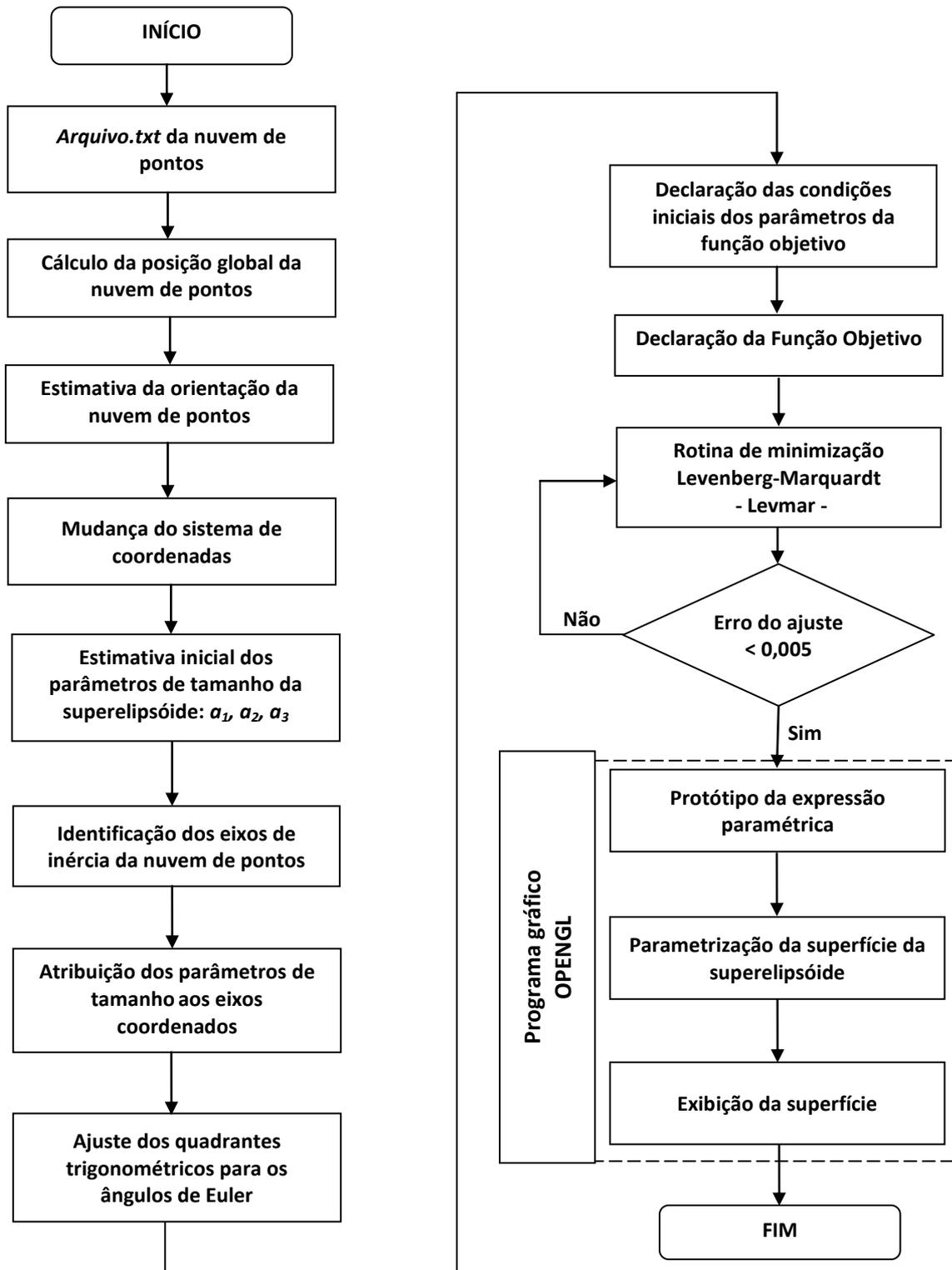


Figura 4.1 - Fluxograma computacional desenvolvido para o processo de ajuste das superelipsóides.

4.1.1 Nuvem de pontos

As coordenadas dos pontos são registradas em arquivos.txt convencionais e um fragmento do seu conteúdo é ilustrado na figura 4.2. A palavra chave *Vertices* é interpretada pelo programa principal para reconhecer a natureza do arquivo, enquanto o escalar 2402 indica a quantidade de pontos 3D que descreve o objeto. Essa informação é utilizada pelas rotinas de leitura das coordenadas de cada ponto, exigidas nos cálculos do baricentro, da matriz de covariância e estimativa do tamanho da nuvem de pontos na direção dos eixos coordenados.

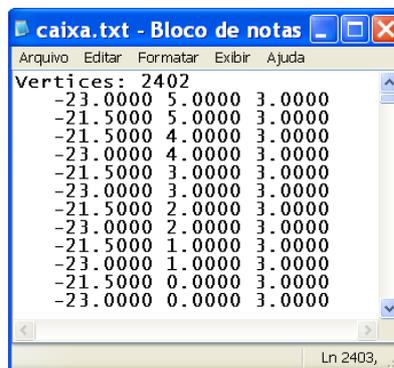


Figura 4.2 - Captura de tela de um *arquivo.txt* utilizado para registrar as coordenadas dos pontos.

4.1.2 Cálculo da posição global da nuvem de pontos

A primeira, segunda e terceira colunas do arquivo da seção anterior fornecem as coordenadas globais X , Y e Z para cada um dos 2402 pontos. As linhas de código a seguir calculam a média aritmética de cada coluna para obter o baricentro $(\bar{X}, \bar{Y}, \bar{Z})$ da nuvem de pontos.

```
for (i=0; i<N; i++)
{
  X = ((double *)coord)[3*i]; //acesso à 1ª coluna do arquivo
  Y = ((double *)coord)[3*i+1]; //acesso à 2ª coluna do arquivo
  Z = ((double *)coord)[3*i+2]; //acesso à 3ª coluna do arquivo

  Xc = Xc + X; // Somatório das coordenadas X
  Yc = Yc + Y; // Somatório das coordenadas Y
  Zc = Zc + Z; // Somatório das coordenadas Z
}
```

```

Xc = Xc/(double)N; // coordenada X do baricentro
Yc = Yc/(double)N; // coordenada Y do baricentro
Zc = Zc/(double)N; // coordenada Z do baricentro

```

As coordenadas do baricentro em relação aos eixos coordenados x , y e z , respectivamente, são representadas pelas variáveis Xc , Yc e Zc .

Conforme mencionado no capítulo anterior, a maximização da eficiência do método de Levenberg-Marquardt pode ser obtida pela restrição numérica dos parâmetros a serem ajustados. Desse modo, a declaração dos limites inferiores e superiores para os parâmetros da posição global $p[5]$, $p[6]$ e $p[7]$ também é mostrada nas linhas abaixo.

```

if (Xc >= 0.0)
{
    lb[5] = Xc*0.95;    ub[5] = Xc*1.05; // Xc >= 0.0
}
else
{
    lb[5] = Xc*1.05;    ub[5] = Xc*0.95; // Xc < 0.0
}

if (Yc >= 0.0)
{
    lb[6] = Yc*0.95;    ub[6] = Yc*1.05; // Yc >= 0.0
}
else
{
    lb[6] = Yc*1.05;    ub[6] = Yc*0.95; // Yc < 0.0
}

if (Zc >= 0.0)
{
    lb[7] = Zc*0.95;    ub[7] = Zc*1.05; // Zc >= 0.0
}
else
{
    lb[7] = Zc*1.05;    ub[7] = Zc*0.95; // Zc < 0.0
}

```

Como a estimativa das variáveis que designam o tamanho das nuvens de pontos se aproximou dos valores exatos, vide tabela 3.2, foi suficiente estabelecer uma tolerância de $\pm 5\%$ para esses parâmetros.

4.1.3 Cálculo da orientação da nuvem de pontos

Por meio da combinação dos 2402 pontos às coordenadas do baricentro, foi possível calcular os elementos da matriz de covariância;

```

for (i=0; i<N; i++)
{
  X = ((double *)coord)[3*i]; //acesso à 1ª coluna do arquivo
  Y = ((double *)coord)[3*i+1]; //acesso à 2ª coluna do arquivo
  Z = ((double *)coord)[3*i+2]; //acesso à 3ª coluna do arquivo

  // Declaração dos elementos da Matriz de covariância
  C[0][0] += 1E-127 + (X-Xc)*(X-Xc);
  C[0][1] += 1E-127 + (Y-Yc)*(X-Xc);
  C[0][2] += 1E-127 + (Z-Zc)*(X-Xc);

  C[1][0] += 1E-127 + (Y-Yc)*(X-Xc);
  C[1][1] += 1E-127 + (Y-Yc)*(Y-Yc);
  C[1][2] += 1E-127 + (Z-Zc)*(Y-Yc);

  C[2][0] += 1E-127 + (Z-Zc)*(X-Xc);
  C[2][1] += 1E-127 + (Z-Zc)*(Y-Yc);
  C[2][2] += 1E-127 + (Z-Zc)*(Z-Zc);
}

// Somatório normalizado para a quantidade de pontos.
C[0][0] = C[0][0] / (double)N;
C[0][1] = C[0][1] / (double)N;
C[0][2] = C[0][2] / (double)N;

C[1][0] = C[1][0] / (double)N;
C[1][1] = C[1][1] / (double)N;
C[1][2] = C[1][2] / (double)N;

C[2][0] = C[2][0] / (double)N;
C[2][1] = C[2][1] / (double)N;
C[2][2] = C[2][2] / (double)N;

```

É importante observar que um resíduo igual a 1E-127 (valor mínimo adotado no programa para indicar zero) foi adicionado a cada um dos elementos dessa matriz para garantir a exatidão da rotina computacional utilizada para calcular o auto-sistema definido por C . A chamada a essa rotina é exibida nas linhas de código mostradas a seguir.

As variáveis C , V e d representam a matriz de covariância, os autovetores e autovalores, respectivamente.

```

// Chamada à rotina responsável pelo cálculo do auto-sistema
    eigen_decomposition(C, V, d);

// Exibição dos resultados: AUTO-VALORES e AUTO-VETORES
for(int i=0; i<3; i++)
{
    printf("Eigenvalue [%i]: %lf\n", i, d[i]); // Auto-Valores.
    for(int j=0; j<3; j++)
        printf(" %lf\n", V[j][i]); // Auto-Vetores NORMALIZADOS !
}

```

Em particular, a rotina utilizada, denominada *eigen_decomposition* (Hicklin *et al.*, 2010), é uma distribuição gratuita de um método de cálculo de auto-sistemas. Finalmente, a relação matemática discutida na seção 3.52 para estimar a orientação da nuvem a partir dos autovetores é implementada nas linhas de código mostradas a seguir:

```

// * Cálculo da orientação da nuvem de pontos */

    PHI_inicial = atan2(V[2][1],V[2][0]);
    THETA_inicial = acos(V[2][2]);
    PSI_inicial = atan2(-V[1][2],V[0][2]);

```

4.1.4 Mudança do sistema de coordenadas da nuvem de pontos

Conforme mencionado no capítulo anterior, as coordenadas dos pontos devem ser representadas no sistema de coordenadas localizado na origem da superelipsóide. Para isso, a substituição dos ângulos de Euler da matriz de rotação pelos parâmetros a serem ajustados pelo método de minimização é mostrada nas seguintes linhas de código.

```

x = (cos(p[5])*cos(p[6])*cos(p[7])-sin(p[5])*sin(p[7]))*(X-p[8]) +
    (sin(p[5])*cos(p[6])*cos(p[7])+cos(p[5])*sin(p[6]))*(Y-p[9]) +
    (-sin(p[6])*cos(p[7]))*(Z-p[10]);

y = (-cos(p[5])*cos(p[6])*sin(p[7])-sin(p[5])*cos(p[7]))*(X-p[8]) +
    (-sin(p[5])*cos(p[6])*sin(p[7])+cos(p[5])*cos(p[6]))*(Y-p[9]) +
    (sin(p[6])*sin(p[7]))*(Z-p[10]);

z = (cos(p[5])*sin(p[6]))*(X-p[8])+(sin(p[5])*sin(p[6]))*(Y-p[9])+
    (cos(p[6]))*(Z-p[10]);

```

Os ângulos que expressam a orientação da nuvem de pontos ϕ , θ e ψ foram substituídos pelos parâmetros $p[5]$, $p[6]$ e $p[7]$, respectivamente, enquanto x , y e z representam as coordenadas locais dos pontos.

4.1.5 Estimativa inicial do tamanho da nuvem de pontos

As linhas de código abaixo preparam a equação (3.41) para o cálculo das estimativas iniciais das variáveis que designam o tamanho da nuvem de pontos em relação aos eixos coordenados.

```
// Determinação do tamanho da nuvem de pontos //
for(int k=0; k <3; k++)
{
  for(i=0;i<N;i++)
  {
    X = ((double *)coord)[3*i]; //acesso à 1ª coluna do arquivo
    Y = ((double *)coord)[3*i+1]; //acesso à 1ª coluna do arquivo
    Z = ((double *)coord)[3*i+2]; //acesso à 1ª coluna do arquivo

    // declaração das variáveis auxiliares para o tamanho
    p_aux[i][0] = (X-Xc)*V[0][0]+(Y-Yc)*V[1][0]+(Z-Zc)*V[2][0]; //eixo_x
    p_aux[i][1] = (X-Xc)*V[0][1]+(Y-Yc)*V[1][1]+(Z-Zc)*V[2][1]; //eixo_y
    p_aux[i][2] = (X-Xc)*V[0][2]+(Y-Yc)*V[1][2]+(Z-Zc)*V[2][2]; //eixo_z
  }
}
```

As variáveis auxiliares $p_aux[i][0]$, $p_aux[i][1]$ e $p_aux[i][2]$ foram empregadas em substituição aos parâmetros de tamanho $p[0]$, $p[1]$ e $p[2]$, uma vez que esta etapa ainda descreve um estágio intermediário. Desse modo, os resultados deste fragmento de código ainda não podem ser diretamente utilizados na finalização da função objetivo a ser minimizada. A próxima seção descreve o procedimento responsável pela atribuição correta das estimativas iniciais para o tamanho da superelipsóide em relação a cada um dos eixos coordenados.

4.1.6 Identificação dos eixos de inércia

Os escalares que retratam os autovalores foram organizados em ordem crescente de tal modo que os autovetores correspondentes pudessem indicar a direção dos eixos de inércia, como mostram as linhas de código a seguir.

```
// Organização dos parâmetros de tamanho p[0], p[1], p[2]
// Somente para objetos regulares (todos os lados iguais)
```

```
if((float)p[0] == (float)p[1])
{
    printf("a[0] = a[1]\n\n");
    p_maior = p[0];
    if(p[1] >= p[2])
    {
        printf("p[1] eh o intermediario\n");
        p_int = p[1];
        printf("p[2] eh o menor\n");
        p_menor = p[2];
    }
}

if((float)p[0] == (float)p[2])
{
    printf("p[0] = p[2]\n\n");
    p_maior = p[0];
    if(p[2] >= p[1])
    {
        printf("p[2] eh o intermediario\n");
        p_int = p[2];
        printf("p[1] eh o menor\n");
        p_menor = p[1];
    }
}

if((float)p[1] == (float)p[2])
{
    printf("p[1] = p[2]\n\n");
    p_maior = p[1];
    if(p[1] >= p[0])
    {
        printf("p[1] eh o intermediario\n");
        p_int = p[1];
        printf("p[0] eh o menor\n");
        p_menor = p[0];
    }
}
```

```

// SOMENTE para objetos irregulares
if((p[0] != p[1]) && (p[0] != p[2]) && (p[2] != p[1]))
{
    if(p[2] > p[0]) //p[2] > p[0] > p[1]
    {
        aux_maior = p[2];

        if(aux_maior > p[1])
        {
            p_maior = aux_maior;
            printf("p[2] eh o maior valor\n");
            p_maior = p[2];
            if(p[0] > p[1])
            {
                printf("p[0] eh intermediario\n");
                p_int = p[0];
                printf("p[1] eh o menor valor\n");
                p_menor = p[1];
            }
            else if(p[1] > p[0])
            {
                printf("p[1] eh intermediario\n");
                p_int = p[1];
                printf("p[0] eh o menor valor\n");
                p_menor = p[0];
            }
        }
    }

    if(p[1] > p[2]) // p[1] > p[2] > p[0]
    {
        aux_maior = p[1];

        if(aux_maior > p[0])
        {
            p_maior = aux_maior;
            printf("p[1] eh o maior valor\n");
            p_maior = p[1];
            if(p[0] > p[2])
            {
                printf("p[0] eh o intermediario\n");
                p_int = p[0];
                printf("p[2] eh o menor\n");
                p_menor = p[2];
            }
            else if(p[2] > p[0])
            {
                printf("p[2] eh o intermediario\n");
                p_int = p[2];
                printf("p[0] eh o menor\n");
                p_menor = p[0];
            }
        }
    }
}

```

```

else
{
    p_maior = p[2];
    printf("p[2] eh o maior valor\n");
    p_maior = p[2];
    if(p[0] > p[1])
    {
        printf("p[0] eh intermediario\n");
        p_int = p[0];
        printf("p[1] eh o menor\n");
        p_menor = p[1];
    }
    else if(p[1] > p[0])
    {
        printf("p[1] eh intermediario\n");
        p_int = p[1];
        printf("p[0] eh o menor\n");
        p_menor = p[0];
    }
}
else if(p[0] > p[1]) // p[0] > p[1] > p[2]
{
    aux_maior = p[0];

    if(aux_maior > p[2])
    {
        p_maior = aux_maior;
        printf("p[0] eh o maior valor\n");
        p_maior = p[0];
        if(p[1] > p[2])
        {
            printf("p[1] eh intermediario\n");
            p_int = p[1];
            printf("p[2] eh o menor\n");
            p_menor = p[2];
        }
        else if(p[2] > p[1])
        {
            printf("p[2] eh intermediario\n");
            p_int = p[2];
            printf("p[1] eh o menor\n");
            p_menor = p[1];
        }
    }
}
}

```

Com a ordenação das variáveis que representam a extensão das nuvens de pontos, a próxima seção atribui esses escalares aos respectivos eixos do sistema de coordenadas localizado na origem da superelipsóide candidata a modelar tais pontos.

4.1.7 Atribuição dos parâmetros de tamanho aos eixos coordenados

Por intermédio das instruções condicionais das linhas de código exibidas a seguir, foi possível associar corretamente as estimativas iniciais para a extensão da superelipsóide em relação aos respectivos eixos coordenados locais.

```
// Atribuição dos parâmetros p[0] p[1] p[2] de acordo com os auto-vetores.
{
    if(FABS((float)V[0][0]) >= 0.5)          p[0] = p_menor;
    if(FABS((float)V[1][0]) >= 0.5)          p[1] = p_menor;
    if(FABS((float)V[2][0]) >= 0.5)          p[2] = p_menor;

    if(FABS((float)V[0][1]) >= 0.5)          p[0] = p_int;
    if(FABS((float)V[1][1]) >= 0.5)          p[1] = p_int;
    if(FABS((float)V[2][1]) >= 0.5)          p[2] = p_int;

    if(FABS((float)V[0][2]) >= 0.5)          p[0] = p_maior;
    if(FABS((float)V[1][2]) >= 0.5)          p[1] = p_maior;
    if(FABS((float)V[2][2]) >= 0.5)          p[2] = p_maior;
}
```

Como consequência, os parâmetros de tamanho $p[0]$, $p[1]$ e $p[2]$ da função objetivo da equação (3.26) estão consistentes com aqueles que descrevem o tamanho das nuvens de pontos. A constante igual a 0,5 das instruções condicionais foi obtida a partir de ensaios experimentais realizados com um amplo elenco de nuvem de pontos, enquanto os limites inferiores e superiores para esses parâmetros são expostos a seguir.

```
lb[0] = p[0]*0.95;          ub[0] = p[0]*1.05;
lb[1] = p[1]*0.95;          ub[1] = p[1]*1.05;
lb[2] = p[2]*0.95;          ub[2] = p[2]*1.05;
```

4.1.8 Ajuste dos quadrantes trigonométricos para os ângulos de Euler

Como as expressões das superelipsóides são definidas em relação à origem do sistema de coordenadas local, enquanto as suas extensões estão alinhadas aos eixos coordenados, foi necessário configurar a orientação da nuvem de pontos à pose do modelo geométrico. Além disso, a eficiência da rotina de minimização depende da atribuição correta dos quadrantes de trabalho para os ângulos de Euler, observando que

este procedimento é fundamental para a convergência do método numérico para o mínimo local esperado. Desse modo, esta seção do trabalho representou uma dificuldade adicional a ser superada.

Com o objetivo de simplificar a adequação desses quadrantes dois blocos computacionais foram desenvolvidos: o primeiro deles foi dedicado à definição dos quadrantes de trabalho para objetos altos, enquanto o segundo se concentrou em itens baixos. Em particular, este processo demandou uma tarefa onerosa de inúmeras tentativas orientadas por pequenos erros do ajuste das superelipsóides às nuvens de pontos.

```
// Declaração dos quadrantes trigonométricos para objetos altos
// Ângulo PHI -----
if(PHI_inicial >= 0.0 && PHI_inicial < (float)(M_PI/2.0))
{
    lb[5] = 0.0;                ub[5] = M_PI/2.0;
    printf("PHI: 1o Quadrante\n");
}

if(PHI_inicial >= (float)(M_PI/2.0) && PHI_inicial < M_PI)
{
    lb[5] = PHI_inicial-M_PI/2.0;    ub[5] = M_PI/2.0;
    printf("PHI: 2o Quadrante\n");
}

if(PHI_inicial >= - M_PI && PHI_inicial < -(float)(M_PI/2.0))
{
    lb[5] = M_PI;                ub[5] = 3*M_PI/2.0;
    printf("PHI: 3o Quadrante\n");
}

if(PHI_inicial >= -(float)(M_PI/2.0) && PHI_inicial < 0.0)
{
    lb[5] = PHI_inicial+M_PI/2.0;    ub[5] = M_PI/2.0;
    printf("PHI: 4o Quadrante\n");
}

// Ângulo THETA -----
if(THETA_inicial >= 0.0 && THETA_inicial < (float)(M_PI/2.0))
{
    lb[6] = THETA_inicial*0.95;    ub[6] = THETA_inicial*1.05;
    printf("THETA: 1o Quadrante\n");
}

if(THETA_inicial >= (float)(M_PI/2.0) && THETA_inicial < M_PI)
{
    lb[6] = THETA_inicial-M_PI/2.0;    ub[6] = M_PI/2.0;
}
```

```

printf("THETA: 2o Quadrante\n");
}

// Ângulo PSI -----
if(PSI_inicial >= 0.0 && PSI_inicial < (float)(M_PI/2.0))
{
    lb[7] = M_PI/2.0;                ub[7] = M_PI;
    printf("PSI: 1o Quadrante\n");
}

if(PSI_inicial >= (float)(M_PI/2.0) && PSI_inicial < M_PI)
{
    lb[7] = PSI_inicial-M_PI/2.0;    ub[7] = M_PI/2.0;
    printf("PSI: 2o Quadrante\n");
}

if(PSI_inicial >= - M_PI && PSI_inicial < -(float)(M_PI/2.0))
{
    lb[7] = PSI_inicial+M_PI;        ub[7] = M_PI/2.0;
    printf("PSI: 3o Quadrante\n");
}

if(PSI_inicial >= -(float)(M_PI/2.0) && PSI_inicial < 0.0)
{
    lb[7] = - M_PI/2.0;              ub[7] = 0.0;
    printf("PSI: 4o Quadrante\n");
}

```

Em contrapartida, os quadrantes para objetos baixos foram declarados como mostram as linhas de código a seguir.

```

// Declaração dos quadrantes de trabalho para objetos baixos
// Ângulo PHI -----
if(PHI_inicial >= 0.0 && PHI_inicial < (float)(M_PI/2.0))
{
    lb[5] = 0.0;                    ub[5] = M_PI/2.0;
    printf("PHI: 1o Quadrante\n");
}

if(PHI_inicial >= (float)(M_PI/2.0) && PHI_inicial <= M_PI)
{
    lb[5] = PHI_inicial-M_PI/2.0;    ub[5] = M_PI/2.0;

    printf("PHI: 2o Quadrante\n");
}

if(PHI_inicial >= - M_PI && PHI_inicial <= -(float)(M_PI/2.0))
{

```

```

    lb[5] = PHI_inicial + (M_PI/2.0);          ub[5] = M_PI;
    printf("PHI: 3o Quadrante\n");
}

if(PHI_inicial > -(float)(M_PI/2.0) && PHI_inicial < 0.0)
{
    lb[5] = PHI_inicial;                      ub[5] = M_PI;
    printf("PHI: 4o Quadrante\n");
}

// Ângulo THETA -----
if(THETA_inicial >= 0.0 && THETA_inicial < (float)(M_PI/2.0))
{
    lb[6] = THETA_inicial;                    ub[6] = THETA_inicial*1.05;
    printf("THETA: 1o Quadrante\n");
}

if(THETA_inicial >= (float)(M_PI/2.0) && THETA_inicial < (float)(M_PI))
{
    lb[6] = M_PI;                            ub[6] = M_PI;
    printf("THETA: 2o Quadrante\n");
}

// Ângulo PSI -----
if(PSI_inicial >= (float)(0.0) && PSI_inicial < (float)(M_PI/2.0))
{
    lb[7] = PSI_inicial*0.95;                 ub[7] = PSI_inicial*1.05;
    printf("PSI: 1o Quadrante\n");
}

if(PSI_inicial >= (float)(M_PI/2.0) && PSI_inicial < (float)M_PI)
{
    lb[7] = PSI_inicial-M_PI/2.0;             ub[7] = M_PI/2.0;
    printf("PSI: 2o Quadrante\n");
}

if(PSI_inicial >= -(float)(M_PI) && PSI_inicial < -(float)(M_PI/2.0))
{
    lb[7] = PSI_inicial;                      ub[7] = PSI_inicial;
    printf("PSI: 3o Quadrante\n");
}

if(PSI_inicial >= -(float)(M_PI/2.0) && PSI_inicial < (float)(0.0))
{
    lb[7] = PSI_inicial+M_PI/2.0;             ub[7] = M_PI/2.0;
    printf("PSI: 4o Quadrante\n");
}

```

Novamente, os termos **lb[.]** e **ub[.]** representam, nesta ordem, os limites inferiores e superiores permitidos para as estimativas da orientação **p[8]**, **p[9]** e **p[10]**, os quais retratam os ângulos de Euler ϕ , θ e ψ , respectivamente.

4.1.9 Declaração das condições iniciais dos parâmetros da função objetivo

As condições iniciais para cada um dos onze parâmetros da expressão (3.27) foram declaradas como mostram as linhas a seguir:

```
// Declaração das condições iniciais para o tamanho da superelipsóide
p[0] = lb[0]; // a1_inicial = limite inferior
p[1] = lb[1]; // a2_inicial = limite inferior
p[2] = lb[2]; // a3_inicial = limite inferior

// Declaração das condições iniciais para os expoentes de forma
p[3] = 1.0; // E1_inicial
p[4] = 1.0; // E2_inicial

// Declaração das condições iniciais para a Posição
p[5] = Xc; // coordenada X do baricentro
p[6] = Yc; // coordenada Y do baricentro
p[7] = Zc; // coordenada Z do baricentro

// Declaração das condições iniciais para a Orientação
p[8] = lb[8]; // PHI_inicial = limite inferior
p[9] = lb[9]; // THETA_inicial = limite inferior
p[10] = lb[10]; // PSI_inicial = limite inferior
```

4.1.10 Função objetivo a ser minimizada

Finalmente, a combinação dos resultados das subseções anteriores se manifesta na função objetivo representada pelo escalar $x[i]$:

$$x[i] = (\text{Potencia}(\text{Potencia}(\text{Potencia}(\text{FABS}(x)/p[0], 2.0/p[4]), 2.0/p[4]) + \text{Potencia}(\text{FABS}(y)/p[1], 2.0/p[4]), (\text{float})p[4]/(\text{float})p[3]) + \text{Potencia}(\text{FABS}(z)/p[2], 2.0/p[3])), p[3]/2.0) - 1.0)^*$$

$$(\text{Potencia}(\text{Potencia}(\text{Potencia}(\text{FABS}(x)/p[0], 2.0/p[4]) + \text{Potencia}(\text{FABS}(y)/p[1], 2.0/p[4]), (\text{float})p[4]/(\text{float})p[3]) + \text{Potencia}(\text{FABS}(z)/p[2], 2.0/p[3])), p[3]/2.0) - 1.0);$$

Nesta expressão, as variáveis que designam o tamanho da superelipsóide a_1 , a_2 e a_3 foram substituídas pelos parâmetros $p[0]$, $p[1]$ e $p[2]$, nesta ordem, enquanto a variável $p[3]$ representa o fator de forma ε_1 e $p[4]$ o expoente ε_2 .

No entanto, a dificuldade identificada para trabalhar com essa expressão foi evidenciada no surgimento de valores pertencentes ao conjunto dos números complexos, resultantes de coordenadas negativas dos pontos elevadas a expoentes de forma (ε_1 e ε_2) do conjunto dos números reais. Rotinas usuais que calculam tais funções exponenciais não são configuradas para cálculos de tamanha complexidade. Para combater este comportamento indesejável, a rotina **Potencia** foi criada de acordo com as linhas de código a seguir.

```
// Definição da função potência para números complexos
double Potencia(complex<double> base, double expoente)
{
    double sinal = +1.0;
    complex<double> res = pow(base, expoente);
    float argumento_base = arg(base)*180/M_PI; // Argumento
        utilizado para o sinal do resultado Base^expoente.

    if(argumento_base == 180.0) sinal = -1.0;
    else if (argumento_base == 0.0) sinal = +1.0;

    return sinal*abs(res);
}
```

4.1.11 Rotina de minimização Levenberg-Marquardt

Com o encerramento dos procedimentos que antecedem esta subseção, todas as informações necessárias para definir a função objetivo foram obtidas. A conclusão dessas etapas permitiu que a rotina de minimização *levmar* ajustasse os onze parâmetros da expressão (3.27) para obter a superelipsóide que melhor descrevesse uma determinada nuvem de pontos. As linhas de código a seguir exibem a chamada a essa rotina computacional.

```
// ----- Chamada à função de minimização (levmar) -----//
ret = dlevmar_bc_dif(SQ, p, x, m, n, lb, ub, 8, opts, info, NULL, NULL,
(void *) coord);

// Exibição dos resultados
printf("\nLevenberg-Marquardt returned %d in %g iter, reason %g
\n\nSolution: ", ret, info[5], info[6]);
for(i=0; i<m; ++i)
    printf("%.7g ", p[i]);
```

```
printf("\n\nMinimization info:\n");
for(i=0; i<LM_INFO_SZ; ++i)
    printf("%g ", info[i]);
```

Uma captura de tela da execução do programa principal para ajustar uma superelipsóide a uma caixa irregular é exibida na figura 4.3.

```
2402 vertices
Baricentro: Xc: -8.000000 Yc: -5.000000 Zc: 9.000000
Eigenvalue [0]: 20.053289
0.000000
0.000000
1.000000
Eigenvalue [1]: 55.703580
0.000000
1.000000
0.000000
Eigenvalue [2]: 125.333056
1.000000
-0.000000
-0.000000
Orientacao da nuvem de pontos:
PHI_inicial = 0.000000
THETA_inicial = 90.045652
PSI_inicial = 0.000000
a1 = 6.000000
a2 = 10.000000
a3 = 15.000000
a[2] eh o maior valor
a[1] eh intermediario
a[0] eh o menor valor
Objeto baixo: a_Z = a_int ou a_menor
a_maior = 15.000000
a_int = 10.000000
a_menor = 6.000000
Eixo principal: Eixo_Z = e1
a1 = 15.000000
a2 = 10.000000
a3 = 6.000000
PHI: 1o Quadrante
THETA: 2o Quadrante
PSI: 1o Quadrante
Objeto baixo
Levenberg-Marquardt returned 8 in 8 iter, reason 3
Solution: 15.3 10.2 6.12 0.1 0.1 0.0001870221 3.14 0 -8 -5.000346 9
Minimization info:
41.4323 0.00030464 0.000428951 2.00093e-008 2.88207e-005 8 3 105 8 8
Tempo de execucao: 3 segundos
Pressione qualquer tecla para continuar. . .
```

Figura 4.3 – Captura de tela após a execução do programa principal para ajustar uma superelipsóide a uma caixa irregular.

As duas linhas iniciais expõem a quantidade de vértices do arquivo de entrada e as coordenadas do baricentro (-8, -5, 9), respectivamente. Essas informações pertencem a uma caixa irregular de dimensões iguais a 15, 10 e 6 unidades nas

direções dos eixos coordenados x , y e z , nesta ordem. Na sequência, os escalares 20,05, 55,70 e 125,33 retratam os autovalores, e os respectivos autovetores normalizados, obtidos a partir da matriz de covariância. A estimativa da orientação da nuvem de pontos revela uma rotação de 90 graus em relação ao eixo y , enquanto os parâmetros de tamanho intermediários são descritos $p_aux[i][0] = 6$, $p_aux[i][1] = 10$ e $p_aux[i][2] = 15$. Embora esses valores representem a verdadeira extensão da nuvem de pontos, eles ainda estão incoerentes com os eixos coordenados. A ordenação designada por $a[2]$, $a[1]$ e $a[0]$ classificam esses resultados como maior, intermediário e menor valor, respectivamente.

A próxima linha deste conjunto de informações indica que o objeto é baixo e o eixo z é alinhado à direção da menor dimensão da nuvem de pontos. Os próximos indicadores a_maior , a_int e a_menor configuram os escalares de tamanho em ordem decrescente, enquanto a direção do eixo de inércia principal é definida pelo autovetor v_1 . Finalmente, a estimativa para o tamanho do modelo é atribuída corretamente aos parâmetros $p[0]$, $p[1]$, $p[2]$ a partir das variáveis $a1 = 15$, $a2 = 10$ e $a3 = 6$. Esses escalares representam os termos a_1 , a_2 e a_3 da equação (3.26). A identificação dos quadrantes originais para cada um dos ângulos que expressam a orientação da nuvem de pontos é exibida pelas variáveis **PHI**, **THETA** e **PSI**.

O encerramento de execução do programa é caracterizado por oito iterações da implementação *levmar* definida pela opção de interrupção três (quantidade máxima de iterações completada). Em contrapartida, a linha seguinte exhibe os resultados otimizados para os onze parâmetros da equação (3.26). Especificamente:

$p[0] = 15,3$; $p[1] = 10,2$; $p[2] = 6,12$; para o tamanho da superelipsóide.

$p[3] = 0,1$; $p[4] = 0,1$; para os fatores de forma do modelo final.

$p[5] = 0,0$; $p[6] = 3,14$; $p[7] = 0,0$; para a orientação designada por ϕ , θ e ψ , respectivamente.

$p[8] = -8,0$; $p[9] = -5,0$ e $p[10] = 9,0$; para as coordenadas do baricentro.

Esses resultados demonstram que a melhor superelipsóide capaz de representar a nuvem de pontos da caixa irregular é modelada como uma forma geométrica de bordas retas ($\varepsilon_1 = \varepsilon_2 = 0,1$) com dimensões iguais a 15, 10 e 6 nas direções dos eixos coordenados x , y e z , respectivamente.

O elevado erro (41,43) para o ajuste inicial da modelagem é obtido do cálculo da função objetivo da equação (3.37) considerando os valores iniciais dos 11 parâmetros do modelo ($a_1 = 15$, $a_2 = 10$, $a_3 = 6$, $\varepsilon_1 = \varepsilon_2 = 1,0$, $\phi = 0^\circ$, $\theta = 90,04^\circ$, $\psi = 0^\circ$, $p_x = -8$, $p_y = -5$, $p_z = 9$). Em contrapartida, a reduzida distorção (0,00030) entre o objeto original e o modelo descrito pela superelipsóide, é resultante dos valores otimizados de tais parâmetros ($a_1 = 15,3$, $a_2 = 10,2$, $a_3 = 6,12$, $\varepsilon_1 = \varepsilon_2 = 0,1$, $\phi = 0^\circ$, $\theta = 0,0^\circ$, $\psi = 0^\circ$, $p_x = -8$, $p_y = -5$, $p_z = 9$).

Informações adicionais complementam o elenco de resultados iniciando com a aproximação mostrada na equação (3.44) para $\mathbf{J}^T \cdot \epsilon = 0,000428$ e $\delta_p = 2,0e-8$ para o escalar que descreve a solução da otimização por mínimos quadrados exibido na equação (3.42). O reduzido valor igual a $2,88e-5$ expressa o resultado da divisão do termo de amortecimento pelo máximo valor de $\mathbf{J}^T \cdot \mathbf{J}$, necessário para o cálculo dos elementos da diagonal da matriz \mathbf{N} que representa a aproximação da Jacobiana da seção 4.0. Esses valores representam os melhores resultados para os termos do método de minimização, calculados para cada uma das estimativas propostas pelo procedimento de otimização. Na seqüência, o escalar oito indica a quantidade de iterações calculadas, enquanto o motivo de interrupção da rotina é designado pela opção três (quantidade máxima de iterações executada). O volume de repetições envolvendo a relação funcional f da seção mencionada é indicado pelo valor 105, o qual envolveu oito cálculos das matrizes de Jacobi e oito tentativas para reduzir o erro de ajuste descrito por igual quantidade de sistemas lineares calculados. Finalmente, três segundos foi o tempo envolvido na execução de todo o programa.

4.1.12 Verificação do erro do ajuste

A finalização do processo de busca pelo melhor modelo é determinada pelo erro do ajuste inferior a 0,005. Em particular, uma instrução condicional direciona o

processamento de um novo conjunto de parâmetros candidatos no caso de uma distorção superior à tolerância máxima permitida. Em contrapartida, o fluxo de execução do programa segue para a exibição da superfície resultante, em programa independente, quando esta diferença for inferior ao critério numérico de ajuste.

4.2 Programa gráfico

Embora o programa numérico discutido nas subseções anteriores seja responsável pela modelagem das nuvens de pontos, ele não contempla os procedimentos computacionais necessários para a visualização dos modelos resultantes. Como consequência, foi necessário desenvolver o estágio de apresentação mostrado na figura 4.1. Inspirado no algoritmo desenvolvido por Cieslak (2010) algumas bibliotecas gráficas da OpenGL são as responsáveis pela exibição das nuvens de pontos e dos modelos volumétricos sugeridos pelo método de minimização.

Uma vez que este programa é utilizado como ferramenta de visualização para a interpretação dos resultados simulados, uma breve descrição dos aspectos mais relevantes deste código é discutida nas subseções que encerram o capítulo.

4.2.1 Declaração da expressão paramétrica

Para toda programação envolvendo superelipsóides, deve-se investir particular atenção no cálculo numérico das equações que descrevem tais superfícies. De fato, todos os termos exponenciais da função explícita da equação (3.9) são da forma x^ε , na qual ε pertence ao conjunto dos números reais positivos. Em computação numérica é fundamental garantir o cálculo correto desses termos para assegurar que o resultado não seja um valor descrito pelo conjunto dos números complexos quando x for um valor real negativo.

Desse modo, é adotado neste trabalho que toda função exponencial representa, de fato, uma função computacional definida como:

$$x^\varepsilon = \text{sign}(x) \cdot |x|^\varepsilon = \begin{cases} +|x|^\varepsilon, & \text{para } x \geq 0 \\ -|x|^\varepsilon, & \text{para } x < 0 \end{cases} \quad (4.1)$$

Além do mais, para evitar *overflow* numérico e dificuldades com singularidades, os fatores de forma foram restritos ao intervalo $(0, 1 \leq \{\varepsilon_1, \varepsilon_2\} \leq 1, 0)$. Frequentemente esses detalhes são omitidos nos artigos relacionados à síntese, modelagem e reconstrução de objetos empregando técnicas de ajuste por mínimos quadrados. Todavia, tais aspectos são fundamentais para o sucesso da implementação computacional. Como consequência, as linhas de código abaixo expõem as declarações das funções trigonométricas da expressão paramétrica da equação (3.9).

```
// Declaração da função cos(v,n) = sgnf(cos(v)) * |cos(v)|^n
float sqCos ( float v, float n ) {
    return sgnf((float)cos(v)) * float powf(absf((float)cos(v)),n); }

// Declaração da função sen(v,n) = sgnf(sen(v)) * |sen(v)|^n
float sqSen ( float v, float n ) {
    return sgnf((float)sin(v)) * float powf(absf((float)sin(v)),n); }
```

Neste segmento de código mostrado, as funções **sgnf** e **absf** são programadas como:

```
/* Retorno do sinal de x */
float sgnf ( float x ) {
    if ( x < 0 ) return -1;
    if ( x > 0 ) return 1;
    return 0; }

/* Retorno do valor absoluto de x */
float absf ( float x ) {
    if ( x < 0 ) return -x;
    return x; }
```

Todavia, as linhas geométricas responsáveis pela composição das malhas de modelagem das superelipsóides revelaram uma inaceitável distribuição irregular ao longo da superfície de tais modelos. Em particular, grandes concentrações dessas linhas se manifestam nas regiões de maior curvatura dos objetos. Uma vez que o propósito deste trabalho tem seu foco direcionado à manipulação de objetos, este comportamento indesejável foi combatido pela parametrização uniforme estudada na próxima subseção.

4.2.2 Parametrização regular da superfície da superelipsóide

Para atender aos requisitos do projeto foi necessário promover uma distribuição uniforme dos pontos ou linhas que modelam a superfície do modelo calculado pelo método de minimização. A proposta utilizada foi inspirada na abordagem apresentada por Bardinet *et al.* (1995) para modelar algumas anatomias humanas.

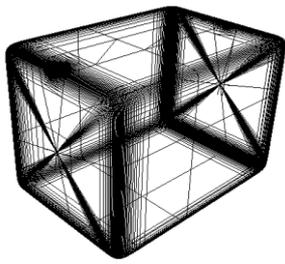
$$\rho = \left[\left(|\cos(\omega) \cdot \cos(\eta)|^{\frac{2}{\varepsilon_2}} \right) + \left(|\sen(\omega) \cdot \cos(\eta)|^{\frac{2}{\varepsilon_2}} \right)^{\frac{\varepsilon_2}{\varepsilon_1}} + |\sen(\eta)|^{\frac{2}{\varepsilon_1}} \right]^{-\frac{\varepsilon_1}{2}}, \quad \begin{matrix} -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \omega \leq \pi \end{matrix} \quad (4.2)$$

Como consequência, as coordenadas locais dos pontos que formam a malha de modelagem assumiram a seguinte notação:

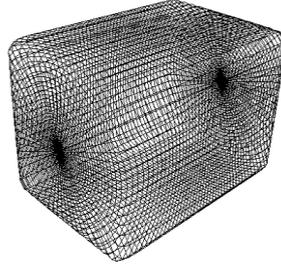
$$\begin{aligned} x &= \rho \cdot a_1 \cdot \cos(\eta) \cdot \cos(\omega), & -\pi/2 \leq \eta \leq \pi/2 \\ y &= \rho \cdot a_2 \cdot \cos(\eta) \cdot \sen(\omega), & -\pi \leq \omega \leq \pi \\ z &= \rho \cdot a_3 \cdot \sen(\eta) \end{aligned} \quad (4.3)$$

A figura 4.4(a) mostra a superfície de uma superelipsóide modelada com arranjo irregular das linhas geométricas. Como pode ser observada nesta figura, uma grande concentração de linhas se manifesta nas bordas do modelo, resultando em regiões carentes dessas entidades no restante do volume. Em contrapartida, a modelagem da figura 4.4(b), proporcionada por uma superfície com linhas distribuídas uniformemente ao longo do volume, caracteriza a parametrização regular utilizada.

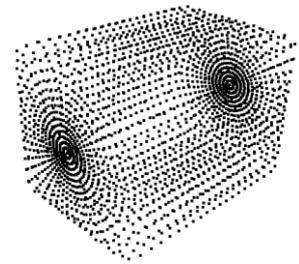
A representação alternativa por pontos da figura 4.4(c) utiliza a mesma modalidade de parametrização uniforme da figura 4.4(b) e, apesar da semelhança visual com as nuvens de pontos consideradas na modelagem de objetos, é importante enfatizar que independente da quantidade de pontos da figura 4.4(c) o modelo resultante sempre é representado por apenas cinco escalares.



(a) Parametrização dependente da curvatura.



(b) Parametrização uniforme representada por linhas.



(c) Parametrização uniforme caracterizada por pontos.

Figura 4.4 - Parametrização da malha da superfície de modelagem das superelipsóides. (a) grande concentração de linhas ocorre próximo às bordas do modelo. (b) parametrização regular das linhas e (c) distribuição uniforme representada por pontos.

Análogo aos aspectos matemáticos discutidos na subseção anterior, o cálculo das funções exponenciais do fator de parametrização, ρ , empregou a notação computacional estudada nesta seção. O desempenho e a eficiência do programa computacional desenvolvido são demonstrados nas simulações conduzidas no capítulo V.

Modelagem automática de nuvem de pontos

Para demonstrar o desempenho e a eficiência do programa computacional desenvolvido, diversas simulações envolvendo um elenco de primitivas composto por esferas, cubos, paralelepípedos e cilindros sintéticos, organizadas em diferentes configurações, serão conduzidas neste capítulo.

No entanto, com o objetivo de simplificar a interpretação dos resultados obtidos na representação volumétrica de tais formas elementares, a seção 5.1 introduz uma discussão sistemática da modelagem de um paralelepípedo.

5.1 Exemplo numérico

A caixa irregular da Figura 5.1 com dimensões iguais a 15 cm, 10 cm e 6 cm ao longo dos eixos coordenados x , y e z , respectivamente, tem o seu centro de massa localizado na posição $(-8,0 \ -5,0 \ 9,0)$ cm em relação à origem do sistema de coordenadas global. Além disso, a orientação deste objeto foi definida por uma rotação de 0° em torno dos eixos x e z e 90° em relação ao eixo y .

Os dados da simulação de todos os objetos sintéticos foram gerados no ambiente gráfico do *software* comercial *Autodesk 3ds Max*. Para representar cada um desses objetos uma nuvem de pontos com 2402 coordenadas também foi produzida nesta ferramenta. Por conveniência, a sequência do processo de modelagem segue aquela definida no capítulo anterior para os procedimentos computacionais ilustrados no fluxograma da Figura 4.1.

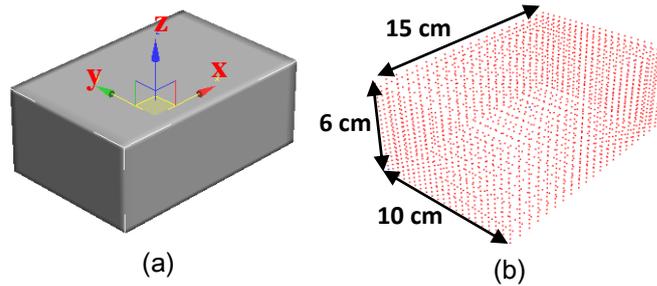


Figura 5.1 – (a) paralelepípedo e o sistema de coordenadas original; (b) nuvem de pontos correspondente ao objeto.

5.1.1 Estimativa da posição da nuvem de pontos

As coordenadas calculadas para o baricentro que representa a posição da nuvem de pontos em relação ao sistema de coordenadas global são iguais a $(-8,0 -5,0 9,0)$ cm, em relação aos eixos coordenados x , y e z , respectivamente. Quando comparados aos escalares que descrevem a posição exata do objeto este resultado se apresenta perfeitamente consistente.

5.1.2 Estimativa da orientação da nuvem de pontos

Conforme mencionado na seção 3.5.2 a orientação da nuvem de pontos é determinada por intermédio dos elementos da matriz de covariância, a partir do cálculo de seus autovetores. Desse modo, a informação proporcionada pelo baricentro da nuvem de pontos foi suficiente para a obtenção dos elementos da matriz C .

$$C = \begin{bmatrix} 125,33 & 0,00 & 0,00 \\ 0,00 & 55,70 & 0,00 \\ 0,00 & 0,00 & 20,05 \end{bmatrix} \quad (5.1)$$

Como o objeto da figura 5.1 não foi submetido a rotações em relação aos eixos coordenados x e y , uma análise desta matriz a caracteriza como diagonal. Todavia, objetos com orientações diferentes daquela da caixa irregular resultam em valores não nulos para os demais termos desta matriz. De qualquer modo, a rotina responsável pelo cálculo do auto-sistema da equação (5.1) produziu os resultados da tabela 5.1.

Tabela 5.1 – Autovalores e autovetores para o paralelepípedo do exemplo numérico.

Autovalores	Autovetores normalizados
$\lambda_1 = 20,05$	$v_1 = [0 \ 0 \ 1]$
$\lambda_2 = 55,70$	$v_2 = [0 \ 1 \ 0]$
$\lambda_3 = 125,33$	$v_3 = [1 \ 0 \ 0]$

A partir desses autovetores a relação expressa pela equação (4.4) permitiu estimar a orientação da nuvem de pontos quando centrada na origem do sistema de coordenadas local.

$$\begin{aligned}
 \phi &= \arccos(v_{3,3}) = \arccos(0) \rightarrow \phi = 90^\circ \\
 \theta &= \arctan\left(\frac{v_{2,3}}{v_{1,3}}\right) = \arctan\left(\frac{0}{1}\right) \rightarrow \theta = 0^\circ \\
 \psi &= \arctan\left(\frac{v_{3,2}}{-v_{3,1}}\right) = \arctan\left(\frac{0}{-1}\right) \rightarrow \psi = 0^\circ
 \end{aligned} \tag{5.2}$$

Estes ângulos estão coerentes com a orientação da caixa irregular descrita por uma rotação de 90° em relação ao eixo z original seguida por rotações consecutivas de 0° em torno dos novos eixos coordenados y e z , respectivamente.

5.1.3 Determinação do eixo de inércia principal da superelipsóide

Por intermédio dos autovalores a interpretação do critério estabelecido na subseção 3.5.2 sugere que a direção do eixo de inércia principal da nuvem de pontos corresponda àquela mostrada na figura 5.2.

A partir da condição enunciada $|\lambda_1 - \lambda_2| < |\lambda_2 - \lambda_3|$ os valores mostrados na tabela 5.1 confirmam a comparação expressa por $|20,05 - 55,70| < |55,70 - 125,33|$. Como este resultado define a direção do eixo de inércia principal a partir das coordenadas do autovetor v_3 , a direção do eixo z do novo sistema de coordenadas é mostrada na figura 5.2.

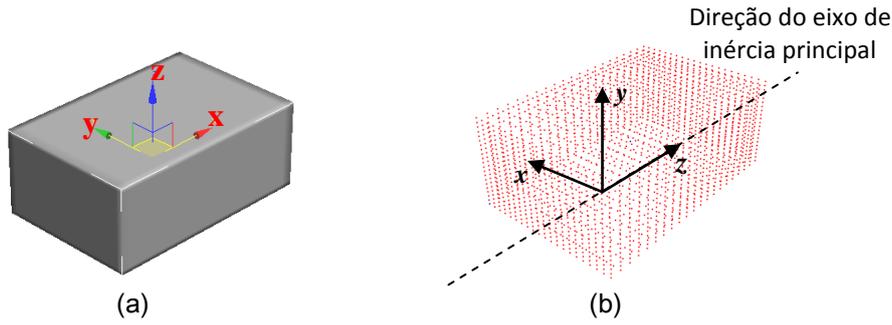


Figura 5.2 - Interpretação geométrica da direção do eixo de inércia principal da nuvem de pontos: (a) objeto com sistema de coordenadas original; (b) direção do eixo de inércia indicada pelo eixo z do novo sistema de coordenadas.

5.1.4 Estimativa do tamanho da nuvem de pontos

De acordo com a relação mostrada na equação (3.12) as estimativas das extensões da nuvem de pontos na direção dos eixos coordenados x , y e z são apresentadas na equação (5.3).

$$\begin{aligned}
 a_1 &= \max_{j=1}^{2402} \{ [P_j - (-8,00 \ -5,00 \ 9,00)]. [0 \ 0 \ 1] \} \rightarrow a_1 = 15,0 \text{ cm} \\
 a_2 &= \max_{j=1}^{2402} \{ [P_j - (-8,00 \ -5,00 \ 9,00)]. [0 \ 1 \ 0] \} \rightarrow a_2 = 10,0 \text{ cm} \\
 a_3 &= \max_{j=1}^{2402} \{ [P_j - (-8,00 \ -5,00 \ 9,00)]. [1 \ 0 \ 0] \} \rightarrow a_3 = 6,0 \text{ cm}
 \end{aligned} \quad (5.3)$$

Como se pode observar, estes escalares descrevem o tamanho exato do objeto ilustrado na figura 5.1.

Finalmente, todos os resultados obtidos nas subseções anteriores foram combinados para compor a função objetivo da equação (4.6). Em particular, os parâmetros iniciais para o tamanho $a_1 = 15$ cm, $a_2 = 10$ cm, $a_3 = 6$ cm; forma $\varepsilon_1 = 1,0$, $\varepsilon_2 = 1,0$; orientação $\theta = 0^\circ$, $\phi = 90^\circ$, $\psi = 0^\circ$ e posição $p_x = -8$ cm, $p_y = -5$ cm e $p_z = 9$ cm, substituem os termos da equação (5.4)

$$d = \sum_{i=1}^{2402} \left[F(a_1, a_2, a_3, \varepsilon_1, \varepsilon_2, \theta, \phi, \psi, p_x, p_y, p_z, X, Y, Z)^{\frac{\varepsilon_1}{2}} - 1 \right]^2 \quad (5.4)$$

5.2. Minimização da função objetivo empregando o método de Levenberg-Marquardt

A tabela 5.2 resume o conjunto de resultados produzido pelo programa computacional a partir da minimização da função objetivo da equação (5.4).

Tabela 5.2 – Resumo dos resultados obtidos para o paralelepípedo do exemplo numérico.

Orientação exata	Orientação estimada	Posição exata (cm)	Posição estimada (cm)	Parâmetro	Valores exatos (cm)	Valores Iniciais (cm)	Valores Estimados (cm)	
$\theta = 0,00^\circ$	$\theta = 0,00^\circ$	$p_x = -8,0$	$p_x = -8,0$	a_1	15,0	15,0	15,3	
$\phi = 90,00^\circ$	$\phi = 90,04^\circ$	$p_y = -5,0$	$p_y = -5,0$	a_2	10,0	10,0	10,2	
$\psi = 0,00^\circ$	$\psi = 0,00^\circ$	$p_z = 9,0$	$p_z = 9,0$	a_3	6,0	6,0	6,1	
					ε_1	0,10	1,00	0,10
					ε_2	0,10	1,00	0,11
Erro do ajuste inicial = 41,43								
Erro do ajuste final (EOF) = 0,0003								
Quantidade de iterações = 5								
Tempo de processamento = 2 segundos								

O elevado erro do ajuste inicial representa o mínimo valor da função objetivo obtida para a elipsóide ($\varepsilon_1 = \varepsilon_2 = 1,0$) do primeiro modelo candidato (ver figura 5.3(c)) a simular a nuvem de pontos da figura 5.3(b). Em contrapartida, a pequena distorção entre o paralelepípedo da figura 5.3(a) e a superelipsóide da figura 5.3(h) é revelada pelo reduzido erro de ajuste final (EOF = 0,0003), obtido com cinco iterações computacionais conduzidas em apenas dois segundos. A seqüência das figuras 5.3 (d), (e), (f) e (g) ilustra a evolução do processo de ajuste com subseqüentes reduções na diferença entre a forma volumétrica esperada para aquela definida pelo objeto original.

Com o objetivo de validar o algoritmo de ajuste, a tabela 5.3 ilustra o eixo de inércia principal estimado pelo programa computacional para cada uma das oito primitivas simuladas neste capítulo. Para a conveniência de visualização, modelos volumétricos originais são exibidos na tabela para facilitar a interpretação dos resultados, embora as 2402 coordenadas de cada nuvem de pontos tenham sido utilizadas no processo de modelagem.

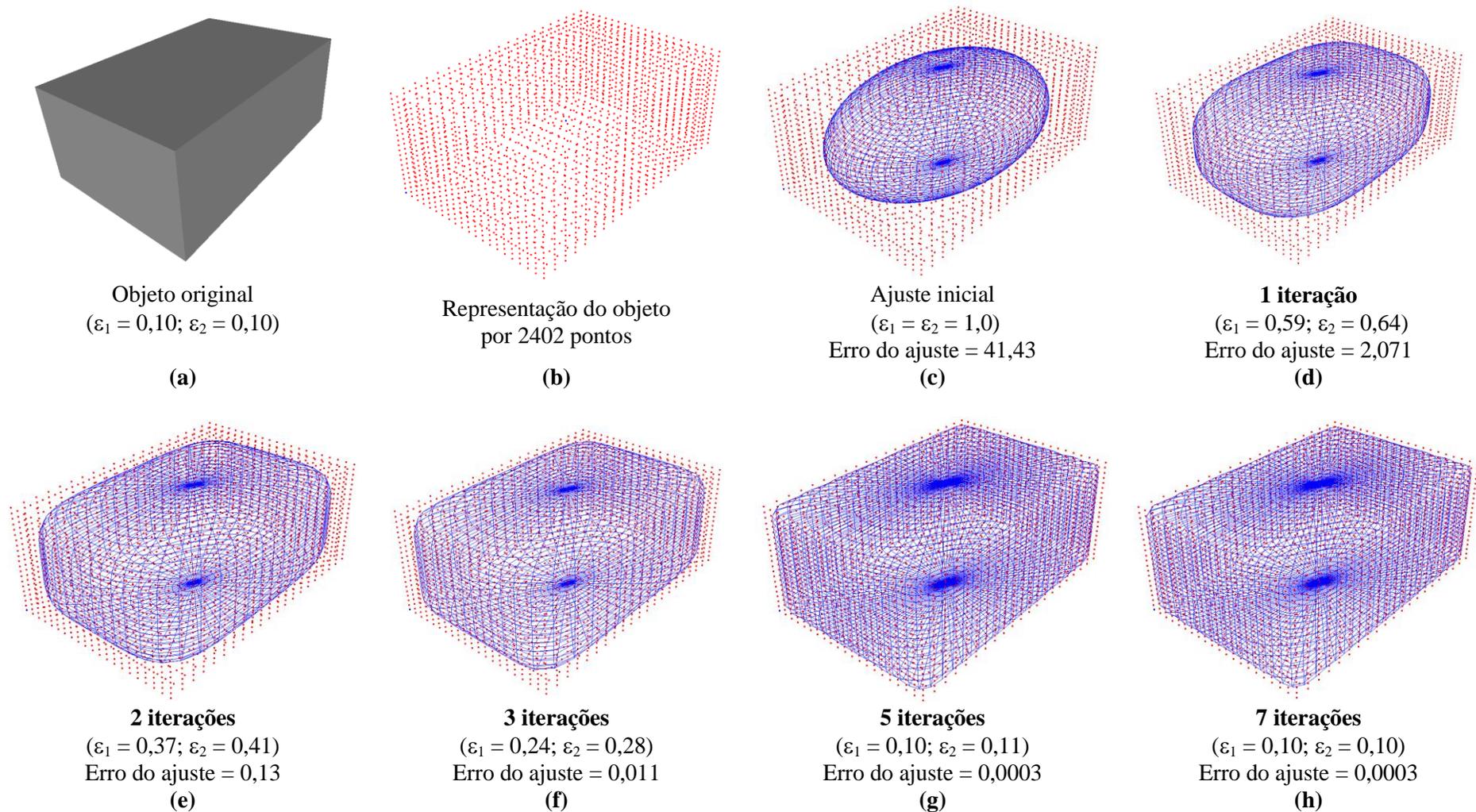
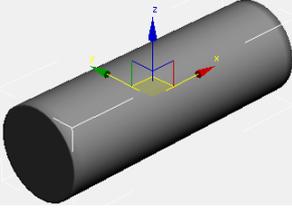
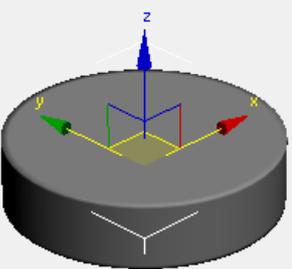
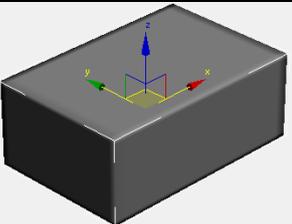
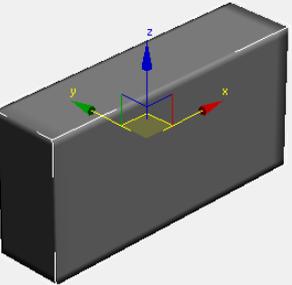
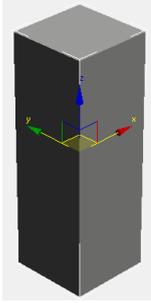
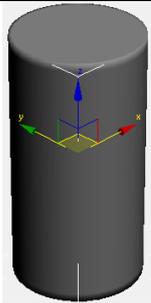


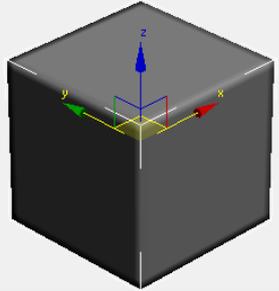
Figura 5.3 - Evolução do ajuste de uma superelipsóide a um paralelepípedo: (a) objeto sintético; (b) nuvem de pontos da caixa irregular; (c) superelipsóide inicial; modelo volumétrico após uma (d), duas (e) e três (f) iterações; (g) e (h) melhores modelos para cinco e sete iterações, respectivamente.

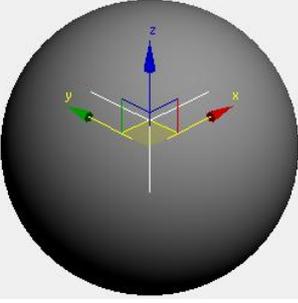
Tabela 5.3 - Estimativa da orientação do eixo de inércia principal das primitivas.

Objeto	Autovalores	Auto-vetores normalizados	Estimada da direção do eixo de inércia principal
 <p>(a) Cilindro tombado *</p>	$\lambda_1 = 7,74$	$v_1 = [0,00 \ 0,96 \ 0,26]$	<i>Eixo x</i>
	$\lambda_2 = 7,74$	$v_2 = [0,00 \ 0,26 \ -0,96]$	
	$\lambda_3 = 161,74$	$v_3 = [-1,00 \ 0,00 \ 0,00]$	
 <p>(b) Cilindro baixo *</p>	$\lambda_1 = 19,74$	$v_1 = [0,00 \ 0,00 \ 1,00]$	<i>Eixo z</i>
	$\lambda_2 = 109,97$	$v_2 = [0,11 \ -0,99 \ 0,00]$	
	$\lambda_3 = 109,97$	$v_3 = [-0,99 \ -0,11 \ 0,00]$	
 <p>(c) Caixa tombada *</p>	$\lambda_1 = 20,05$	$v_1 = [0,00 \ 0,00 \ 1,00]$	<i>Eixo x</i>
	$\lambda_2 = 55,70$	$v_2 = [0,00 \ 1,00 \ 0,00]$	
	$\lambda_3 = 125,33$	$v_3 = [1,00 \ 0,00 \ 0,00]$	
 <p>(d) Caixa larga tombada *</p>	$\lambda_1 = 13,93$	$v_1 = [0,00 \ 1,00 \ 0,00]$	<i>Eixo x</i>
	$\lambda_2 = 55,70$	$v_2 = [0,00 \ 0,00 \ 1,00]$	
	$\lambda_3 = 222,81$	$v_3 = [1,00 \ 0,00 \ 0,00]$	

 <p>(e) Caixa alta*</p>			<i>Eixo z</i>
	$\lambda_1 = 13,99$	$v_1 = [-1,00 \ 0,00 \ 0,00]$	
	$\lambda_2 = 16,57$	$v_2 = [0,00 \ -1,00 \ 0,00]$	
	$\lambda_3 = 101,31$	$v_3 = [0,00 \ 0,00 \ 1,00]$	

 <p>(f) Cilindro alto*</p>			<i>Eixo z</i>
	$\lambda_1 = 30,71$	$v_1 = [-0,13 \ 0,99 \ 0,00]$	
	$\lambda_2 = 30,71$	$v_2 = [0,99 \ 0,13 \ 0,00]$	
	$\lambda_3 = 289,38$	$v_3 = [0,00 \ 0,00 \ 1,00]$	

 <p>(g) Cubo*</p>			<i>Eixo x</i>
	$\lambda_1 = 348,14$	$v_1 = [1,00 \ 0,00 \ 0,00]$	
	$\lambda_2 = 348,14$	$v_2 = [0,00 \ 1,00 \ 0,00]$	
	$\lambda_3 = 348,14$	$v_3 = [0,00 \ 0,00 \ 1,00]$	

 <p>(h) Esfera*</p>			<i>Eixo z</i>
	$\lambda_1 = 102,77$	$v_1 = [0,00 \ 0,99 \ 0,00]$	
	$\lambda_2 = 102,77$	$v_2 = [0,99 \ 0,00 \ 0,00]$	
	$\lambda_3 = 194,45$	$v_3 = [0,00 \ 0,00 \ 1,00]$	

A interpretação desses resultados segue o que foi previamente discutido no caso da caixa irregular do exemplo numérico, ou seja, se $|\lambda_1 - \lambda_2| < |\lambda_2 - \lambda_3|$ a direção do eixo de inércia principal é definida pelo autovetor v_3 , enquanto o autovetor v_1 indicará a direção deste eixo em caso contrário.

A direção do eixo de inércia principal para as primitivas da tabela 5.3 (a), (c), e (d) é determinada pelo eixo x do sistema de orientação original, resultante do módulo da maior coordenada do autovetor v_3 . Em contrapartida, o módulo do maior escalar do autovetor v_1 foi o responsável pela direção do eixo de inércia ao longo dos eixos z e x para o cilindro baixo e cubo das tabelas 5.3 (b) e (g), respectivamente. A direção estabelecida pelo terceiro autovetor para o eixo de inércia ao longo do eixo z para a caixa alta (ver tabela 5.3(e)), cilindro alto (ver tabela 5.3(f)) e esfera (ver tabela 5.3(h)) também é ilustrada nesta tabela.

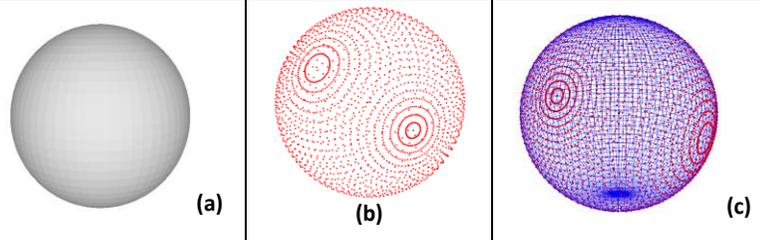
Em razão dos arredondamentos numéricos da linguagem de programação C/C++, a direção do eixo de inércia para o cubo e a esfera das tabelas 5.3 (g) e (f), respectivamente, poderia diferir dos resultados obtidos devido à natureza simétrica dessas primitivas em torno dos eixos coordenados. Todavia, esse comportamento não influenciou na obtenção de resultados coerentes com a geometria dos objetos investigados.

Embora a esfera da tabela 5.3(h) seja virtualmente simétrica, é pertinente observar que a diferença entre os autovalores (λ_1, λ_2) e λ_3 desse objeto indica que não houve uma distribuição espacial regular para todos os 2402 pontos. Uma observação mais detalhada da nuvem de pontos evidenciou maior concentração de pontos nos pólos deste objeto, contribuindo para a distorção numérica indicada pelos autovalores.

A seqüência das tabelas 5.4 até 5.10 estendem a aplicação do algoritmo de ajuste na simulação de quatro primitivas de interesse para os propósitos deste trabalho. Além disso, os objetos modelados foram configurados em diferentes posições e orientações.

Para a conveniência da interpretação dos resultados, tais tabelas exibem o objeto original, sua representação por uma nuvem de pontos e a modelagem baseada em uma superelipsóide sobreposta às coordenadas dos pontos.

Tabela 5.4 - Superelipsóide ajustada à uma esfera.

Objeto	Nuvem de pontos do objeto	Superelipsóide ajustada à nuvem de pontos	Quantidade de pontos	Parâmetros	Valor exato	Valores estimados pelo método
						pela minimização da função objetivo: $d(\Lambda, x, y, z) = [F(\Lambda, x, y, z)^{\varepsilon_1/2} - 1]^2$
			2402	a_1	20,00 cm	20,03 cm
				a_2	20,00 cm	20,88 cm
				a_3	20,00 cm	19,99 cm
				ε_1	1,00	1,00
				ε_2	1,00	0,98
Pose do objeto						
Orientação exata	$\phi = 180,00^\circ$	$\theta = 0,00^\circ$	$\psi = -90,00^\circ$	⁽¹⁾ Erro do ajuste inicial = 67,17		
Orientação estimada	$\phi = 145,96^\circ$	$\theta = 0,00^\circ$	$\psi = -124,32^\circ$	⁽²⁾ Erro do ajuste final = 0,0031		
* Posição exata	$p_x = 0,00$ cm	$p_y = 0,00$ cm	$p_z = 0,00$ cm	⁽³⁾ Quantidade de iterações: 7		
* Posição estimada	$p_x = 0,00$ cm	$p_y = 0,00$ cm	$p_z = 0,00$ cm	⁽⁴⁾ Tempo de processamento = 2 segundos		

⁽¹⁾ Erro do ajuste inicial: Função objetivo calculada com os parâmetros iniciais.

⁽²⁾ Erro do ajuste final: Função objetivo calculada com os parâmetros finais estimados pelo método de minimização.

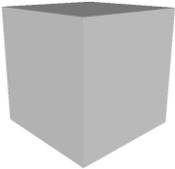
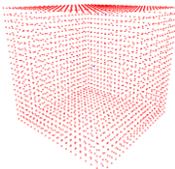
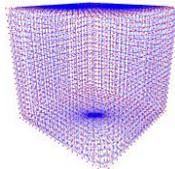
⁽³⁾ Quantidade de iterações utilizada pelo método para determinar os parâmetros da superelipsóide que melhor se ajustou à nuvem de pontos.

⁽⁴⁾ Tempo de processamento do programa principal estimado pela instrução *difftime*(tempo_final, tempo_inicial) nativa da linguagem C++.

*Posição: coordenadas do vetor posição em relação à origem do sistema de coordenadas global.

Observação: Os parâmetros iniciais $a_1, a_2, a_3, \varepsilon_1 = 1,0$ e $\varepsilon_2 = 1,0$ foram fornecidos automaticamente ao método de minimização.

Tabela 5.5 - Superelipsóide ajustada a um cubo.

Objeto	Nuvem de pontos do objeto	Superelipsóide ajustada à nuvem de pontos	Quantidade de pontos	Parâmetros	Valor exato	Valores estimados pelo método
						pela minimização da função objetivo: $d(\Lambda, x, y, z) = [F(\Lambda, x, y, z)^{\varepsilon_1/2} - 1]^2$
 (a)	 (b)	 (c)	2402	a_1	25,00 cm	25,07 cm
				a_2	25,00 cm	25,12 cm
				a_3	25,00 cm	25,00 cm
				ε_1	0,10	0,10
				ε_2	0,10	0,10
Pose do objeto						
Orientação exata	$\phi = 0,00^\circ$	$\theta = 0,00^\circ$	$\psi = 0,00^\circ$	⁽¹⁾ Erro do ajuste inicial = 48,61		
Orientação estimada	$\phi = 0,00^\circ$	$\theta = 0,00^\circ$	$\psi = 0,00^\circ$	⁽²⁾ Erro do ajuste final = 0,00034		
* Posição exata	$p_x = 0,00$ cm	$p_y = 25,00$ cm	$p_z = 0,0$ cm	⁽³⁾ Quantidade de iterações: 7		
* Posição estimada	$p_x = 0,00$ cm	$p_y = 25,00$ cm	$p_z = 0,0$ cm	⁽⁴⁾ Tempo de processamento = 3 segundos		

⁽¹⁾ Erro do ajuste inicial: Função objetivo calculada com os parâmetros iniciais.

⁽²⁾ Erro do ajuste final: Função objetivo calculada com os parâmetros finais estimados pelo método de minimização.

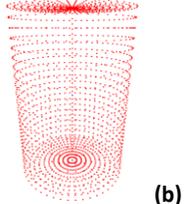
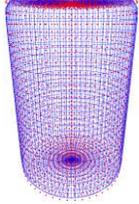
⁽³⁾ Quantidade de iterações utilizada pelo método para determinar os parâmetros da superelipsóide que melhor se ajustou à nuvem de pontos.

⁽⁴⁾ Tempo de processamento do programa principal estimado pela instrução `difftime(tempo_final, tempo_inicial)` nativa da linguagem C++.

*Posição: coordenadas do vetor posição em relação à origem do sistema de coordenadas global.

Observação: Os parâmetros iniciais $a_1, a_2, a_3, \varepsilon_1 = 1,0$ e $\varepsilon_2 = 1,0$ foram fornecidos automaticamente ao método de minimização.

Tabela 5.6 - Superelipsóide ajustada à um cilindro alto.

Objeto	Nuvem de pontos do objeto	Superelipsóide ajustada à nuvem de pontos	Quantidade de pontos	Parâmetros	Valor exato	Valores estimados pelo método
						pela minimização da função objetivo: $d(\Lambda, x, y, z) = [F(\Lambda, x, y, z)^{\varepsilon_1/2} - 1]^2$
 (a)	 (b)	 (c)	2402	a_1	10,00 cm	10,04 cm
				a_2	10,00 cm	10,45 cm
				a_3	20,00 cm	20,00 cm
				ε_1	0,10	0,10
				ε_2	1,00	1,00
Pose do objeto						
Orientação exata	$\phi = -180,00^\circ$	$\theta = 0,00^\circ$	$\psi = -90,00^\circ$	⁽¹⁾ Erro do ajuste inicial = 36,62		
Orientação estimada	$\phi = -172,11^\circ$	$\theta = 0,00^\circ$	$\psi = -90,04^\circ$	⁽²⁾ Erro do ajuste final = 0,00037		
* Posição exata	$p_x = 0,00$ cm	$p_y = 0,00$ cm	$p_z = 20,0$ cm	⁽³⁾ Quantidade de iterações: 8		
* Posição estimada	$p_x = 0,00$ cm	$p_y = 0,00$ cm	$p_z = 20,0$ cm	⁽⁴⁾ Tempo de processamento = 2 segundos		

⁽¹⁾ Erro do ajuste inicial: Função objetivo calculada com os parâmetros iniciais.

⁽²⁾ Erro do ajuste final: Função objetivo calculada com os parâmetros finais estimados pelo método de minimização.

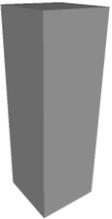
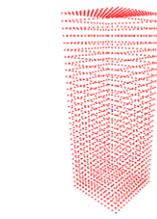
⁽³⁾ Quantidade de iterações utilizada pelo método para determinar os parâmetros da superelipsóide que melhor se ajustou à nuvem de pontos.

⁽⁴⁾ Tempo de processamento do programa principal estimado pela instrução *difftime*(tempo_final, tempo_inicial) nativa da linguagem C++.

*Posição: coordenadas do vetor posição em relação à origem do sistema de coordenadas global.

Observação: Os parâmetros iniciais $a_1, a_2, a_3, \varepsilon_1 = 1,0$ e $\varepsilon_2 = 1,0$ foram fornecidos automaticamente ao método de minimização.

Tabela 5.7 - Superelipsóide ajustada à um paralelepípedo.

Objeto	Nuvem de pontos do objeto	Superelipsóide ajustada à nuvem de pontos	Quantidade de pontos	Parâmetros	Valor exato	Valores estimados pelo método
						pela minimização da função objetivo: $d(\Lambda, x, y, z) = [F(\Lambda, x, y, z)^{\epsilon_1/2} - 1]^2$
 (a)	 (b)	 (c)	2402	a_1	5,00 cm	5,00 cm
				a_2	5,00 cm	5,05 cm
				a_3	15,00 cm	15,11 cm
				ϵ_1	0,10	0,10
				ϵ_2	0,10	0,10
Pose do objeto						
Orientação exata	$\phi = -90,00^\circ$	$\theta = 0,00^\circ$	$\psi = 90,00^\circ$	⁽¹⁾ Erro do ajuste inicial = 40,04		
Orientação estimada	$\phi = -86,90^\circ$	$\theta = 0,00^\circ$	$\psi = 86,90^\circ$	⁽²⁾ Erro do ajuste final = 0,0017		
* Posição exata	$p_x = 0,00$ cm	$p_y = 0,00$ cm	$p_z = 100$ cm	⁽³⁾ Quantidade de iterações: 8		
* Posição estimada	$p_x = 0,00$ cm	$p_y = 0,00$ cm	$p_z = 100$ cm	⁽⁴⁾ Tempo de processamento = 3 segundos		

⁽¹⁾ Erro do ajuste inicial: Função objetivo calculada com os parâmetros iniciais.

⁽²⁾ Erro do ajuste final: Função objetivo calculada com os parâmetros finais estimados pelo método de minimização.

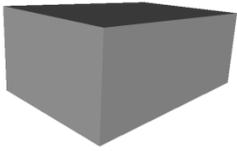
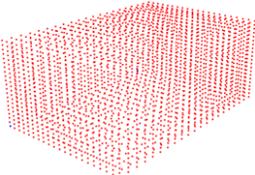
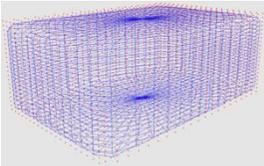
⁽³⁾ Quantidade de iterações utilizada pelo método para determinar os parâmetros da superelipsóide que melhor se ajustou à nuvem de pontos.

⁽⁴⁾ Tempo de processamento do programa principal estimado pela instrução *difftime*(tempo_final, tempo_inicial) nativa da linguagem C++.

*Posição: coordenadas do vetor posição em relação à origem do sistema de coordenadas global.

Observação: Os parâmetros iniciais $a_1, a_2, a_3, \epsilon_1 = 1,0$ e $\epsilon_2 = 1,0$ foram fornecidos automaticamente ao método de minimização.

Tabela 5.8 - Superelipsóide ajustada à um paralelepípedo tombado.

Objeto	Nuvem de pontos do objeto	Superelipsóide ajustada à nuvem de pontos	Quantidade de pontos	Parâmetros	Valores estimados pelo método pela minimização da função objetivo: $d(\Lambda, x, y, z) = [F(\Lambda, x, y, z)^{\epsilon_1/2} - 1]^2$	
					Valor exato	
 <p>(a)</p>	 <p>(b)</p>	 <p>(c)</p>	2402	a_1	15,00 cm	15,30 cm
				a_2	10,0 cm	10,20 cm
				a_3	6,00 cm	6,12 cm
				ϵ_1	0,10	0,10
				ϵ_2	0,10	0,11
Pose do objeto						
Orientação exata	$\theta = 0,00^\circ$	$\phi = 90,00^\circ$	$\psi = 0,00^\circ$	⁽¹⁾ Erro do ajuste inicial = 41,42		
Orientação estimada	$\theta = 0,00^\circ$	$\phi = 90,04^\circ$	$\psi = 0,00^\circ$	⁽²⁾ Erro do ajuste final = 0,00031		
*Posição exata	$p_x = -8,00$ cm	$p_y = -5,00$ cm	$p_z = 9,00$ cm	⁽³⁾ Quantidade de iterações: 5		
*Posição estimada	$p_x = -8,00$ cm	$p_y = -5,00$ cm	$p_z = 9,00$ cm	⁽⁴⁾ Tempo de processamento = 2 segundos		

⁽¹⁾ Erro do ajuste inicial: Função objetivo calculada com os parâmetros iniciais.

⁽²⁾ Erro do ajuste final: Função objetivo calculada com os parâmetros finais estimados pelo método de minimização.

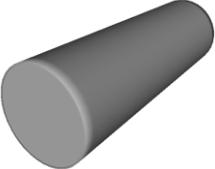
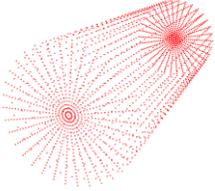
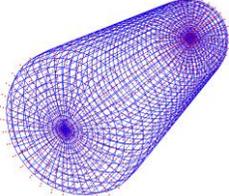
⁽³⁾ Quantidade de iterações utilizada pelo método para determinar os parâmetros da superelipsóide que melhor se ajustou à nuvem de pontos.

⁽⁴⁾ Tempo de processamento do programa principal estimado pela instrução *difftime*(tempo_final, tempo_inicial) nativa da linguagem C++.

*Posição: coordenadas do vetor posição em relação à origem do sistema de coordenadas global.

Observação: Os parâmetros iniciais $a_1, a_2, a_3, \epsilon_1 = 1,0$ e $\epsilon_2 = 1,0$ foram fornecidos automaticamente ao método de minimização.

Tabela 5.9 - Superelipsóide ajustada à um cilindro tombado.

Objeto	Nuvem de pontos do objeto	Superelipsóide ajustada à nuvem de pontos	Quantidade de pontos	Valores estimados pelo método pela minimização da função objetivo: $d(\Lambda, x, y, z) = [F(\Lambda, x, y, z)^{\epsilon_1/2} - 1]^2$		
				Parâmetros	Valor exato	
 (a)	 (b)	 (c)	2402	a_1	15,00 cm	15,30 cm
				a_2	5,00 cm	5,08 cm
				a_3	5,00 cm	5,24 cm
				ϵ_1	0,10	0,10
				ϵ_2	1,00	1,00
Pose do objeto						
Orientação exata	$\phi = -90,00^\circ$	$\theta = 90,00^\circ$	$\psi = 180,00^\circ$	⁽¹⁾ Erro do ajuste inicial = 222,96		
Orientação estimada	$\phi = -74,74^\circ$	$\theta = 90,04^\circ$	$\psi = 180,09^\circ$	⁽²⁾ Erro do ajuste final = 0,00035		
* Posição exata	$p_x = 0,00$ cm	$p_y = 0,00$ cm	$p_z = 15,00$ cm	⁽³⁾ Quantidade de iterações: 11		
* Posição estimada	$p_x = 0,00$ cm	$p_y = 0,00$ cm	$p_z = 14,99$ cm	⁽⁴⁾ Tempo de processamento = 5 segundos		

⁽¹⁾ Erro do ajuste inicial: Função objetivo calculada com os parâmetros iniciais.

⁽²⁾ Erro do ajuste final: Função objetivo calculada com os parâmetros finais estimados pelo método de minimização.

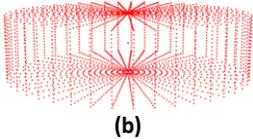
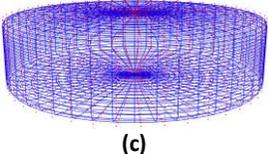
⁽³⁾ Quantidade de iterações utilizada pelo método para determinar os parâmetros da superelipsóide que melhor se ajustou à nuvem de pontos.

⁽⁴⁾ Tempo de processamento do programa principal estimado pela instrução `difftime(tempo_final, tempo_inicial)` nativa da linguagem C++.

*Posição: coordenadas do vetor posição em relação à origem do sistema de coordenadas global.

Observação: Os parâmetros iniciais $a_1, a_2, a_3, \epsilon_1 = 1,0$ e $\epsilon_2 = 1,0$ foram fornecidos automaticamente ao método de minimização.

Tabela 5.10 - Superelipsóide ajustada à um cilindro baixo.

Objeto	Nuvem de pontos do objeto	Superelipsóide ajustada à nuvem de pontos	Quantidade de pontos	Parâmetros	Valor exato	Valores estimados pelo método pela minimização da função objetivo:
						$d(\Lambda, x, y, z) = [F(\Lambda, x, y, z)^{\epsilon_1/2} - 1]^2$
 (a)	 (b)	 (c)	2402	a_1	20,00 cm	20,38 cm
				a_2	20,00 cm	20,42 cm
				a_3	5,00 cm	5,10 cm
				ϵ_1	0,10	0,11
				ϵ_2	1,00	1,00
Pose do objeto						
Orientação exata	$\phi = 0,00^\circ$	$\theta = 90,0^\circ$	$\psi = 180,00^\circ$	⁽¹⁾ Erro do ajuste inicial = 54,16		
Orientação estimada	$\phi = 0,00^\circ$	$\theta = 90,04^\circ$	$\psi = 173,63^\circ$	⁽²⁾ Erro do ajuste final = 0,00042		
* Posição exata	$p_x = -8,00$ cm	$p_y = 3,00$ cm	$p_z = 6,00$ cm	⁽³⁾ Quantidade de iterações: 7		
* Posição estimada	$p_x = -8,00$ cm	$p_y = 3,00$ cm	$p_z = 6,00$ cm	⁽⁴⁾ Tempo de processamento = 3 segundos		

⁽¹⁾ Erro do ajuste inicial: Função objetivo calculada com os parâmetros iniciais.

⁽²⁾ Erro do ajuste final: Função objetivo calculada com os parâmetros finais estimados pelo método de minimização.

⁽³⁾ Quantidade de iterações utilizada pelo método para determinar os parâmetros da superelipsóide que melhor se ajustou à nuvem de pontos.

⁽⁴⁾ Tempo de processamento do programa principal estimado pela instrução `difftime(tempo_final, tempo_inicial)` nativa da linguagem C++.

* Posição: coordenadas do vetor posição em relação à origem do sistema de coordenadas global.

Observação: Os parâmetros iniciais $a_1, a_2, a_3, \epsilon_1 = 1,0$ e $\epsilon_2 = 1,0$ foram fornecidos automaticamente ao método de minimização.

Com a finalização de sete iterações o elevado erro do ajuste inicial igual a 67,17 para a esfera da tabela 5.4, localizada na origem do sistema de coordenadas global, foi reduzido a um EOF = 0,0031. No entanto, a distorção apresentada pela estimativa da orientação da nuvem de pontos não comprometeu a eficiência do processo de modelagem, evidenciando, desse modo, a robustez do algoritmo a aproximações pouco precisas para esses parâmetros.

Resultado semelhante pode ser observado para o cubo da tabela 5.5 localizado a vinte e cinco unidades da origem do sistema de referência. Os parâmetros de forma iguais a 0,1 expõem a natureza dos cantos retos típicos desse objeto.

Na seqüência, oito iterações foram suficientes para modelar o cilindro alto da tabela 5.6, revelado pelos fatores de forma complementares ($\varepsilon_1 = 0,1$ e $\varepsilon_2 = 1,0$), independente da pequena diferença na estimativa da orientação.

Posteriormente, os menores valores permitidos para os parâmetros de forma ($\varepsilon_1 = \varepsilon_2 = 0,1$) caracterizam o volume de cantos retos, verticalmente pronunciado, da caixa alta da tabela 5.7. A redução do alto erro do ajuste inicial (40,04) para um EOF = 0,0017, com apenas oito iterações em três segundos, também demonstra a habilidade do método de ajuste na modelagem de objetos de diferentes configurações geométricas. Novamente, a pequena distorção para a orientação não influenciou na estimativa dos parâmetros finais do modelo.

Cinco iterações processadas em dois segundos foram suficientes para simular a nuvem de pontos do paralelepípedo tombado da tabela 5.8. Um fator de qualidade igual a 0,00031 representa o mínimo valor da função objetivo iniciada no inaceitável patamar descrito pelo erro do ajuste inicial 41,42.

É importante observar que a inerente dificuldade em se trabalhar com a função implícita da equação (3.26), sobretudo para fatores de forma complementares, não impediu a correta modelagem do cilindro tombado da tabela 5.9. O desafio numérico na modelagem deste item foi ligeiramente enfatizado pela sua disposição espacial, exigindo a maior quantidade de iterações para atingir a aceitável tolerância exposta por EOF = 0,00035.

A considerável distorção na estimativa da orientação do cilindro baixo da tabela 5.10 não influenciou na obtenção de um EOF = 0,0004 produzido por sete iterações conduzidas em apenas três segundos.

Independente do objeto modelado, os resultados obtidos para as variáveis que designam o tamanho e os expoentes de forma demonstraram coerência para com os valores exatos. Esse comportamento evidencia a eficiência do algoritmo de modelagem na estimativa correta dos parâmetros geométricos das primitivas simuladas. A baixa média de duração dos processamentos, em torno de cinco segundos, avaliou o desempenho aceitável do método de ajuste.

De qualquer modo, a excelente qualidade da modelagem indicada pelos baixos valores das funções objetivos demonstra a eficiência do método. Além disso, as distorções obtidas estão coerentes com aquelas demonstradas por Chevalier *et al.* (2003) (EOF = 0,0035) e Zhang (2003a) (EOF = 0,40), para objetos sintéticos.

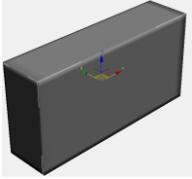
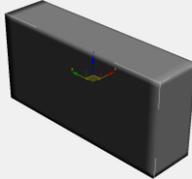
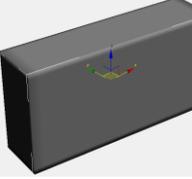
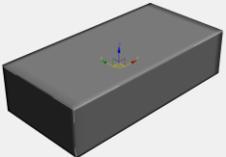
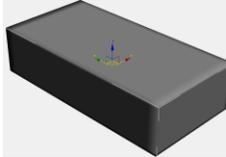
Todavia, o esforço computacional pode ser reduzido diante da aceitação de uma distorção maior para os fatores de forma, uma vez que a estimativa para os parâmetros de tamanho dos modelos é obtida por rotinas de cálculo independentes.

A instrução *difftime*, nativa da linguagem C++, foi utilizada somente para estimar a duração do processamento, não devendo, desse modo, ser interpretada como critério isolado na classificação do desempenho do método de ajuste. Especificamente, ela calcula a diferença entre os instantes de início e término da simulação. Em particular, um processador de 2,80 GHz, com arquitetura de 32 bits e 1,50 GB de memória RAM, foi o responsável pela execução do programa computacional.

5.3 Coerência na interpretação dos modelos ajustados

Para demonstrar a coerência espacial e compatibilidade dos modelos simulados, dado um paralelepípedo de referência, a tabela 5.11 exhibe a modelagem de seis configurações para este item. Por intermédio de ajustes nas proporções, essa primitiva pode representar o celular, a *pen drive* e a chave de ignição das figuras 1.2 (a), (h) e (i), respectivamente.

Tabela 5.11 – Modelagem de um paralelepípedo organizado em seis configurações.

					
(a) Objeto referência localizado na origem do sistema de coordenadas global.	(b) Rotação de 90° em torno do eixo z e mudança na posição do objeto.	(c) Rotação de -170° em torno do eixo z e mudança na posição do objeto.			
Posição e orientação iniciais estimados pelo programa					
⁽¹⁾ Posição	⁽²⁾ Orientação	⁽¹⁾ Posição	⁽²⁾ Orientação	⁽¹⁾ Posição	⁽²⁾ Orientação
$p_x = 0 \text{ cm}$	$\theta_x = 90^\circ$	$p_x = 5 \text{ cm}$	$\theta_x = 90^\circ$	$p_x = 38 \text{ cm}$	$\theta_x = 90^\circ$
$p_y = 0 \text{ cm}$	$\theta_y = 90^\circ$	$p_y = 11 \text{ cm}$	$\theta_y = 90^\circ$	$p_y = 45 \text{ cm}$	$\theta_y = 90^\circ$
$p_z = 0 \text{ cm}$	$\theta_z = 0^\circ$	$p_z = 115 \text{ cm}$	$\theta_z = -90^\circ$	$p_z = 103 \text{ cm}$	$\theta_z = -170^\circ$
Parâmetros ajustados pelo método de minimização: Levenberg-Marquardt					
$a_1 = 20,40 \text{ cm}$	$\varepsilon_1 = 0,10$	$a_1 = 5,10 \text{ cm}$	$\varepsilon_1 = 0,10$	$a_1 = 20,47 \text{ cm}$	$\varepsilon_1 = 0,10$
$a_2 = 5,10 \text{ cm}$	$\varepsilon_2 = 0,10$	$a_2 = 20,40 \text{ cm}$	$\varepsilon_2 = 0,10$	$a_2 = 5,25 \text{ cm}$	$\varepsilon_2 = 0,13$
$a_3 = 10,20 \text{ cm}$		$a_3 = 10,20 \text{ cm}$		$a_3 = 10,32 \text{ cm}$	
					
(d) Rotação de -90° e 0° em torno dos eixos x e z, respectivamente, além da mudança na posição do objeto.	(e) Rotação de -90° , -90° e 0° em torno dos eixos x, y e z, respectivamente, além da mudança na posição do objeto.	(f) Rotação de 90° , 90° e 0° em torno dos eixos x, y e z, respectivamente, além da mudança na posição do objeto.			
Posição e orientação iniciais estimados pelo programa					
⁽¹⁾ Posição	⁽²⁾ Orientação	⁽¹⁾ Posição	⁽²⁾ Orientação	⁽¹⁾ Posição	⁽²⁾ Orientação
$p_x = 20 \text{ cm}$	$\theta_x = 0^\circ$	$p_x = 20 \text{ cm}$	$\theta_x = 0^\circ$	$p_x = -31 \text{ cm}$	$\theta_x = 0^\circ$
$p_y = 0 \text{ cm}$	$\theta_y = 90^\circ$	$p_y = 0 \text{ cm}$	$\theta_y = 0^\circ$	$p_y = 55 \text{ cm}$	$\theta_y = 90^\circ$
$p_z = 130 \text{ cm}$	$\theta_z = 0^\circ$	$p_z = 130 \text{ cm}$	$\theta_z = 0^\circ$	$p_z = 118 \text{ cm}$	$\theta_z = -90^\circ$
Parâmetros ajustados pelo método de minimização: Levenberg-Marquardt					
$a_1 = 20,40 \text{ cm}$	$\varepsilon_1 = 0,10$	$a_1 = 5,00 \text{ cm}$	$\varepsilon_1 = 0,10$	$a_1 = 10,20 \text{ cm}$	$\varepsilon_1 = 0,10$
$a_2 = 10,20 \text{ cm}$	$\varepsilon_2 = 0,10$	$a_2 = 10,00 \text{ cm}$	$\varepsilon_2 = 0,10$	$a_2 = 20,40 \text{ cm}$	$\varepsilon_2 = 0,10$
$a_3 = 5,10 \text{ cm}$		$a_3 = 18,90 \text{ cm}$		$a_3 = 5,10 \text{ cm}$	

⁽¹⁾ Posição do baricentro da nuvem de pontos estimada pelo programa em relação ao sistema de coordenadas global;

⁽²⁾ Orientação da nuvem de pontos estimada pelo programa;

Apesar da caixa irregular de referência ser representada por 2402 pontos, para a conveniência de visualização, modelos volumétricos são ilustrados na tabela 5.11 para facilitar a interpretação dos resultados. Análogo aos procedimentos utilizados na seção anterior, todos os parâmetros iniciais necessários para a completa execução do método de ajuste foram fornecidos automaticamente pelo programa principal a partir da nuvem de pontos.

Estas simulações foram motivadas pelo fato da decomposição de objetos complexos resultar, na maioria dos casos, em primitivas em poses genéricas. Tal observação revela aspectos pertinentes à modelagem de itens de uso cotidiano, tais como aqueles ilustrados na figura 1.1.

Os resultados mostrados na tabela 5.11 são auto-explicativos. Contudo, é importante enfatizar que os parâmetros originais de tamanho são iguais a $a_x = 20$ cm, $a_y = 10$ cm e $a_z = 5$ cm nas direções dos eixos coordenados x , y e z , respectivamente, enquanto os expoentes de forma são representados por $\varepsilon_1 = \varepsilon_2 = 0,10$. A notação alternativa para tais termos se justifica observando que para um mesmo sistema de coordenadas o lado de maior dimensão do paralelepípedo, inicialmente alinhado ao eixo x da tabela 5.11(a), foi indicado pela direção do eixo z na configuração da tabela 5.11(e). Comportamento análogo pode ser verificado para a dimensão igual a dez unidades da tabela 5.11(b) indicada pelo eixo z , enquanto a menor dimensão do paralelepípedo da tabela 5.11(d) ser revelada na direção deste eixo, apenas para citar alguns exemplos. É relevante mencionar que essa interpretação ainda pode ser estendida para outras configurações ilustradas na tabela.

Embora a configuração da tabela 5.11(c) e (e) sugira potenciais desafios computacionais para o método de ajuste, em razão da orientação de 30° em relação a cada um dos eixos coordenados x e y , o programa ainda demonstrou coerência na modelagem desses itens. Apesar do paralelepípedo na pose da tabela 5.11(a) ter sido utilizado como referência outros objetos atenderiam ao mesmo propósito.

A pose da tabela 5.11(f) foi obtida em oito iterações, quatro segundos e EOF = 0,0004. Em contrapartida, a modelagem da pose da tabela 5.11(e) envolveu 25 iterações em 12 segundos para reduzir o inaceitável erro do ajuste inicial igual a 150,49 para a baixa distorção final igual a 0,0013.

Para citar uma comparação adicional de qualidade, a configuração da tabela 5.11 (b) foi simulada em quatro iterações (2 segundos) e produziu um modelo com bordas ligeiramente arredondadas ($\varepsilon_1 = 0,16$ e $\varepsilon_2 = 0,18$) e $EOF = 0,0011$, em contraste à perfeita modelagem caracterizada por seis iterações (2 segundos), $EOF = 0,0003$ e $\varepsilon_1 = \varepsilon_2 = 0,10$. Comportamento análogo foi observado na representação da pose da tabela 5.11(a) com ($\varepsilon_1 = 0,16$ e $\varepsilon_2 = 0,18$), $EOF = 0,001$ (4 iterações e 1 segundo) contra ($\varepsilon_1 = \varepsilon_2 = 0,10$) e $EOF = 0,0003$ (5 iterações em 2 segundos) para uma aproximação com bordas retas.

Por intermédio da comparação entre as grandezas originais e aquelas obtidas pelo programa de ajuste para os parâmetros de forma e tamanho, a simulação das seis configurações do paralelepípedo da tabela 5.11 demonstrou coerência espacial e geométrica para com o objeto de referência.

A modelagem de objetos cotidianos manipulados por agentes humanos contemplam, tipicamente, primitivas orientadas na vertical ou horizontal. Conforme mencionado, essa observação justificou a representação de itens volumétricos arranjados segundo tais configurações. Krivic (2006) observa que a configuração de objetos, para o caso genérico, é difícil de ser estimada devido à ambigüidade rotacional inerente às superelipsóides. Como consequência, o processo de modelagem pode demandar o desenvolvimento de procedimentos dedicados a cada tipo ou grupos de objeto.

Diante da diversidade das formas geométricas simuladas, as possíveis combinações envolvidas na modelagem de objetos de uso cotidiano empregando tais primitivas representam um desafio adicional para uma abordagem heurística, até mesmo para itens modelados por formas básicas semelhantes. Para mencionar alguns exemplos, a escolha pela primitiva de maior dimensão para objetos constituídos por formas básicas semelhantes, tais como ocorre para facas (dois paralelepípedos) e garrafas (dois cilindros), poderia resultar na seleção correta do volume da pega para utilizar tais itens, porém falharia na escolha do cabo (cilíndrico) de uma tampa (cilíndrica) de panela. O mesmo comportamento indesejável é revelado pela escolha incorreta da seção mais alongada de uma “tábua de carne” (composta por dois paralelepípedos). Essa observação é ainda mais desafiadora para objetos constituídos

por primitivas volumétricas diferentes, como acontece com a associação do cabo cilíndrico e o paralelepípedo da lâmina de borracha de um “rodinho de pia”. Conclusão idêntica se aplica a um “ralador de legumes” formado pelas primitivas mencionadas.

Com a finalidade de acomodar a abrangência de combinações envolvendo a modelagem de objetos, o modo da pega e a primitiva adequada para o desenvolvimento pleno de uma determinada tarefa, uma abordagem alternativa baseada em um sistema de auto-aprendizagem é desenvolvida no próximo capítulo.

Introdução ao estado da arte do planejamento de estratégias para a pega de objetos

Ao longo das últimas décadas, diversos métodos analíticos (Nguyen, 1987; Mirtich, 1994; Liu, 1999; Ding *et al.*, 2000; e Zhu *et al.*, 2003), somente para citar algumas propostas relevantes, tiveram o seu foco exclusivamente direcionado à análise e síntese de pegas estáveis de objetos envolvendo pinças e mãos robóticas antropomórficas. Estes trabalhos representam contribuições fundamentais para a compreensão e aprimoramento de pegas de objetos para operações *pick-and-place*, mas eles são inadequados para a manipulação de objetos desconhecidos, haja vista a variedade de pegas estáveis que podem se mostrar impróprias para a realização de determinadas tarefas.

Por intermédio de formalismos cinemático e dinâmico, abordagens analíticas consideram a configuração e as posições dos dedos de uma mão robótica para a síntese de uma pega. Tais técnicas otimizam funções objetivo, geralmente baseadas em critérios de estabilidade de pega ou especificações da tarefa, para analisar maneiras adequadas de promover simples interações com objetos. Em contrapartida, métodos alternativos utilizam algoritmos de auto-aprendizagem para tentar reproduzir atitudes humanas tipicamente envolvidas na manipulação de objetos.

As duas primeiras seções deste capítulo apresentam uma breve descrição das principais estratégias analítica e empírica freqüentemente adotadas na síntese de pegas de objetos. Ao longo do texto, o termo tarefa é caracterizado como o espaço tridimensional de forças e torques que devem ser aplicados por uma mão robótica com a finalidade de desenvolver interações com objetos.

6.1 Métodos analíticos de pegadas de objeto orientadas pela tarefa

O primeiro indicador analítico da qualidade de uma pega relacionada à tarefa foi apresentado por Li e Sastry (1988). O estudo mostrou que a escolha de uma pega orientada pela ação deve ser baseada na capacidade dos dedos da mão robótica em produzir forças e torques (*wrenches*) relevantes àquela tarefa. Assumindo um conhecimento a priori da interação com o objeto e a geometria do espaço de trabalho, (ver figura 6.1), o planejamento prévio da trajetória de deslocamento do objeto envolveu a modelagem da tarefa explorando o espaço de trabalho definido pela mão robótica.

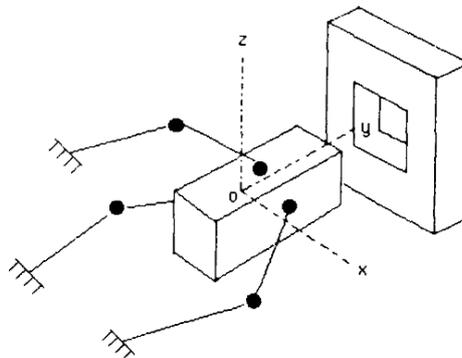


Figura 6.1 – Tarefa Peg-in-hole (Li e Sastry, 1988)

A dificuldade em modelar as forças e torques que devem ser desenvolvidos para atender à tarefa revela a restrição da proposta de Li e Sastry (1988), observando que as intensidades dessas grandezas devem pertencer àquelas passíveis de serem aplicadas pelo sistema robótico.

Baseado em uma determinada forma de pega de referência, definida como um modo de alta qualidade para agarrar um objeto protótipo, Pollard (1997) projetou um sistema capaz de determinar configurações de pegadas com porcentagem de qualidade aceitável para agarrar objetos similares àqueles adotados na base de dados. Considerando que a probabilidade de ocorrência das forças e torques que podem perturbar a estabilidade da pega de um objeto é igual para qualquer direção, Pollard (1997) utilizou uma esfera para modelar o espaço dessas grandezas. O indicador de qualidade foi representado pela capacidade da mão robótica em explorar o objeto com o propósito de resistir às forças e torques demandados pela tarefa, enquanto mantém o

objeto agarrado. Ao passo em que este sistema pode ser considerado uma das ferramentas de síntese de pega mais genérica, ele apresenta o desafiante compromisso de escolher a pega de referência, uma vez que as pegas sugeridas pela proposta podem ser inadequadas para a execução de determinadas tarefas.

Em uma abordagem inovadora, Borst *et al.* (2004) incorporou a descrição geométrica do objeto à composição do espaço das forças e torques definido pela pega. Como resultado, o volume delimitado por essas entidades modela qualquer perturbação externa que possa atuar sobre a superfície do objeto (figura 6.3).

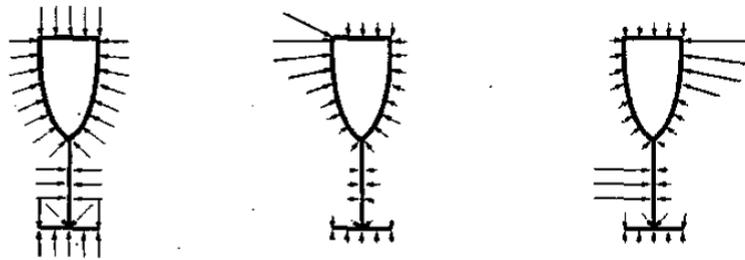


Figura 6.3 – Ilustração das diferentes distribuições das forças e torques que podem atuar sobre o objeto. A soma da intensidade normalizada de todos os vetores de força resulta em um valor unitário. Borst *et al.* (2004)

Combinado à modelagem analítica de Li e Sastry (1988), Borst *et al.* (2004) obtiveram uma nova descrição para o espaço das forças e torques que especificam uma tarefa. A partir deste procedimento, a qualidade da pega foi determinada pela comparação do volume definido pelas forças e torques exigidos pela tarefa com aqueles permitidos pela configuração dos dedos da mão robótica.

Diferentemente das propostas mencionadas acima, a abordagem de Haschke *et al.* (2005) determinou índices de qualidade para as tarefas como critério de seleção das pegas. Em particular, tarefas podem ser especificadas como um conjunto de *wrenches* que pode ser aplicado em uma das possíveis direções contidas no interior de um cone de aceitação de forças (figura 6.4a) localizado no ponto de contato produzido pela ponta do dedo sobre a superfície do objeto. Resultados baseados no formalismo *Linear Matrix Inequality* (LMI) para uma modelagem mais precisa do cone de aceitação e

simultânea otimização das forças aplicadas, conduziram a soluções capazes de indicar a máxima intensidade da força e do torque possíveis de serem aplicados na direção da tarefa (*wrench*) especificada (figura 6.4b).

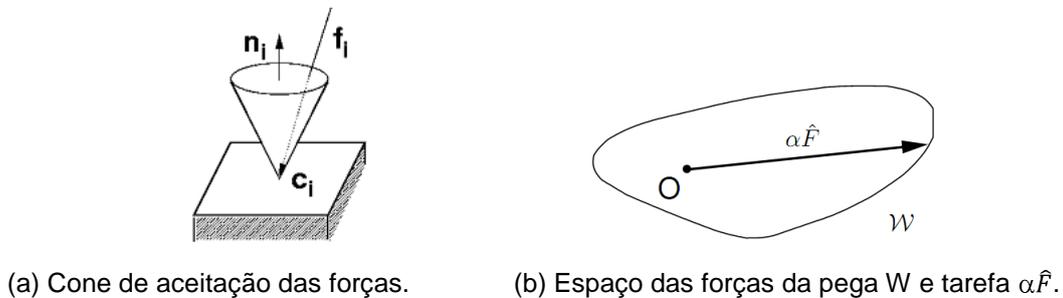


Figura 6.4 – A proposta baseada no índice de qualidade da pega maximiza o valor de α de uma determinada combinação de forças e torques aplicados na execução de uma determinada tarefa contida no interior do espaço W permitido pelas *wrenches* da pega (Haschke *et al.*, 2005).

Em contrapartida, o método analítico de Pratz *et al.* (2007) considerou a tarefa como estratégia para o planejamento da pega. Adotando uma versão geométrica simplificada do objeto e uma descrição da tarefa a ser realizada, o sistema determina automaticamente a pré-forma da pega adequada para desenvolver a ação de interesse. Para interagir com uma maçaneta, por exemplo, a figura 6.5 ilustra as possíveis formas preênses compatíveis com o interesse em abrir uma porta ou gaveta. Cada objeto armazenado no banco de dados é modelado com uma hierarquia de caixas para facilitar o processo de decisão da pega.



Figura 6.5 – Formas preênses orientadas pela tarefa. Pratz *et al.* (2007)

Os métodos analíticos discutidos nesta seção se concentram na análise de pegas específicas ou tarefas particulares. Todavia, a complexidade computacional envolvida

na modelagem das tarefas revela a dificuldade do uso de tais técnicas no contexto de aplicações envolvendo robótica móvel. Além disso, essas abordagens podem se tornar inadequadas diante da alteração das especificações das tarefas ou mesmo quando novos objetos são apresentados ao sistema, situações que podem demandar a redefinição do oneroso formalismo exigido. Com o objetivo de evitá-la, métodos alternativos têm sido introduzidos como abordagens na escolha de formas preênses compatíveis com as especificações das tarefas como descreve a próxima seção.

6.2 Métodos empíricos de pegas de objeto orientadas pela tarefa

É importante destacar que, apesar do aspecto da estabilidade enfatizado pelas técnicas analíticas ser uma condição necessária na interação com objetos, ela não é suficiente para assegurar o sucesso do planejamento das pegas. As comparações da figura 6.6(a) para a frequência de sucesso entre pegas humanas e versões sintéticas geradas aleatoriamente revelou que a elevada concentração de pegas aleatórias, figura 6.6(b), se concentra próxima ao baixo índice de qualidade de pegas humanas. A partir desta observação, Borst *et al.* (2003) demonstraram que não é necessário sintetizar pegas robóticas ótimas quando um critério de qualidade média é aceitável.

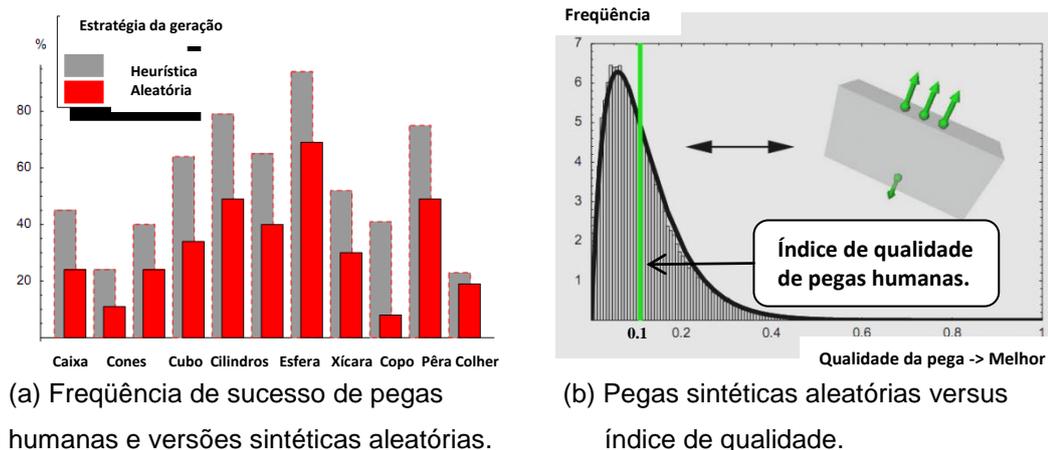


Figura 6.6 – Comparação entre pegas humanas e versões sintéticas aleatórias. A barra vertical da imagem (b) indica o índice de qualidade de uma pega humana para a caixa ilustrada no gráfico. Borst *et al.* (2003)

Este resultado expõe condições motivadoras para justificar o uso de técnicas de planejamento empíricas que tentam reproduzir as estratégias humanas para pegar objetos. As modalidades empíricas podem ser divididas em dois grandes grupos: (1^o) sistemas baseados na observação de um ser humano desenvolvendo uma pega e (2^o) sistemas baseados na observação do objeto a ser agarrado, que foi a modalidade adotada neste trabalho. Dentre as diversas propostas recentes orientadas pela tarefa, empregando métodos empíricos (Wolfe *et al.*, 2010; Stilman, 2010; Chang *et al.*, 2010; e Popovic *et al.*, 2010), esta seção teve seu foco direcionado àquelas abordagens relevantes para a realização deste trabalho.

Considerando o desafio de interagir com novos objetos, especialmente aqueles apresentados pela primeira vez a um sistema de visão computacional, Saxena *et al.* (2008) apresentaram um algoritmo baseado em aprendizagem para explorar características visuais de objetos como forma de estimar pontos de contato favoráveis ao processo da pega. A técnica identifica um ponto sobre a superfície de um objeto 3D a partir da captura de imagens bidimensionais, enquanto um sistema de aprendizagem supervisionada é utilizado para identificar seções dessas imagens que possam conter pontos de pega interessantes para o propósito da aplicação. Para realizar esta etapa da abordagem, imagens sintéticas de objetos identificados com a localização bidimensional dos pontos de contato foram adotadas no treinamento da rede neural artificial (figura 6.7).

De acordo com o algoritmo de Saxena *et al.* (2008), a partir de duas ou mais imagens bidimensionais de um objeto, identificam-se pontos de contato em cada uma delas que possam corresponder a localizações favoráveis ao ato da pega. Encerrado este procedimento, o conjunto de pontos resultante é submetido a um processo de triangulação para produzir a localização real do ponto sobre a superfície do objeto 3D.

Por intermédio do uso de um banco de dados formado por versões de pegas humanas, Li e Pollard (2007) apresentaram uma abordagem para sintetizar configurações dos dedos para objetos diferentes daqueles armazenados no sistema (ver figura 6.8a).

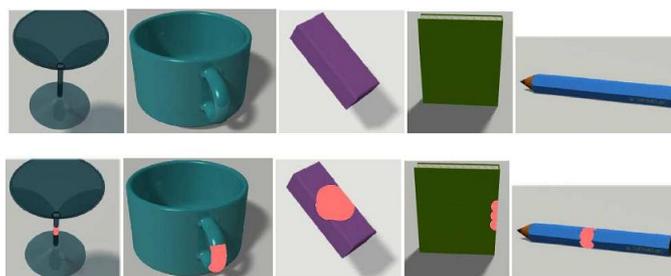


Figura 6.7 – Classes de objetos utilizadas no estágio de treinamento da rede neural artificial. A linha superior da figura ilustra os objetos originais, enquanto a corresponde linha inferior destaca a identificação das regiões favoráveis à pega (Saxena *et al.*, 2008).

Tal processo é conduzido por um algoritmo de *shape matching* capaz de comparar as características de forma do objeto àquelas apresentadas pelas poses da mão humana, capturando informações sobre as configurações relativas das posições de contato e suas respectivas componentes normais em cada ponto da superfície do objeto (figura 6.8b).

A pose de uma mão humana para agarrar um *mouse* é ilustrada na figura 6.8b. A imagem mostra pontos de contato sobre os dedos e a palma da mão, bem como vetores normais à superfície do objeto. É possível observar que a palma da mão contém uma grande quantidade de informações sobre a forma geométrica do objeto. Quando características similares são identificadas em novos objetos, é possível que o mesmo modo de pega seja reproduzido.

Finalizada a etapa de comparação, algumas poses para os dedos podem ser inadequadas para a tarefa. Desse modo, Li e Pollard (2007) adotaram um critério de qualidade que considera a configuração da mão e as especificações da tarefa para avaliar as formas preênses sugeridas pelo sistema. Apesar desta consideração, o usuário deve selecionar manualmente a pega desejável a partir do elenco de possibilidades apresentado pelo sistema em virtude de alguns resultados serem não intuitivos.

Baseado no fato de que o cérebro humano representa objetos como formas volumétricas básicas, El-Khoury e Sahbani (2008) apresentaram um algoritmo que, a

partir do modelo geométrico do objeto, identifica a primitiva adequada para o processo da pega.

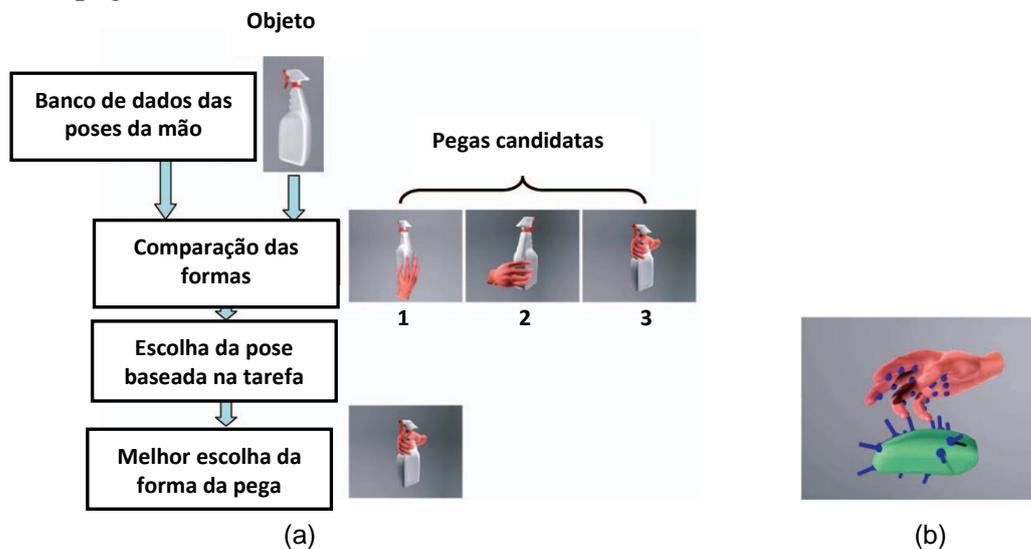


Figura 6.8 – Síntese de pegadas baseada na similaridade das formas. O sistema de comparação de formas procura na base de dados a melhor pose da mão que atende ao objeto. As poses representativas 1, 2 e 3 são, nesta ordem, as configurações dos dedos para agarrar um mouse, uma jarra e uma lâmpada, Li e Pollard (2007).

El-Khoury e Sahbani (2008) acreditam que muitos objetos são equipados com regiões especificamente projetadas para simplificar a pega na execução de tarefas rotineiras. A partir de um modelo tridimensional, um estágio de segmentação baseado na análise de curvaturas decompõe objetos desconhecidos em formas geométricas básicas. Na seqüência, a descrição geométrica de cada primitiva é representada por superelipsóides. Finalizada a modelagem das formas básicas, um sistema de auto-aprendizagem permite que a proposta desenvolva uma analogia à escolha humana sobre a primitiva adequada para o processo da pega (figura 6.9).

Embora os resultados da figura 6.9 demonstrem a capacidade do sistema de aprendizagem em reproduzir algumas atitudes humanas para interagir com objetos, uma limitação de aplicação da proposta é revelada pela ausência da especificação da tarefa a ser conduzida. É importante enfatizar que para atender aos propósitos de manipulação de objetos envolvendo assistentes robóticos móveis, a escolha da forma preênsil adequada para a execução completa de uma determinada ação desempenha

aspectos fundamentais no tempo de processamento computacional e duração das limitadas fontes de energia desses agentes.

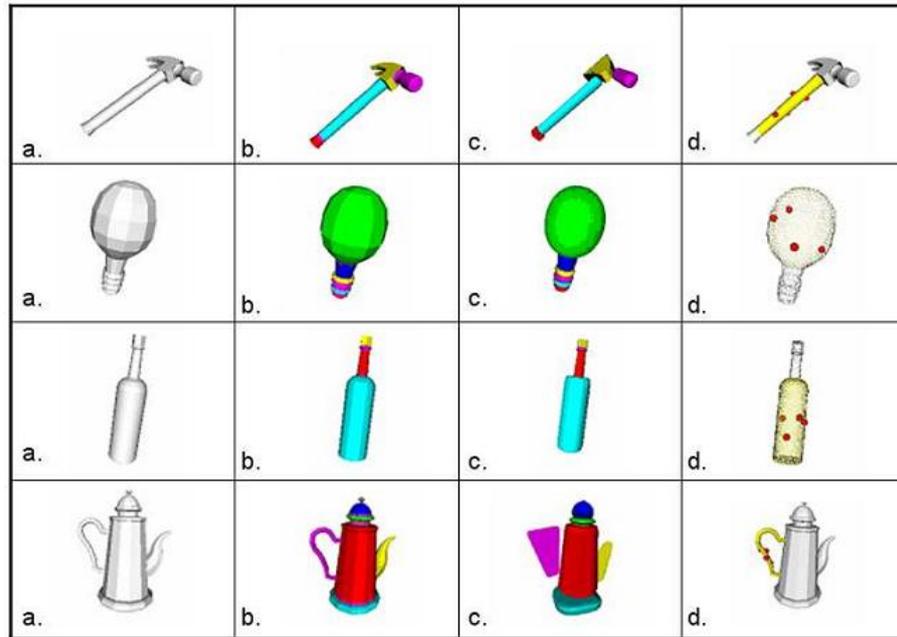


Figura 6.9 – Objetos utilizados no teste do sistema de auto-aprendizagem: (a) objeto original; (b) segmentação em formas básicas; (c) modelagem por superelipsóides; (d) pontos de contato sobre as superfícies dos objetos. El-Khoury e Sahbani (2008)

Para sustentar um recipiente repleto de líquido, por exemplo, a distribuição dos pontos de contato sobre a superfície da garrafa da figura 6.9 sugere um modo de pega de precisão, no qual somente as pontas dos dedos estabelecem contato com a superfície do objeto. No entanto, este caso particular pode exigir elevadas intensidades de forças, bem como correspondentes torques nas juntas dos dedos, para preservar a orientação espacial da garrafa. Essa condição pode favorecer um custo adicional de energia, observando que o centro de massa do recipiente, supostamente localizado no interior do corpo da garrafa, poderia demandar uma potência suficientemente alta das baterias para classificar a pega ilustrada como insatisfatória.

Esta condição indesejável pode ser enfatizada pelo maior tempo de processamento envolvido na localização de pontos de contato estratégicos, e conseqüente dispendioso planejamento da trajetória dos dedos, somente para sustentar

a orientação da garrafa durante um suposto deslocamento. Neste contexto, a forma preênsil envolvente, proporcionada pelo polegar, dedos e palma da mão, conduziria a uma pega dinamicamente mais estável e menos onerosa sob o ponto de vista do consumo de energia elétrica, além de se aproximar das formas tipicamente humanas para agarrar objetos cilíndricos com dimensões compatíveis àquelas definidas pela distância do polegar à ponta dos demais dedos.

Comportamento similar pode ser observado (1º) na escolha da alça e (2º) pela elevada concentração de pontos sobre a superfície dessa região na chaleira da figura 6.9. Novamente, a distribuição dos pontos sugere uma pega definida exclusivamente pelas pequenas áreas de contato entre as pontas dos dedos e a superfície da primitiva, ao invés do modo estável e envolvente proporcionado pelo polegar, dedos e palma da mão.

Adicionalmente, a indicação pela rede neural do corpo da garrafa da figura 6.9 como região compatível com o interesse em alterar a sua posição, com preservação da orientação do objeto, desclassificaria a escolha do sistema de aprendizagem de El-Khoury e Sahbani (2008) quando o recipiente estiver sobre a superfície de uma mesa repleta de outros objetos. Resultado insatisfatório também poderia se originar do desejo em conduzir a mesma garrafa para o interior de espaços delimitados, por exemplo, por divisórias instaladas em engradados de embalagens. É importante enfatizar que tais condições descrevem aspectos tipicamente definidos por cenários não controlados, nos quais os objetos não se encontram isolados e totalmente disponíveis para a execução completa do processo da pega. Uma vez que o objetivo da manipulação explora a possibilidade de interagir com objetos, independente das suas condições espaciais, a presença de outros itens próximos aqueles de interesse não pode impor desafios concorrentes para o sistema robótico, além daqueles definidos pelo ato da pega.

Uma proposta alternativa para combater tais resultados indesejados é apresentada no próximo capítulo, o qual desenvolve uma estratégia de pega de objetos orientada à tarefa.

Estratégia de pega de objetos orientada à tarefa

Normalmente, quando um usuário se aproxima de um objeto, existe um objetivo ou uma tarefa a ser conduzida sobre ele. Com a finalidade de desenvolver a ação desejada, o usuário deve pegar o objeto de modo compatível com pelo menos um dos atributos da tarefa, sejam eles descritos pela forma preênsil adotada para pegar uma xícara de café ou a forma envolvente de agarrar uma ferramenta para produzir torque. No caso específico deste trabalho, tais modalidades de pega envolvem duas atitudes tipicamente adotadas por seres humanos para interagir com objetos de uso cotidiano no desenvolvimento de duas tarefas rotineiras:

- *Guardar*: pegar objeto para mudá-lo de posição, preservando a sua orientação;
- *Utilizar*: pegar objeto para explorar suas características funcionais por intermédio da mudança de posição e orientação.

A teoria do *Reconhecimento por Componente* de Biederman (1987) sugere que os seres humanos são capazes de reconhecer objetos segmentando-os em partes a partir da observação de regiões geométricas de concavidades significativas. Baseada nesta observação, Biederman (1987) afirma que a identificação do item a ser pego não depende da familiaridade com o objeto e a decisão pela forma da pega é conduzida da mesma maneira para qualquer objeto, independente do conhecimento prévio do usuário.

Com a finalidade de facilitar a pega dos objetos, a maioria desses itens é especificamente projetada para simplificar o processo de interação. A utilização típica de uma xícara ou caneca para servir líquido é geometricamente representada pela alça formada por uma seção cilíndrica curva. Analogamente, o segmento alongado de

escovas de dente, chaves de fenda, facas, pincéis, garrafas, entre outros de categoria similar, proporcionam uma componente muito simples e direta de interação. Desse modo, a identificação dessas regiões pode resultar na escolha adequada da primitiva a ser pega para o sucesso da manipulação (El-khoury e Sahbani, 2008).

Considerando objetos de uso cotidiano como itens representativos para os propósitos de aplicação deste trabalho, os seguintes critérios foram adotados:

- os objetos são modelados por duas primitivas pertencentes ao elenco de cinco formas básicas: esfera, cilindro, cubo, elipsóide e paralelepípedo;
- objetos são especificamente projetados para simplificar o processo da pega;
- dependendo da tarefa a ser desenvolvida, objetos com primitivas similares podem ser pegos de modos diferentes;
- o tamanho relativo das formas básicas é fundamental para a seleção da primitiva a ser pega.

Embora a estratégia de pega baseada em “*Grasping By Components*” de Sahbani e El-Khoury (2008) desenvolva uma analogia ao comportamento do ser humano para pegar um determinado objeto, a avaliação de desempenho do sistema é conduzida pela simples comparação do resultado do sistema de auto-aprendizagem com a escolha típica de um agente humano para interagir com os objetos. Conforme exposto no capítulo anterior, a irrelevância da tarefa a ser realizada expõe a restrição de aplicação do método para os propósitos de aplicação em robótica móvel.

Com a finalidade de apresentar uma proposta alternativa para auxiliar no planejamento de estratégias de manipulação, o presente capítulo desenvolve uma técnica baseada em uma rede neural artificial suficientemente capaz de aprender a associar primitivas que compõem objetos de geometrias simples a formas preênses compatíveis com a especificação de determinadas tarefas.

7.1 Pegas de objetos compatíveis com a tarefa

Com a finalidade de evitar a complexa descrição analítica exigida na modelagem de tarefas, uma rede neural artificial (RNA) com método de auto-aprendizagem supervisionada (Haykin, 2001) foi utilizada para atender aos propósitos de aplicação deste trabalho. Especificamente, o algoritmo de pega aprende a utilizar as informações de forma e tamanho dos modelos básicos que compõem a maioria dos objetos de uso cotidiano para (1^o) selecionar a primitiva adequada com uma determinada tarefa e (2^o) indicar o modo preênsil que deve ser adotado para a completa realização da ação.

O programa de modelagem das primitivas revelou a necessidade de onze parâmetros para descrever uma superelipsóide em uma configuração genérica no espaço. Uma vez que o desempenho de tal estágio se mostrou invariante à posição e orientação das formas básicas, somente os parâmetros de forma (ϵ_1 e ϵ_2) e tamanho (a_1 , a_2 , a_3) são utilizados para representar as primitivas adotadas.

Embora a diversidade de itens a serem reconhecidos pela rede neural artificial possa ser composta por um elenco relativamente restrito de objetos, ela ainda representa um volume significativo de informações. Além disso, dada a necessidade de explorar duas saídas do sistema de auto-aprendizagem, uma para a primitiva e outra para a forma preênsil, a natureza constitutiva básica de um *Perceptron* (Haykin, 2001) impede o uso de uma RNA simples para os propósitos de aplicação do trabalho. Esta observação é enfatizada pelo fato da natureza geométrica dos objetos de uso cotidiano não conduzir a classes linearmente separáveis.

Com o objetivo de extrair a maioria das características representativas do conjunto de dados de entrada, o *Perceptron* Multicamadas (PMC) (Haykin, 2001) da figura 7.1 ilustra a arquitetura *feedforward* escolhida. Esta rede é formada por uma camada de entrada composta por 11 sinais, duas camadas neurais escondidas constituídas, respectivamente, por 8 e 6 neurônios, e uma camada de saída com dois neurônios representando os correspondentes valores de saída da aplicação.

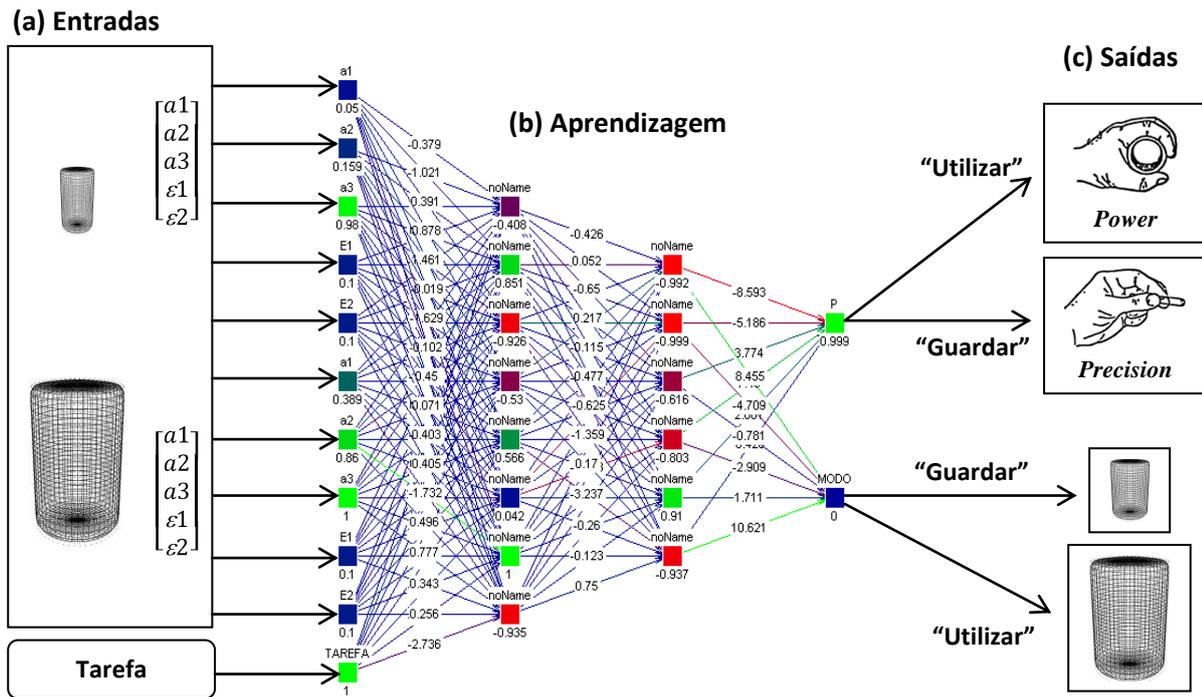


Figura 7.1: Visão geral da seleção da primitiva compatível com a tarefa desejada.

As entradas da PMC são formadas pelo primeiro ($a_1^A, a_2^A, a_3^A, \epsilon_1^A, \epsilon_2^A$) e segundo ($a_1^B, a_2^B, a_3^B, \epsilon_1^B, \epsilon_2^B$) elencos de parâmetros pertencentes ao conjunto dos números reais para descrever a forma e o tamanho das primitivas, além de uma conexão adicional para a entrada manual da tarefa desejada. A inclusão da ação a ser realizada e a forma preênsil escolhida pela RNA representam as contribuições inovadoras da proposta. A saída “P” da figura 7.1 indica a forma básica selecionada pela RNA e, baseado neste resultado, o neurônio “MODO” sugere a forma preênsil conforme descrito a seguir:

- *power grasp*: o polegar se opõe aos demais dedos. Nesta configuração, os dedos envolvem o objeto e aplicam forças na direção da palma da mão. Esta organização dos dedos descreve uma pega segura o bastante para ser adotada quando há espaço suficiente ao redor do objeto e a direção do movimento é desconhecida.
- *precision grasp*: o polegar se opõe aos demais dedos, porém os pontos de contatos com a superfície do objeto são definidos somente pelas pontas dos dedos, enquanto as forças exercidas pelo polegar e demais dedos podem

atuar em sentidos opostos de mesma direção. Esta configuração é útil quando a direção do movimento é conhecida.

Uma vez que não existem metodologias analíticas que assegurem a topologia apropriada, isto é, a quantidade de neurônios em cada camada e o número de camadas ocultas de uma rede PMC, a topologia da RNA da figura 7.1 foi obtida empiricamente para a menor soma do erro quadrático representado pela diferença entre a resposta desejada e aquela obtida pelo sistema. Para atender aos propósitos de aplicação deste projeto foi adotado um patamar de erro inferior a 1,0 % para um total de 100 épocas de treinamento da rede neural. Funções de ativação logística e tangentes hiperbólicas foram adotadas nas camadas de entrada e saída, ao passo que as camadas intermediárias empregaram funções.

Baseado no conhecimento heurístico de que os seres humanos decompõem mentalmente objetos complexos ou desconhecidos em um conjunto de geometrias básicas especificamente projetadas para atender as suas necessidades (ver Biederman, 1987), este trabalho adotou o fato de que um dos aspectos favoráveis à escolha correta pela RNA da primitiva adequada para desenvolver uma determinada tarefa está contido nas informações de forma e tamanho de tais volumes.

No entanto, o aproveitamento direto dos elevados valores que representam as dimensões de objetos reais poderiam conduzir os neurônios das camadas intermediárias a saturação, impedindo, desse modo, que a RNA desenvolvesse uma generalização satisfatória. Para tornar o sistema de auto-aprendizagem invariante às dimensões das primitivas e simultaneamente compatível com a faixa de variação dinâmica das funções de ativação das camadas ocultas, os parâmetros de tamanho das duas primitivas foram normalizados para o maior valor. Os fatores de forma, por sua vez, foram preservados, sempre indicados como 0,1 ou 1,0.

Um fragmento do arquivo utilizado nos processos de treinamento, validação e teste, a serem detalhados nos próximos parágrafos, é mostrado na tabela 7.1.

Tabela 7.1: Padronização adotada nos arquivos de treinamento, validação e teste da RNA na descrição das primitivas, tarefa e saídas desejadas.

Entrada NORMALIZADA												SAÍDA desejada	
OBJETO	a ₁	a ₂	a ₃	ε ₁	ε ₂	a ₁	a ₂	a ₃	ε ₁	ε ₂	TAREFA	Primitiva	Modo preênsil
	Cabo/Corpo/Bocal: 1					Haste/Lâmina/Gargalo: 0							
caneca_300ml	0,89	0,89	1,00	0,1	1,0	0,30	0,68	1,00	0,1	0,1	1	0	1
panela_6	0,63	1,00	1,00	0,1	1,0	1,00	0,19	0,19	0,1	0,1	1	0	0
colher_7	1,00	0,19	0,16	0,1	0,1	1,00	0,58	0,29	1,0	1,0	1	1	1
chave_fenda_4	1,00	0,33	0,33	0,1	1,0	0,04	0,04	1,00	0,1	1,0	1	1	1
caneca_220ml	0,88	0,88	1,00	0,1	1,0	0,38	0,49	1,00	0,1	0,1	0	1	1
faca_4	1,00	0,15	0,13	0,1	0,1	1,00	0,11	0,01	0,1	0,1	1	1	0
garrafa_Lyon	1,00	0,37	0,37	0,1	1,0	0,25	0,25	1,00	0,1	1,0	1	1	0
panela_10	0,88	1,00	1,00	0,1	1,0	1,00	0,15	0,15	0,1	0,1	1	0	0

- a₁, a₂, a₃: parâmetros de tamanho normalizados para o maior valor das dimensões das duas primitivas;
- ε₁, ε₂: parâmetros de forma originais;
- Tarefa: Utilizar (1) ou guardar (0);
- Primitiva: (1) Cabo da faca ou panela, Corpo da garrafa ou caneca, Bocal da taça;
(0) Haste da taça, Lâmina da faca, Gargalo da garrafa.
- Modo da pega: *Power* (0) ou *Precision* (1)

A base de dados completa utilizada na etapa de aprendizagem da RNA é apresentada no Apêndice A.

Em um algoritmo de auto-aprendizagem dedicado ao reconhecimento ou classificação de padrões, uma elevada quantidade de exemplos de treinamento é necessária para a obtenção de uma generalização satisfatória. Embora a coleta direta de informações de forma e tamanho para objetos reais seja um procedimento oneroso, uma média de 10 exemplares do mesmo objeto se mostrou suficiente para produzir um erro quadrático inferior a 1,0 %.

Métodos de auto-aprendizagem supervisionada exigem que o conjunto das primitivas envolvidas no processo de treinamento seja suficientemente representativo para o elenco dos potenciais objetos a serem reconhecidos pela rede neural, como também para aqueles itens geométricos que não participaram da etapa de treinamento. Conseqüentemente, a escolha dos objetos para a fase de treinamento é uma etapa fundamental para o sucesso da aplicação da RNA. A tabela 7.2 ilustra os oito objetos selecionados para representar os padrões de treinamento e validação, bem como as formas preênsis sugeridas para uma mão humana executar as tarefas desejadas.

É importante ressaltar que a primitiva destacada pelas linhas tracejadas indica as regiões típicas de pega adotadas por usuários quando realizam as tarefas consideradas. Em contrapartida, a seta indica a primitiva escolhida pelo sistema de auto-aprendizagem para a mesma ação.

O estágio de preparação das primitivas se inicia com os objetos originais da primeira coluna da tabela 7.3. Uma vez que os objetos originais não podem ser diretamente encaminhados à rede neural, uma segmentação manual desses itens é realizada e disposta na segunda coluna da mesma tabela. Na seqüência, as primitivas que compõem cada um dos objetos são modeladas e fornecidas à rede neural.

Tabela 7.2: Conjunto dos oito padrões utilizados no processo de aprendizagem da RNA. A seta indica a primitiva que deve ser escolhida pela RNA, enquanto as linhas tracejadas denotam a seleção de um ser humano para a mesma tarefa.

	Objeto/Tarefa	Forma preênsil
a.	 Utilizar	 Power
b.	 Guardar	 Precision
c.	 Utilizar	 Power
d.	 Utilizar	 Precision
e.	 Utilizar	 Precision
f.	 Guardar	 Precision
g.	 Utilizar	 Precision
h.	 Guardar	 Precision
i.	 Utilizar	 Power
j.	 Utilizar	 Precision
k.	 Guardar	 Precision
l.	 Utilizar	 Power

Tabela 7.3: Sequência de preparação dos padrões de entrada para o processo de aprendizagem e fases de testes da RNA.

	(a) Objeto original	(b) Segmentação manual	(c) Modelagem automática das primitivas
a.			
b.			
c.			
d.			
e.			
f.			
g.			
h.			

Uma exceção à modelagem de objetos baseados em duas primitivas é concedida à taça da tabela 7.3(e) para demonstrar a robustez do algoritmo de auto-aprendizagem em selecionar duas formas básicas semelhantes para realizar tarefas distintas empregando a mesma forma preênsil. Finalmente, a terceira coluna ilustra a modelagem automática realizada pelo programa de ajuste de superelipsóides desenvolvido na primeira parte deste trabalho.

Uma vez que os conjuntos de dados para o mesmo objeto são compostos somente pela variação do seu tamanho, apenas um exemplar de cada objeto foi modelado automaticamente. As demais variações foram obtidas manualmente, tendo em vista que o processo de representação por superelipsóides pode ser igualmente aplicado aos demais itens com resultados igualmente representativos.

Como alguns objetos podem ser requisitados de duas maneiras distintas (*guardar* ou *utilizar*), os conjuntos de parâmetros das taças, garrafas, xícaras e canecas foram replicados no processo de aprendizagem para acomodar essas duas possibilidades. Dessa maneira, um total de 136 padrões constituiu a base completa de aprendizagem da RNA. O Apêndice A apresenta esses dados, enquanto a normalização dos escalares de tamanho é mostrada no Apêndice B.

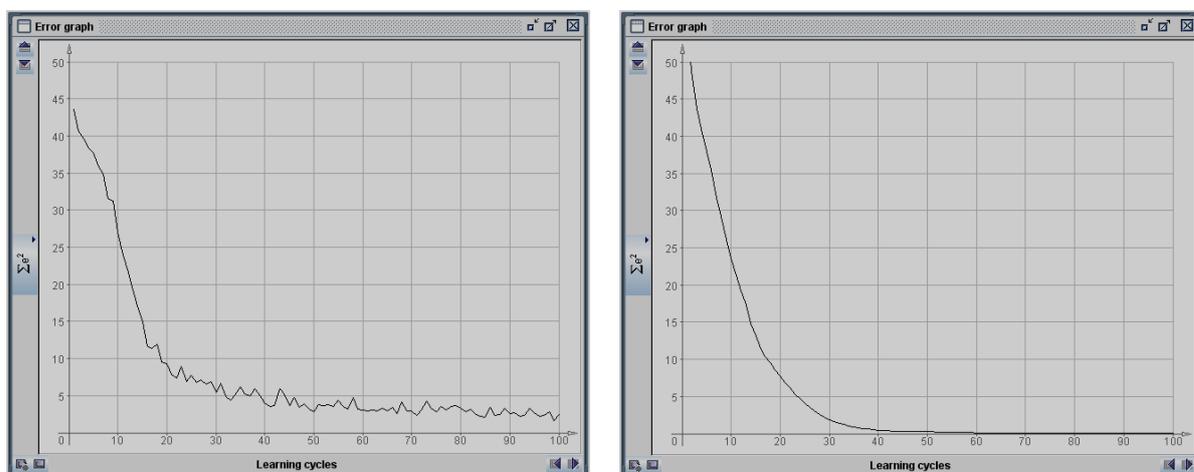
Finalmente, a partir de uma distribuição aleatória dos padrões de aprendizagem, 95 (70%) conjuntos de dados foram selecionados para compor a base de treinamento, enquanto os 41 (30%) padrões restantes formaram o arquivo de validação da rede neural artificial. Estes resultados estão apresentados no Apêndice C.

Para a fase de treinamento, o algoritmo *Backpropagation* (Haykin, 2001) convencional foi selecionado como o primeiro método de aprendizagem para a RNA, ao passo que as matrizes dos pesos sinápticos receberam valores aleatórios entre $-2,4/N$ e $2,4/N$ (ver LeCun, 1989), sendo que N representa a quantidade de neurônios da camada de entrada. Esta estratégia é adotada para tentar evitar que a soma ponderada dos valores das entradas pelos respectivos pesos implique na saturação das funções de ativação das camadas ocultas (Silva, *et al.*, 2010). No entanto, a instabilidade revelada pela curva de treinamento da figura 7.2 (a), associada a uma soma do erro quadrático igual a 1,43 para a fase de validação, impediram o uso deste método para os propósitos de aplicação deste trabalho.

Com o objetivo de tornar o processo de convergência da aprendizagem mais eficiente, o curto ciclo de treinamento (60 épocas) combinado a baixa soma do erro quadrático igual 0,37 para a fase de validação (ver figura 7.2(b)), conduziu à escolha do método *Resilient Propagation* (Haykin, 2001). O erro calculado para os dois métodos de aprendizagem é resultante do somatório do quadrado da diferença entre a resposta desejada (0 ou 1) e aquela produzida pela RNA. Uma vez que o somatório envolve valores pertencentes ao conjunto dos números reais o erro é adimensional.

7.2 Resultados experimentais

Com o encerramento das fases de treinamento e validação, diferentes experimentos foram conduzidos para avaliar a capacidade de generalização do sistema de auto-aprendizagem.



(a) Soma do erro quadrático da validação = 1,43. (b) Soma do erro quadrático da validação = 0,37.

Figura 7.2 – Comparação de desempenho dos melhores resultados para os algoritmos de aprendizagem: (a) *Backpropagation* e (b) *Resilient Propagation*, para a mesma topologia de rede.

É importante enfatizar que o critério adotado pelo sistema de auto-aprendizagem para sugerir a primitiva a ser pega considera ações tipicamente executada por usuários de acordo com o contexto. Para citar alguns exemplos, a

indicação da rede neural pelo gargalo de uma garrafa e da tampa de um pote de vidro, nesta ordem, sugere o desejo de alterar a posição desses objetos quando localizados em uma mesa repleta de outros itens transferindo-os, por exemplo, para o interior de um engradado de embalagens ou volumes restritos. Uma faca, “rodinho de cozinha”, “tábua de carne”, espátula, entre outros objetos similares, por sua vez, são pegos pelo cabo para possibilitar o desenvolvimento de tarefas rotineiras. Considerando outra categoria de objetos, uma garrafa, xícaras ou canecas são agarradas pelo seu corpo e alça, respectivamente, para os propósitos cotidianos de servir líquido a um usuário.

A primeira fase de experimentos para teste da RNA adota objetos pertencentes à mesma categoria de itens utilizados no processo de aprendizagem, porém com tamanhos diferentes, como ilustra a tabela 7.4. Esta tabela também ilustra a forma preênsil tipicamente adotada por humanos e aquela sugerida pela rede neural.

Uma vez que funções de ativação logística foram adotadas nos neurônios da camada de saída, o critério de arredondamento simétrico da equação 7.1 foi utilizado para o pós-processamento das saídas, y_i , apresentadas pela rede neural para os propósitos de análise dos comportamentos exibidos pelo sistema de auto-aprendizagem.

$$y_i^{pós} = \begin{cases} 1, & \text{se } y_i \geq 0,5 \\ 0, & \text{se } y_i < 0,5 \end{cases} \quad (7.1)$$

A figura 7.3 ilustra um fragmento do arquivo de saída e a interpretação considerada para avaliar os resultados apresentados pela rede neural.

A cerquilha localizada depois do cabeçalho inicia a exibição dos resultados para a garrafa térmica da tabela 7.4(a). Os dez escalares da primeira linha deste conjunto de informações indicam as entradas normalizadas para representar as primitivas, enquanto a décima primeira entrada fornece a tarefa desejada.

Para este caso específico, a ação de interesse consiste em *utilizar* a garrafa. As seqüências binárias das duas linhas subseqüentes correspondem, nesta ordem, à primitiva (1) e forma preênsil *power grasp* (0) selecionada por um agente humano e,

primitiva (0,999) e forma preênsil (0,01513) escolhidos pela RNA. A interpretação para esse resultado indica que ocorreu uma concordância de atitudes.

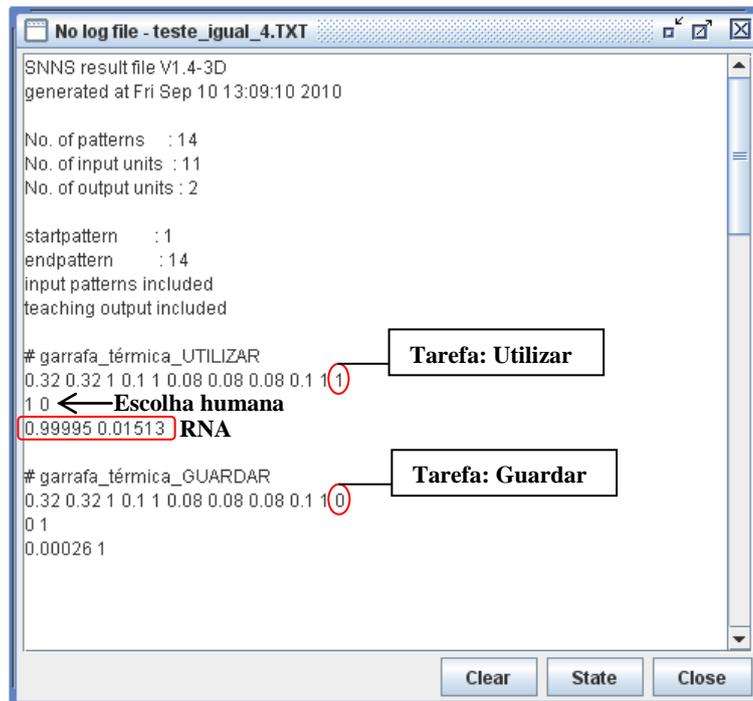


Figura 7.3 – Fragmento do arquivo de saída utilizado na interpretação do comportamento da rede neural artificial.

Quando a garrafa térmica foi submetida à ação *guardar* (ver tabela 7.4b), o segundo conjunto de informações indicou 0,0 e 1,0 para as escolhas da RNA, contra 0,0 e 1,0 esperados para a seleção da primitiva e forma preênsil humana, respectivamente, demonstrando resultados consistentes com aqueles esperados. Uma extensão da análise do comportamento da rede neural para os demais objetos é apresentada no Apêndice D. Para facilitar a interpretação dos resultados apresentados pela rede neural, as saídas desejadas também foram incluídas no arquivo de saída, porém elas são ignoradas pelo simulador.

Uma análise do comportamento da RNA demonstra a sua capacidade em selecionar corretamente as regiões tipicamente adotadas pelos seres humanos para os

14 contextos da tabela 7.4. Este resultado corresponde a uma taxa de acerto de 100%, considerando a escolha correta da primitiva bem como a forma preênsil adequada.

Este ensaio demonstra um aspecto preliminar motivador e de fundamental importância para a continuidade da proposta, pois ele revela a habilidade da rede neural em trabalhar com objetos similares àqueles adotados no processo de treinamento. Caso a conclusão fosse insatisfatória, seria prudente afirmar que as características dos conjuntos de treinamento não foram suficientemente representativas.

A segunda etapa de testes explorou objetos completamente diferentes daqueles utilizados no processo de treinamento da RNA. O propósito deste experimento consiste em avaliar a capacidade da rede neural em generalizar o aprendizado para novos objetos. O elenco formado pelos 22 objetos da tabela 7.5 (ver Apêndice E para descrição completa dos conjuntos de dados) representa a classe dos novos itens e seus respectivos modos de pega para cada tarefa.

Análogo ao primeiro ensaio, o sistema de auto-aprendizagem considera as aplicações típicas envolvendo objetos de uso cotidiano para sugerir a primitiva e a respectiva forma preênsil. Novamente, a seta indica a primitiva selecionada pela RNA, enquanto as linhas tracejadas revelam a escolha humana para a mesma ação e o marcador X registra os resultados incorretos, sejam eles relacionados à primitiva ou ao modo da pega.

Como os resultados mostrados na tabela 7.5 são auto-explicativos, somente aspectos relevantes para avaliar o desempenho do sistema de auto-aprendizagem são discutidos. As indicações da RNA para a forma preênsil *power grasp* sobre a seção cilíndrica alongada da lupa, lanterna e martelo das tabela 7.5 (a1), (b) e (q) revela a primeira consistência de resultados quando comparado às escolhas humanas para a mesma tarefa.

Tabela 7.4 – Objetos utilizados para demonstrar a capacidade da RNA em trabalhar com itens similares aqueles adotados no processo de treinamento. As linhas tracejadas indicam as regiões tipicamente escolhidas pelos seres humanos, enquanto a seta aponta a seleção da RNA para a mesma tarefa.

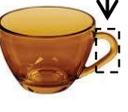
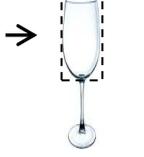
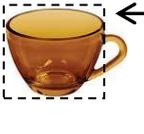
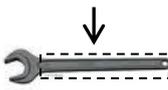
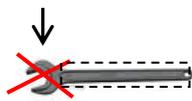
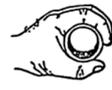
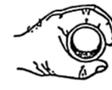
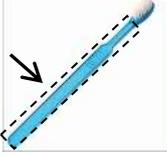
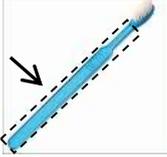
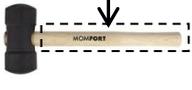
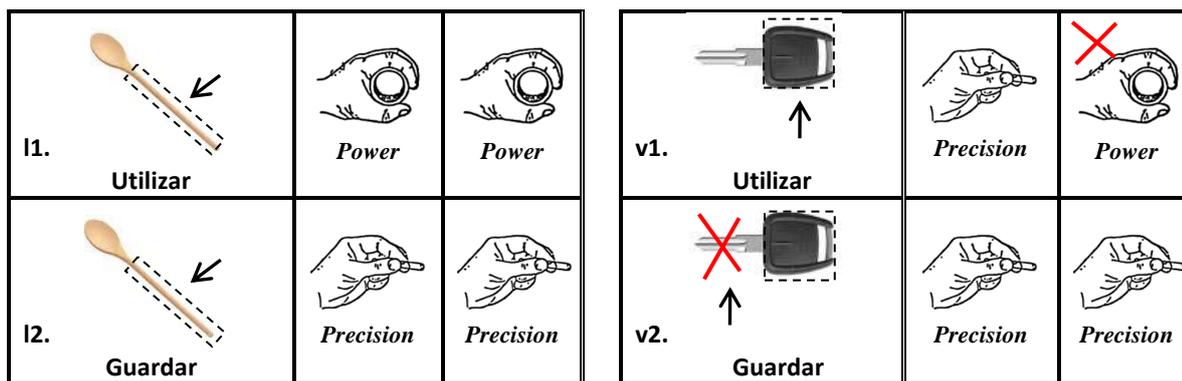
Objeto/Tarefa	Forma preênsil	
	Humano	RNA
a.  Utilizar	 <i>Power</i>	 <i>Power</i>
b.  Guardar	 <i>Precision</i>	 <i>Precision</i>
c.  Utilizar	 <i>Power</i>	 <i>Power</i>
d.  Guardar	 <i>Precision</i>	 <i>Precision</i>
e.  Utilizar	 <i>Power</i>	 <i>Power</i>
f.  Guardar	 <i>Precision</i>	 <i>Precision</i>
g.  Utilizar	 <i>Power</i>	 <i>Power</i>
h.  Utilizar	 <i>Power</i>	 <i>Power</i>
i.  Utilizar	 <i>Precision</i>	 <i>Precision</i>
j.  Utilizar	 <i>Precision</i>	 <i>Precision</i>
k.  Utilizar	 <i>Precision</i>	 <i>Precision</i>
l.  Guardar	 <i>Precision</i>	 <i>Precision</i>
m.  Utilizar	 <i>Precision</i>	 <i>Precision</i>
n.  Guardar	 <i>Precision</i>	 <i>Precision</i>

Tabela 7.5 – Objetos inéditos utilizados para testar a capacidade de generalização da RNA. A seta indica a escolha da rede neural, enquanto as linhas tracejadas denotam a região tipicamente selecionada pelos seres humanos para a mesma tarefa.

Objeto/Tarefa	Forma preênsil		Objeto/Tarefa	Forma preênsil	
	Humano	RNA		Humano	RNA
 a1. Utilizar			 m1. Utilizar		
 a2. Guardar			 m2. Guardar		
 b. Utilizar			 n1. Utilizar		
 c1. Utilizar			 n2. Guardar		
 c2. Guardar			 o. Utilizar		
 d. Utilizar			 p1. Utilizar		
 e1. Utilizar			 p2. Guardar		

e2.  Guardar	 Precision	 Precision
f.  Utilizar	 Power	 Power
g1.  Utilizar	 Power	 Precision 
g2.  Guardar	 Precision	 Precision
h.  Utilizar	 Power	 Power
i.  Utilizar	 Power	 Power
j.  Utilizar	 Power	 Power
k1.  Utilizar	 Precision	 Precision
k2.  Guardar	 Precision	 Precision

q.  Utilizar	 Power	 Power
r1.  Utilizar	 Precision	 Precision
r2.  Guardar	 Precision	 Precision
s1.  Utilizar	 Precision	 Precision
s2.  Guardar	 Precision	 Precision
t1.  Utilizar	 Power	 Power
t2.  Guardar	 Precision	 Precision
u1.  Utilizar	 Power	 Precision 
u2.  Guardar	 Precision	 Precision



No entanto, o primeiro resultado insatisfatório produzido pela rede neural corresponde à indicação da lente da lupa da tabela 7.5 (a2) como região adequada para *guardar* o objeto.

Para os objetos modelados por um cilindro e um paralelepípedo (ver tabela 7.5 (m1), (n1), (n2), (p1), (p2) e (u2)), as primitivas e as formas preênsis sugeridas pela RNA foram compatíveis com aquelas adotadas por agentes humanos, com exceção à forma *precision grasp* inadequada para o uso do “creme dental” da tabela 7.5 (u1).

O modo *power grasp*, tipicamente adotado por humanos para *utilizar* a faca, espátula, pá de lixo, escova de dente, espátula de cozinha, “tábua de carne” e a machadinha da tabela 7.5 (c1), (c2), (e1), (e2), (f), (g1), (g2), (h), (i) e (o), também representou a escolha correta da RNA para as respectivas primitivas. Apenas o modo *precision grasp* sugerido pela rede neural artificial para *utilizar* a escova de dente da tabela 7.5 (g1) revelou a segunda escolha incorreta do sistema de auto-aprendizagem para itens modelados por dois paralelepípedos. Em contrapartida, quando o interesse consistir em *utilizar* uma *pen drive* ou chave de ignição, o modo *precision grasp* deve ser adotado. No entanto, o terceiro e quarto erros da rede neural são expostos, nesta ordem, pela forma preênsil *power grasp* para *utilizar* a chave de ignição da tabela (v1) e a primitiva adequada da tabela (v2) para *guardar* o mesmo objeto.

Diferentemente da escolha humana pela forma *precision grasp* para *utilizar* taças de sobremesa, a tabela 7.5 (s1) e (s2) contabilizam mais dois resultados incorretos da RNA. Embora este objeto seja similar às taças de vinho utilizadas na base de treinamento da rede neural, o interesse em *utilizar* a taça de sobremesa por intermédio do seu bocal revela um paradoxo da aprendizagem, observando que a haste

alongada da taça de vinho da tabela 7.2 (e) foi adotada como primitiva correta para a mesma tarefa. Além disso, a indesejável seleção da RNA pela base volumosa da taça de sobremesa, independente da tarefa, é enfatizada pelas dimensões mínimas e máximas (ver Apêndice E) diferentes daquelas adotadas para as taças de vinho empregadas na fase de aprendizagem da rede neural.

O aspecto inédito dos objetos desconhecidos pela rede é caracterizado pela forma geométrica peculiar de algumas primitivas utilizadas na modelagem dos volumes parciais dos itens selecionados. Apenas para citar alguns exemplos, o cabo cilíndrico ($\varepsilon_1 = 0,1$ e $\varepsilon_2 = 1,0$) do rodinho, da colher de madeira e do ralador da tabela 7.5 (n1), (n2), (l1), (l2), (p1) e (p2), difere do padrão cúbico ($\varepsilon_1 = \varepsilon_2 = 0,1$) representativo de uma categoria similar de objetos empregada no estágio de treinamento para representar a natureza geométrica típica desses itens. Outras observações interessantes podem ser conduzidas para as elipsóides adotadas como primitivas representativas das seções côncavas das colheres da tabela 7.5 (k1), (k2), (l1) e (l2), região oval da raquete de tênis da tabela 6.5 (j) e forma esférica do microfone da tabela (t1) e (t2).

Uma análise do comportamento da rede neural envolvendo as 36 combinações (objeto *versus* tarefa) da tabela 7.5 demonstra que ela foi suficientemente genérica para trabalhar de maneira satisfatória com 25 objetos totalmente desconhecidos. Esse resultado corresponde a uma taxa de sucesso igual a 69,0 %, considerando a escolha correta da primitiva como também a forma preênsil adequada. Os 36 contextos foram adotados para formar uma base compatível de comparação com os resultados demonstrados por Sahbani e El-Khoury (2008).

Considerando somente a seleção correta da primitiva como critério de avaliação, o comportamento da RNA para os mesmos objetos inéditos revela uma taxa de acerto igual a 83%, o qual confirma que 30 das 36 primitivas foram escolhidas corretamente pelo sistema de auto-aprendizagem para atender as especificações das tarefas. Este resultado é comparável ao índice de sucesso de 80% da proposta de Sahbani e El-Khoury (2008) para a mesma quantidade de objetos. Uma vez que a

forma geométrica básica a ser agarrada representa um fator decisório para o processo de manipulação, este resultado é satisfatório para os propósitos preliminares de aplicação deste trabalho envolvendo objetos desconhecidos pelo sistema de auto-aprendizagem. Além disso, adotando-se apenas o modo correto de pega para testar a capacidade de generalização da RNA, o índice de aceitação se eleva para 91,0%. As informações utilizadas na interpretação dos resultados da tabela 7.5 são apresentadas no Apêndice F.

A implementação e execução de todas as fases de treinamento, validação e teste da RNA foram realizados no software gratuito JavaNNS versão 1.1 (Zell *et al.*, 2010).

Considerações finais

Este trabalho apresentou um método automático de modelagem de objetos sintéticos descritos por nuvens de pontos 3D baseado no ajuste de superelipsóides. As principais vantagens desta técnica são reveladas pelas atrativas propriedades de um modelo compacto, controlável, intuitivo e de fácil reconstrução geométrica a partir de apenas 11 parâmetros.

Tendo em vista que a acomodação de superelipsóides a nuvens de pontos caracteriza desafios típicos de mínimos quadrados, as principais propostas para as funções distância foram investigadas. O trabalho também contemplou um breve estudo sobre o método de Levenberg-Marquardt utilizado na minimização da função objetivo adotada. Para atender às necessidades da proposta, especial atenção foi dedicada à configuração da implementação em C/C++ do algoritmo.

O desempenho e a eficiência do programa computacional desenvolvido foram demonstrados na modelagem experimental de quatro primitivas organizadas em diferentes configurações. As baixas distorções obtidas para os modelos simulados expressam a robustez da representação de formas elementares empregando superelipsóides sem deformações. Além disso, a coerência espacial e geométrica dos modelos obtidos, sobretudo em relação os parâmetros de forma e tamanho, também foi confirmada na modelagem de um paralelepípedo organizado em seis poses. Esses efeitos motivam a continuidade deste estudo, uma vez que tais termos serão interpretados por um sistema de auto-aprendizagem na assistência à manipulação de objetos de uso geral.

Dado o fato de que a natureza simétrica dos objetos presentes em cenários não controlados é predominante no elenco dos itens de uso cotidiano dos seres humanos, esta forma de justa proporção foi escolhida para as simulações. Todavia, a modelagem de objetos não simétricos é igualmente possível com a ampliação da capacidade de

representação das superelipsóides com deformações globais adicionadas aos modelos padrões.

Baseado nas informações de forma e tamanho das primitivas utilizadas na modelagem de objetos simples, uma rede neural artificial foi capaz de aprender a relacionar a escolha por formas básicas ao modo preênsil adequado para a execução de determinadas tarefas. A elevada taxa de acerto de 100% para objetos idênticos aqueles adotados na base de treinamento da RNA demonstrou que as características dos padrões utilizados foram suficientemente representativas.

Comportamento igualmente significativo foi observado quando a RNA foi capaz de generalizar o aprendizado para itens totalmente diferentes da base de treinamento. Uma taxa de acerto igual a 69% demonstrou a capacidade de aproveitamento do sistema de auto-aprendizagem em selecionar corretamente a primitiva a ser pega e a forma preênsil adequada para a realização de uma dada tarefa.

É importante enfatizar que o índice de sucesso igual a 83%, quando somente a seleção correta da primitiva de objetos inéditos é adotada como critério de avaliação, é comparável à taxa de acerto de 80% da proposta de El-Khoury e Sahbani (2008). Estendendo-se o conceito de interpretação da capacidade de generalização da RNA o elevado índice de aceitação igual a 90% descreve a escolha correta da forma preênsil pelo sistema de auto-aprendizagem.

Embora a especificação da tarefa tenha adotado um procedimento manual, a automatização desta escolha pode ser contemplada por um sistema de apoio suficientemente elaborado para identificar o contexto no qual o objeto está inserido.

A integração da presente proposta com uma generalização do método desenvolvido pelo grupo de trabalho do Laboratório de Mecatrônica da Escola de Engenharia de São Carlos (ver Pedro, 2008) para a fixação de objetos desconhecidos, pode oferecer reais possibilidades de manipulação de objetos 3D.

Finalmente, a capacidade da RNA em aprender a identificar características funcionais de objetos básicos para a execução de tarefas rotineiras demonstra um aspecto fundamental para o sucesso da inserção de assistentes robóticos móveis em cenários predominantemente humanos.

8.1 Contribuições do trabalho

Os aspectos relevantes e a contribuição original da presente pesquisa, nesta ordem, são expostos a seguir:

- configuração do método computacional de Levenberg-Marquardt para atender aos propósitos gerais de ajuste de superfícies 3D à nuvem de pontos;
- desenvolvimento e divulgação de um programa computacional capaz de modelar primitivas em poses genéricas;
- implementação de uma forma de parametrização uniforme da malha de representação de superfícies 3D;
- proposta de uma estratégia inovadora para pegadas de objetos baseada em redes neurais artificiais, a qual aprende a associar primitivas de objetos desconhecidos à formas preênses compatíveis com determinadas tarefas.

8.2 Sugestões para futuros trabalhos

- Teste do algoritmo de modelagem envolvendo objetos reais;
- Implementação de um método de segmentação automática;
- Inclusão da possibilidade da RNA interpretar objetos formados por 3 ou mais primitivas;
- Integração do método de fixação de objetos desconhecidos (Pedro, 2008) à pega de objetos baseada nas características funcionais de itens de uso cotidiano;
- Análises de pegadas no simulador, GraspIt!, para validação da proposta do sistema híbrido (fixação *versus* RNA);
- Reestruturação da mão robótica Kangüera para a execução do sistema híbrido;
- Encerramento das sugestões com a atuação do braço robótico Kuka utilizado para posicionar a mão Kangüera para o desenvolvimento pleno de manipulação de objetos.

REFERÊNCIAS

BARR, A. H.: **Superquadrics and Angle-Preserving Transformations**, *IEEE Computer Graphics and Applications*, vol.(1), pp. 11-22, 1981.

BARR, A. H.: **Global and Local Deformations of Solid Primitives**. *Computer Graphics, SIGGRAPH'84*, Vol. 18, Number 3, July, pp. 21- 30, 1984.

BARDINET, E.; COHEN, L. D.; AYACHE, N.: **A parametric deformable model to fit unstructured 3 d data**. Research report 2617 – INRIA, 1995.

BELINI, V. L.; MATSUO, D. T.; PEDRO, L. M.; STÜCHELI, M.; CAURIN, G. A. P.; PURQUERIO, B. M.: **Integrated System for the development and test of robot hands and grippers**. Proceedings of the XIII International Symposium on Dynamic Problems of Mechanics (DINAME 2009), ABCM, Angra dos Reis-RJ, Brazil, March 2nd - March 6th, 2009.

BHABHRAWALA, T. S.; KROVI, V. N.: **Shape Recovery From Medical Image Data Using Extended Superquadrics**. Proceedings of DETC'05 ASME 2005 Design Engineering Technical Conferences and Computers and Information in Engineering Conference Long Beach, California USA, September, pp. 24-28, 2005.

BHABHRAWALA, T. S.; KROVI, V. N.; MENDEL, F. C.; GOVINDARAJU, V.: **Shape recovery Using Extended Superquadrics**. Technical report, Department of Computer Science State University of New York at Buffalo, Buffalo, New York, pp. 1-93, 2007.

BIEDERMAN, I.: **Recognition-by-Components: A Theory of Human Image Understanding**. *Psychological Review*, vol. 94, pp. 115 – 147, 1987.

BORST, CH.; FISCHER, M.; HIRZINGER, G.: **Grasping the dice by dicing the grasp**. Proceedings of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems, Las Vegas, Nevada, pp. 3692 – 3697, October, 2003.

BORST, CH.; FISCHER, M.; HIRZINGER, G.: **Grasp planning: How to choose a suitable task wrench space**. In: Proceedings of IEEE International Conference on Robotics and Automation, 2004.

BOULT, T. E. ; GROSS, A. D.: **Recovery os superquadrics from depth information**, in Proceedings Spatial Reasoning and Multi-Sensor Fusion Workshop, St. Charles, IL, pp. 128-137, 1987.

BUTTERFASS, J.; FISCHER, M.; Grebenstein, M.; Haidacher, S.; Hirzinger, G.: **Design and Experiences with DLR Hand II**, Proceedings of the World Automation Congress 2004, Seville, Spain, 2004.

CHANG, L. Y.; SRINIVASA, S. S.; POLLARD, N. S.: **Planning pre-grasp manipulation for transport tasks**. IEEE International Conference on Robotics and Automation Anchorage Convention District, pp. 2697-2704, Alaska-USA, 2010.

CARROZZA, M. C.; DARIO, P.; VECCHI, F.; ROCCELLA, S.; ZECCA, M.; SEBASTIANI, F.: **The CyberHand: on the design of a cybernetic prosthetic hand intend to be interfaced to the peripheral nervous system**. IEEE/RSJ. Conference on Intelligent Robots and Systems. Proceedings of the 2003 IEEE/RSJ . Las Vegas, Nevada, 2003.

CIESLAK. P.: **Morphing and loading Objects from a file**. <http://nehe.gamedev.net/data/lessons/lesson.asp?lesson=25>. Acessado em 26/06/2010.

CHEVALIER, L.; JAILLET, F.; BASKURT, A.: **3D shape coding with superquadrics**. In *Proc. of IEEE International Conference on Image Processing ICIP-2001*, Thessaloniki, Grèce, Vol. 2, pp. 93-96, October, 2001.

CHEVALIER, L.; JAILLET, F.; BASKURT, A.: **Segmentation and superquadric modeling of 3D objects.** *Journal of Winter School of Computer Graphics, WSCG'03*, Pilsen, Rep. Tchèque, Vol. 11, N° 2, pp. 232-239, February, 2003.

CHEVALIER, L.: **Modélisation et indexation d'objets 3D à l'aide de superellipsoïdes**, these de doctorado. Université Claude Bernard Lyon 1 – UFR informatique, 2004.

DOP, VAN ERIK; REGTIEN, P.: **Object recognition from range images using superquadric Representations.** MVA '96 IAPR Workshop on Machine Vision Applications, November, pp. 12-14, Tokyo, Japan, 1996.

DING, D.; LIU, Y. H.; WANG, S.: **Computing 3D optimal form-closure grasps.** Proc. IEEE Int. Conf. Robotics and Automation, 2000.

EKVALL, S.; KRAGIC, D.: **Interactive Grasp Learning Based on Human Demonstration.** In *IEEE International Conference on Robotics and Automation, ICRA'04*, New Orleans, USA, 2004.

EL-KHOURY, S.; SAHBANI, A.: **Handling Objects by Their Handles.** In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, WS - "grasp and task learning by imitation", pp. 58-64, Nice, France, 2008.

FERRIE, F.P.; LAGARDE, J.; WHAITE, P.: **Darboux Frames, Snakes, and Super-Quadrics: Geometry from the Bottom Up,** *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(8), 771-783, 1993.

FISCHER, M.; VAN DER SMAGT, P.; HIRZINGER, G.: **Learning techniques in a dataglove based telemanipulation system for the DLR hand.** In Proc. IEEE International Conference on Robotics and Automation, 1998.

GARDNER, M.: **Piet Hein's Superellipse. In Mathematical Carnival: A new Round-Up of Tantalizers and Puzzles** from Scientific American (Vintage, New York), Ch.18, pp.240–254, 1977.

GOLDFEDER, C.; ALLEN, P. K.; LACKNER, C.; PELOSOF, R.: **Grasp Planning via Decomposition Trees**. IEEE Int. Conference on Robotics and Automation, April 13, Rome, 2007.

GROSS, A.; BOULT, T.: **Error of fit measures for recovering parametric solids**. In: Internat. Conf. Computer Vision, 1988.

GUPTA, A.; BOGONI, L.; BAJCSY, R.: **Quantitative and Qualitative Measures For the Evaluation of the Superquadric Models**, Department of Computer & Information Science, Technical Reports (CIS), University of Pennsylvania, pp. 1-25, 1989.

GUPTA, A.; BAJCSY, R.: **Surface and volumetric segmentation of range images using biquadrics and superquadrics**. In: Internat. Conf. Pattern Recognition, Vol. 1, pp. 158 -162, 1992.

GUPTA, A.; BAJCSY, R.: **Volumetric Segmentation of Range Images of 3-D Objects Using Superquadric Models**. Computer Vision, Graphics, and Image Processing Image Understanding, vol. 58, no. 3, pp. 302-326, 1993.

HAYKIN, S.: **Redes neurais: princípios e prática**. 2^a Edição, Porto Alegre, Bookman, 2001.

HASCHKE, R.; STEIL, J.J.; STEUWER, I.; RITTER, H.: **Task-oriented quality measures for dexterous grasping**. Proceedings IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA, pp. 689 – 694, 2005.

HICKLIN, J.; MOLER, C.; WEBB, P.; BOISVERT, R. F.; MILLER, B.; POZO, R.; REMINGTON, K.: **JAMA: A Java Matrix Package**. <http://math.nist.gov/javanumerics/jama>. Acessado em 23/06/2010.

HUESER, M.; BAIER, T.; ZHANG, J.: **Learning of demonstrated grasping skills by stereoscopic tracking of human hand configuration**. In Proc. IEEE International Conference on Robotics and Automation, 2006.

JAKLIC, A.; LEONARDIS, A.; SOLINA, F.: **Superquadrics and their geometric properties**. **Book: Segmentation and Recovery of Superquadrics**, Ch. 2, pp. 13 - 39. Disponível em <http://lrv.fri.uni-lj.si/~franc/SRSbook/geometry.pdf> (acessado em 14/07/2010), 2000.

JAKLIC, A.; SOLINA, F.: **Moments of superellipsoids and their application to range image registration**. IEEE Transactions on Systems, Man, and Cybernetics, Part B 33(4): pp. 648-657, 2003.

KATSOULAS, D.: **Robust Recovery of Piled Box-Like Objects in Range Images**, tese de doutorado, 2004.

KYOTA, F.; WATABE, T.; SAITO, S.; NAKAJIMA, M.: **Detection and Evaluation of Grasping Positions for Autonomous Agents**. Proceedings of the International Conference on Cyberworlds, pp. 453 – 460, 2005.

KRIVIC, J.: **Segmentation and 3D tracking of superquadric modeled objects**. A dissertation in Computer and Information Science. Ljubljana, 2006.

LAMPTON, M.: **Damping-Undamping Strategies for the Levenverg-Marquardt Nonlinear Least-Squares Method**. Computers in Physics Journal, 11(1): 110-115, Jan./Feb, 1997.

LECUN, Y. **Generalization and network design strategies**. Technical Report CRG-TR-98-4, Department of Computer Science, University of Toronto, Canada, 1989.

LEONARDIS, A.; JAKLIČ, A.; SOLINA, F.: **Superquadrics for segmenting and modeling range data**. IEEE trans. pattern analysis and machine intelligence, vol. 19, no. 11, pp. 1269-1295, 1997.

LI, Z.; SASTRY, S.: **Task-Oriented optimal grasping by multifingered robot hands**. IEEE Journal of Robotics and Automation, Vol. 4, No. 1, pp. 32 -44, February, 1988.

LI, Y.; FU, J.L.; POLLARD, N.: **Data-driven grasp synthesis using shape matching and task-based pruning**. IEEE Transactions on Visualization and Computer Graphics. v13: (4), pp. 732-747, 2007.

LIU, Y. H.: **Qualitative test and force optimization of 3-D frictional form closure grasps using linear programming**. IEEE Transactions on Robotics and Automation, vol. 15, No. 1, 1999.

LOURAKIS, M. I. A.: **A Brief Description of the Levenberg-Marquardt Algorithm Implemented by levmar**. Institute of Computer Science Foundation for Research and Technology – Hellas (FORTH), Heraklion, Crete, Greece, February, 2005.

LOURAKIS, M.: **levmar: Levenberg-Marquardt nonlinear least squares algorithms in C/C++**. <http://www.ics.forth.gr/~lourakis/levmar/>. Acessado em 22/06/2010.

MARQUARDT, D.: **An Algorithm for Least-Squares Estimation of Nonlinear Parameters**. SIAM J. Appl. Math., (11): p. 431-441, 1963.

MARTIN, T. B.; AMBROSE, R. O.; DIFTLER, M. A.; PLATT, R.; BUTZER, M. J.: **Tactile Gloves for Autonomous Grasping with the NASA/DARPA Robonaut**. Proceedings of the 2004 IEEE International Conference on Robotics & Automation, New Orleans, LA, April, pp. 1713 – 1718, 2004.

MOURI, T.; KAWASAKI, H.; YOSHIKAWA, K.; TAKAI, J.; ITO, S.: **Anthropomorphic Robot Hand: Gifu Hand III**. Proc. of Int. Conf. ICCAS2002, pp. 1288-1293, 2002.

MOURI, T.; KAWASAKI, H.; UMEBAYASHI, K.: **Developments of New Anthropomorphic Robot Hand and its Master Slave System**. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3474 – 3479, 2005.

MIRTICH, B.; CANNY, J.: **Easily Computable Optimum Grasps in 2-D and 3-D**. Proc. IEEE Int. Conf. Robotics and Automation, vol. 1, pp. 739 – 747, 1994.

NGUYEN, V. D.: **Constructing Stable Grasps in 3D**. Proc. of Int. Conf. on Robotics and Automation, pp. 234 – 239, 1987.

OZTOP, E.; ARBIB, M. A.: **Schema design and implementation of the grasp-related mirror neuron system**. Biological Cybernetics. v87: (2), pp. 116-140, 2002.

PEDRO, L. M.: **Fixação de objetos desconhecidos por garras de robôs: estudo de aplicações de redes neurais e proposta de um algoritmo de auto-aprendizagem para os casos 2D e 3D**. Dissertação de mestrado, EESC-USP, 2008.

PELOSSOF, R.; MILLER, A.; ALLEN, P.; JEBARA, T.: **An SVM learning approach to robotic grasping**. In: Proceedings of IEEE International Conference on Robotics and Automation, 2004.

PENTLAND, A. P.: **Automatic Extraction of Deformable Part Models**. Int'l J. Computer Vision, vol. 4, pp. 107-126, 1990.

PENTLAND, A. P.; SCLAROFF, S.: **Closed-Form Solutions for Physically Based Shape Modeling and Recognition.** IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 13, no. 7, pp. 715-729, July, 1991.

PENTLAND, A.P.: **Perceptual Organization and the Representation of Natural Form.** Artificial Intelligence, vol. 28, no. 2, pp. 293-331, 1986.

PENTLAND, A. P.: **Recognition by Parts.** Proc. First Int'l Conf. Computer Vision, pp. 612-620, London, 1987.

POLLARD, N.S.: **Parallel algorithms for synthesis of whole-hand grasps.** In: Proceedings of IEEE International Conference on Robotics and Automation, 1997.

POPOVIC, M.; KRAFT, D.; BODENHAGEN; ET AL.: **A strategy for grasping unknown objects based on co-planarity and colour information.** Robotics and Autonomous Systems 58, pp. 551-565, 2010.

PRATS, M.; SANZ, P.J.; DEL POBIL A.P.: **Task-oriented grasping using hand preshapes and task frames.** In Proceedings of IEEE International Conference on Robotics and Automation, Roma, Italy, pp. 10 – 14, April 2007.

PRESS, W. H.; TEUKOLSKY, S. A.; VETTERLING, W. T.; FLANERY, B. P.: **Numerical Recipes – Example Book [C].** Cambridge University Press, Second Edition, 1999.

SAHBANI, A.; EL-KHOURY, S.: **A hybrid approach for grasping 3D objects.** IROS 2009: pp. 1272-1277, 2009.

SAXENA, A.; DRIEMEYER, J.; KEARNS, J.; NG, A.Y.: **Robotic grasping of novel objects using vision.** The International Journal of Robotics Research. v27(2), pp. 157-173, 2008.

SILVA, I. N.; SPATTI, D. H. FLAUZINO, R. A.: **Redes Neurais Artificiais para engenharia e ciências aplicadas: curso prático**. Artliber Editora, 2010.

SOLINA, F.; BAJCSY, R.: **Recovery of parametric models from range images: the case for superquadrics with global deformations**. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 12(2): pp. 131-147, 1990.

STILMAN, M.: **Global Manipulation Planning in Robot Joint Space with Task Constraints**. IEEE Transactions on Robotics, vol. 26, No. 3, pp. 576-584, June 2010.

STRAND, M.; DILLMANN, R.: **Segmentation and Approximation of Objects in Pointclouds using Superquadrics**. Proceedings of the 2009 IEEE International Conference on Information and Automation, Zhulai/Macau, China, June, pp. 22-25, 2009.

TERZOPOULOS, D.; METAXAS, D.: **Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics**. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 13, no. 7, pp. 703-714, July 1991.

WHAITE, P.; FERRIE, F.: **From Uncertainty to Visual Exploration**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13 (10): pp. 1038-1049, October, 1991.

WOLFE, J.; MARTHI, B.; RUSSELL, S.: **Combined Task and Motion Planning for Mobile Manipulation**. Proceedings of the Twentieth International Conference on Automated Planning and Scheduling (ICAPS), pp. 254 – 257, 2010.

ZELL, A.; MACHE, N.; MAMIER, G.; VOGT, M. *et al.* **JavaNNS**. <http://www.ra.cs.uni-tuebingen.de/SNNS/>. Acessado em 09/10/2010.

ZHANG, Y.: **Experimental comparison of superquadric fitting objective functions**. Pattern Recognition Letters, 24: pp. 2185–2193, 2003a.

ZHANG, Y.; KOSCHAN, A.; ABIDI, M.: **Superquadrics Based 3D Object Representation of Automotive Parts Utilizing Part Decomposition**. Proc. of SPIE 6th International Conference on Quality Control by Artificial Vision, TN, Vol. 5132, Gatlinburg, May, pp. 241-251, 2003b.

ZHOU, L.; KAMBHAMETTU, C.: **Extending Superquadrics with Exponent Functions: Modeling and Reconstruction**. Graphical Models, Vol. 63-1, pp. 1-20, January, 2000.

ZHU, X.; WANG, J.: **Synthesis of Force-Closure Grasps on 3D Objects Based on the Q Distance**. IEEE Transactions on Robotics and Automation, vol. 19, No. 4, 2003.

Apêndice A - Conjuntos de dados NÃO NORMALIZADOS utilizados no treinamento e validação da RNA.

TREINAMENTO e VALIDAÇÃO														SAÍDAS DESEJADAS		
ENTRADA																
	Parâmetros		a1 (mm)	a2 (mm)	a3 (mm)	E1	E2	a1 (mm)	a2 (mm)	a3 (mm)	E1	E2	TAREFA		PRIMITIVA	FORMA PREËNSIL
	OBJETO		Bocal/Corpo: 1				Alça/Haste/Gargalo/Cabo: 0									
1	xícara_70ml		60	60	50	0,1	1,0	10	25	45	0,1	0,1	1		0	1
2	xícara_70ml		60	60	50	0,1	1,0	10	25	45	0,1	0,1	0		1	1
3	xícara_140		80	80	60	0,1	1,0	11	26	42	0,1	0,1	1		0	1
4	xícara_140		80	80	60	0,1	1,0	11	26	42	0,1	0,1	0		1	1
5	xícara_200ml		80	80	80	0,1	1,0	12	27	43	0,1	0,1	1		0	1
6	xícara_200ml		80	80	80	0,1	1,0	12	27	43	0,1	0,1	0		1	1
7	xícara_60ml		50	50	70	0,1	1,0	11	25	43	0,1	0,1	1		0	1
8	xícara_60ml		50	50	70	0,1	1,0	11	25	43	0,1	0,1	0		1	1
9	xícara_80ml		60	60	60	0,1	1,0	12,8	25	38	0,1	0,1	1		0	1
10	xícara_80ml		60	60	60	0,1	1,0	12,8	25	38	0,1	0,1	0		1	1
11	xícara_210ml		90	90	80	0,1	1,0	10	35	44	0,1	0,1	1		0	1
12	xícara_210ml		90	90	80	0,1	1,0	10	35	44	0,1	0,1	0		1	1
13	caneca_300ml		80	80	90	0,1	1,0	15	34	50	0,1	0,1	1		0	1
14	caneca_300ml		80	80	90	0,1	1,0	15	34	50	0,1	0,1	0		1	1
15	caneca_160ml		65	65	80	0,1	1,0	18	25	40	0,1	0,1	1		0	1
16	caneca_160ml		65	65	80	0,1	1,0	18	25	40	0,1	0,1	0		1	1

17	caneca_350ml		90	90	110	0,1	0,1	19	33	39	0,1	0,1	1		0	1
18	caneca_350ml		90	90	110	0,1	0,1	19	33	39	0,1	0,1	0		1	1
19	caneca_220ml		75	75	85	0,1	1,0	17	22	45	0,1	0,1	1		0	1
20	caneca_220ml		75	75	85	0,1	1,0	17	22	45	0,1	0,1	0		1	1
21	caneca_430ml		75	75	150	0,1	1,0	14	35	41	0,1	0,1	1		0	1
22	caneca_430ml		75	75	150	0,1	1,0	14	35	41	0,1	0,1	0		1	1
23	caneca_410ml		85	85	140	0,1	1,0	14	34	55	0,1	0,1	1		0	1
24	caneca_410ml		85	85	140	0,1	1,0	14	34	55	0,1	0,1	0		1	1
25	caneca_475ml		76	76	135	0,1	1,0	14	37	60	0,1	0,1	1		0	1
26	caneca_475ml		76	76	135	0,1	1,0	14	37	60	0,1	0,1	0		1	1
27	caneca_465ml		85	85	120	0,1	1,0	14	45	65	0,1	0,1	1		0	1
28	caneca_465ml		85	85	120	0,1	1,0	14	45	65	0,1	0,1	0		1	1
29	caneca_500ml		115	115	85	0,1	1,0	14	45	65	0,1	0,1	1		0	1
30	caneca_500ml		115	115	85	0,1	1,0	14	45	65	0,1	0,1	0		1	1
31	taça_250ml		77	77	65	0,1	1,0	10	10	67	0,1	1,0	1		0	1
32	taça_250ml		77	77	65	0,1	1,0	10	10	67	0,1	1,0	0		1	1
33	taça_186ml		43	43	120	0,1	1,0	12	12	80	0,1	1,0	1		0	1
34	taça_186ml		43	43	120	0,1	1,0	12	12	80	0,1	1,0	0		1	1
35	taça_300ml		73	73	80	0,1	1,0	11	11	81	0,1	1,0	1		0	1
36	taça_300ml		73	73	80	0,1	1,0	11	11	81	0,1	1,0	0		1	1
37	taça_263ml		69	69	85	0,1	1,0	10	10	72	0,1	1,0	1		0	1

38	taça_263ml		69	69	85	0,1	1,0	10	10	72	0,1	1,0	0		1	1
39	taça_181ml		69	69	72	0,1	1,0	11	11	70	0,1	1,0	1		0	1
40	taça_181ml		69	69	72	0,1	1,0	11	11	70	0,1	1,0	0		1	1
41	taça_6		55	55	80	0,1	1,0	15	15	100	0,1	1,0	1		0	1
42	taça_6		55	55	80	0,1	1,0	15	15	100	0,1	1,0	0		1	1
43	taça_7		62	62	85	0,1	1,0	15	15	100	0,1	1,0	1		0	1
44	taça_7		62	62	85	0,1	1,0	15	15	100	0,1	1,0	0		1	1
45	taça_8		40	40	70	0,1	1,0	13,5	13,5	80	0,1	1,0	1		0	1
46	taça_8		40	40	70	0,1	1,0	13,5	13,5	80	0,1	1,0	0		1	1
47	taça_9		55	55	85	0,1	1,0	11	11	80	0,1	1,0	1		0	1
48	taça_9		55	55	85	0,1	1,0	11	11	80	0,1	1,0	0		1	1
49	taça_10		70	70	90	0,1	1,0	14	14	90	0,1	1,0	1		0	1
50	taça_10		70	70	90	0,1	1,0	14	14	90	0,1	1,0	0		1	1
51	garrafa_Lyon		74	74	200	0,1	1,0	22,5	22,5	90	0,1	1,0	1		1	0
52	garrafa_Lyon		74	74	200	0,1	1,0	22,5	22,5	90	0,1	1,0	0		0	1
53	garrafa_Lyon_2		82	82	210	0,1	1,0	17,8	17,8	85	0,1	1,0	1		1	0
54	garrafa_Lyon_2		82	82	210	0,1	1,0	17,8	17,8	85	0,1	1,0	0		0	1
55	garrafa_Lyon_3		80	80	190	0,1	1,0	19,5	19,5	73	0,1	1,0	1		1	0
56	garrafa_Lyon_3		80	80	190	0,1	1,0	19,5	19,5	73	0,1	1,0	0		0	1
57	garrafa_600ml		90	90	220	0,1	1,0	25	25	75	0,1	1,0	1		1	0
58	garrafa_600ml		90	90	220	0,1	1,0	25	25	75	0,1	1,0	0		0	1
59	garrafa_4		90	90	220	0,1	1,0	30	30	45	0,1	1,0	1		1	0

60	garrafa_4		90	90	220	0,1	1,0	30	30	45	0,1	1,0	0		0	1
61	garrafa_5		100	100	240	0,1	1,0	22	22	40	0,1	1,0	1		1	0
62	garrafa_5		100	100	240	0,1	1,0	22	22	40	0,1	1,0	0		0	1
63	garrafa_6		100	100	260	0,1	1,0	33	33	40	0,1	1,0	1		1	0
64	garrafa_6		100	100	260	0,1	1,0	33	33	40	0,1	1,0	0		0	1
65	garrafa_7		90	90	230	0,1	1,0	36	36	70	0,1	1,0	1		1	0
66	garrafa_7		90	90	230	0,1	1,0	36	36	70	0,1	1,0	0		0	1
67	garrafa_drink		53	53	122	0,1	1,0	28	28	60	0,1	1,0	1		1	0
68	garrafa_drink		53	53	122	0,1	1,0	28	28	60	0,1	1,0	0		0	1
69	garrafa_seduction		62	62	200	0,1	1,0	31	31	100	0,1	1,0	1		1	0
70	garrafa_seduction		62	62	200	0,1	1,0	31	31	100	0,1	1,0	0		0	1
71	garrafa_cachaça		72	72	210	0,1	1,0	31	31	70	0,1	1,0	1		1	0
72	garrafa_cachaça		72	72	210	0,1	1,0	31	31	70	0,1	1,0	0		0	1
73	pote_azeitona_1		86,5	86,5	136	0,1	1,0	74	74	20	0,1	1,0	1		1	0
74	pote_azeitona_1		86,5	86,5	136	0,1	1,0	74	74	20	0,1	1,0	0		0	1
75	pote_azeitona_2		68,7	68,7	100	0,1	1,0	63	63	10	0,1	1,0	1		1	0
76	pote_azeitona_2		68,7	68,7	100	0,1	1,0	63	63	10	0,1	1,0	0		0	1
77	pote_azeitona_3		60	60	101	0,1	1,0	58	58	10	0,1	1,0	1		1	0
78	pote_azeitona_3		60	60	101	0,1	1,0	58	58	10	0,1	1,0	0		0	1
79	pote_azeitona_4		67,2	67,2	129	0,1	1,0	60	60	15	0,1	1,0	1		1	0
80	pote_azeitona_4		67,2	67,2	129	0,1	1,0	60	60	15	0,1	1,0	0		0	1

81	pote_zeitona_5		62	62	157	0,1	1,0	58	58	10	0,1	1,0	1		1	0
82	pote_zeitona_5		62	62	157	0,1	1,0	58	58	10	0,1	1,0	0		0	1
83	pote_zeitona_6		62	62	180	0,1	1,0	61	61	15	0,1	1,0	1		1	0
84	pote_zeitona_6		62	62	180	0,1	1,0	61	61	15	0,1	1,0	0		0	1
85	pote_zeitona_7		90,5	90,5	168	0,1	1,0	74	74	15	0,1	1,0	1		1	0
86	pote_zeitona_7		90,5	90,5	168	0,1	1,0	74	74	15	0,1	1,0	0		0	1
87	pote_zeitona_8		70,5	70,5	86,5	0,1	1,0	63	63	15	0,1	1,0	1		1	0
88	pote_zeitona_8		70,5	70,5	86,5	0,1	1,0	63	63	15	0,1	1,0	0		0	1
89	pote_zeitona_9		60	60	91,6	0,1	1,0	58	58	15	0,1	1,0	1		1	0
90	pote_zeitona_9		60	60	91,6	0,1	1,0	58	58	15	0,1	1,0	0		0	1
91	pote_zeitona_10		86,9	86,9	130	0,1	1,0	85	85	25	0,1	1,0	1		1	0
92	pote_zeitona_10		86,9	86,9	130	0,1	1,0	85	85	25	0,1	1,0	0		0	1
93	chave_fenda_1		35	35	150	0,1	1,0	6	6	200	0,1	1,0	1		1	1
94	chave_fenda_2		30	30	160	0,1	1,0	8	8	220	0,1	1,0	1		1	1
95	chave_fenda_3		30	30	120	0,1	1,0	6	6	250	0,1	1,0	1		1	1
96	chave_fenda_4		33	33	100	0,1	1,0	10	10	275	0,1	1,0	1		1	1
97	chave_fenda_5		35	35	150	0,1	1,0	10	10	270	0,1	1,0	1		1	1
98	chave_fenda_6		25	25	75	0,1	1,0	5	5	140	0,1	1,0	1		1	1
99	chave_fenda_7		25	25	80	0,1	1,0	3	3	160	0,1	1,0	1		1	1
100	chave_fenda_8		22	22	100	0,1	1,0	3,2	3,2	76	0,1	1,0	1		1	1
101	chave_fenda_9		27	27	120	0,1	1,0	5	5	102	0,1	1,0	1		1	1
102	chave_fenda_10		30	30	140	0,1	1,0	6	6	102	0,1	1,0	1		1	1

103	chave_fenda_11		30	30	150	0,1	1,0	8	8	153	0,1	1,0	1		1	1
104	chave_fenda_12		30	30	130	0,1	1,0	6	6	160	0,1	1,0	1		1	1
105	chave_fenda_13		25	25	100	0,1	1,0	2,5	2,5	120	0,1	1,0	1		1	1
106	chave_fenda_14		30	30	80	0,1	1,0	4	4	50	0,1	1,0	1		1	1
107	panela_1		205	205	54	0,1	1,0	30	30	155	0,1	0,1	1		0	0
108	panela_2		236	236	183	0,1	1,0	35	35	120	0,1	0,1	1		0	0
109	panela_3		174	174	146	0,1	1,0	30	30	180	0,1	0,1	1		0	0
110	panela_4		195	195	16	0,1	1,0	28	28	150	0,1	0,1	1		0	0
111	panela_5		153	153	129	0,1	1,0	33	33	120	0,1	0,1	1		0	0
112	panela_6		300	300	190	0,1	1,0	25	25	130	0,1	0,1	1		0	0
113	panela_7		190	190	148	0,1	1,0	35	35	180	0,1	0,1	1		0	0
114	panela_8		236	236	168	0,1	1,0	30	30	200	0,1	0,1	1		0	0
115	panela_9		220	220	180	0,1	1,0	35	35	180	0,1	0,1	1		0	0
116	panela_10		171	171	150	0,1	1,0	30	30	200	0,1	0,1	1		0	0
117	cutelo_1		19	25	120	0,1	0,1	3	105	225	0,1	0,1	1		1	0
118	cutelo_2		20	20	167	0,1	0,1	4	147	200	0,1	0,1	1		1	0
119	cutelo_3		15	15	100	0,1	0,1	3	55	160	0,1	0,1	1		1	0
120	cutelo_4		20	25	140	0,1	0,1	1,8	100	175	0,1	0,1	1		1	0
121	cutelo_5		15	20	150	0,1	0,1	5	70	185	0,1	0,1	1		1	0
122	cutelo_6		25	25	120	0,1	0,1	3	85	170	0,1	0,1	1		1	0
123	cutelo_7		20	20	140	0,1	0,1	2,5	70	150	0,1	0,1	1		1	0

124	cutelo_8		21	21	120	0,1	0,1	3	80	140	0,1	0,1	1		1	0
125	cutelo_9		22	30	170	0,1	0,1	4	120	254	0,1	0,1	1		1	0
126	cutelo_10		25	30	110	0,1	0,1	3,5	80	150	0,1	0,1	1		1	0
127	pincel_1		8	8	80	0,1	0,1	35	35	5	0,1	0,1	1		1	1
128	pincel_2		12	20	110	0,1	0,1	80	30	12	0,1	0,1	1		1	1
129	pincel_3		6	12	120	0,1	0,1	6	30	70	0,1	0,1	1		1	1
130	pincel_4		5	11	110	0,1	0,1	4,5	22	50	0,1	0,1	1		1	1
131	pincel_5		6	10	100	0,1	0,1	6	25	50	0,1	0,1	1		1	1
132	pincel_6		9,5	17	100	0,1	0,1	8	20	40	0,1	0,1	1		1	1
133	pincel_7		4	15	80	0,1	0,1	4	15	35	0,1	0,1	1		1	1
134	pincel_8		5	13	90	0,1	0,1	4	10	11	0,1	0,1	1		1	1
135	pincel_9		4	6	85	0,1	0,1	3	6	8	0,1	0,1	1		1	1
136	pincel_10		4	15	110	0,1	0,1	5	14	15	0,1	0,1	1		1	1

Apêndice B - Conjuntos de dados NORMALIZADOS utilizados no treinamento e validação da RNA.

ENTRADA														SAÍDAS DESEJADAS	
		a1	a2	a3	E1	E2	a1	a2	a3	E1	E2	TAREFA		PRIMITIVA	FORMA PREÊNSIL
	OBJETO														
1	xícara_70ml	1,00	1,00	0,83	0,1	1,0	0,17	0,42	0,75	0,1	0,1	1		0	1
2	xícara_70ml	1,00	1,00	0,83	0,1	1,0	0,17	0,42	0,75	0,1	0,1	0		1	1
3	xícara_140	1,00	1,00	0,75	0,1	1,0	0,14	0,33	0,53	0,1	0,1	1		0	1
4	xícara_140	1,00	1,00	0,75	0,1	1,0	0,14	0,33	0,53	0,1	0,1	0		1	1
5	xícara_200ml	1,00	1,00	1,00	0,1	1,0	0,15	0,34	0,54	0,1	0,1	1		0	1
6	xícara_200ml	1,00	1,00	1,00	0,1	1,0	0,15	0,34	0,54	0,1	0,1	0		1	1
7	xícara_60ml	0,71	0,71	1,00	0,1	1,0	0,16	0,36	0,61	0,1	0,1	1		0	1
8	xícara_60ml	0,71	0,71	1,00	0,1	1,0	0,16	0,36	0,61	0,1	0,1	0		1	1
9	xícara_80ml	1,00	1,00	1,00	0,1	1,0	0,21	0,42	0,63	0,1	0,1	1		0	1
10	xícara_80ml	1,00	1,00	1,00	0,1	1,0	0,21	0,42	0,63	0,1	0,1	0		1	1
11	xícara_210ml	1,00	1,00	0,89	0,1	1,0	0,11	0,39	0,49	0,1	0,1	1		0	1
12	xícara_210ml	1,00	1,00	0,89	0,1	1,0	0,11	0,39	0,49	0,1	0,1	0		1	1
13	caneca_300ml	0,89	0,89	1,00	0,1	1,0	0,17	0,38	0,56	0,1	0,1	1		0	1
14	caneca_300ml	0,89	0,89	1,00	0,1	1,0	0,17	0,38	0,56	0,1	0,1	0		1	1
15	caneca_160ml	0,81	0,81	1,00	0,1	1,0	0,23	0,31	0,50	0,1	0,1	1		0	1
16	caneca_160ml	0,81	0,81	1,00	0,1	1,0	0,23	0,31	0,50	0,1	0,1	0		1	1
17	caneca_350ml	0,82	0,82	1,00	0,1	0,1	0,17	0,30	0,35	0,1	0,1	1		0	1

18	caneca_350ml		0,82	0,82	1,00	0,1	0,1	0,17	0,30	0,35	0,1	0,1	0		1	1
19	caneca_220ml		0,88	0,88	1,00	0,1	1,0	0,20	0,26	0,53	0,1	0,1	1		0	1
20	caneca_220ml		0,88	0,88	1,00	0,1	1,0	0,20	0,26	0,53	0,1	0,1	0		1	1
21	caneca_430ml		0,50	0,50	1,00	0,1	1,0	0,09	0,23	0,27	0,1	0,1	1		0	1
22	caneca_430ml		0,50	0,50	1,00	0,1	1,0	0,09	0,23	0,27	0,1	0,1	0		1	1
23	caneca_410ml		0,61	0,61	1,00	0,1	1,0	0,10	0,24	0,39	0,1	0,1	1		0	1
24	caneca_410ml		0,61	0,61	1,00	0,1	1,0	0,10	0,24	0,39	0,1	0,1	0		1	1
25	caneca_475ml		0,56	0,56	1,00	0,1	1,0	0,10	0,27	0,44	0,1	0,1	1		0	1
26	caneca_475ml		0,56	0,56	1,00	0,1	1,0	0,10	0,27	0,44	0,1	0,1	0		1	1
27	caneca_465ml		0,71	0,71	1,00	0,1	1,0	0,12	0,38	0,54	0,1	0,1	1		0	1
28	caneca_465ml		0,71	0,71	1,00	0,1	1,0	0,12	0,38	0,54	0,1	0,1	0		1	1
29	caneca_500ml		1,00	1,00	0,74	0,1	1,0	0,12	0,39	0,57	0,1	0,1	1		0	1
30	caneca_500ml		1,00	1,00	0,74	0,1	1,0	0,12	0,39	0,57	0,1	0,1	0		1	1
31	taça_250ml		1,00	1,00	0,84	0,1	1,0	0,13	0,13	0,87	0,1	1,0	1		0	1
32	taça_250ml		1,00	1,00	0,84	0,1	1,0	0,13	0,13	0,87	0,1	1,0	0		1	1
33	taça_186ml		0,36	0,36	1,00	0,1	1,0	0,10	0,10	0,67	0,1	1,0	1		0	1
34	taça_186ml		0,36	0,36	1,00	0,1	1,0	0,10	0,10	0,67	0,1	1,0	0		1	1
35	taça_300ml		0,90	0,90	0,99	0,1	1,0	0,14	0,14	1,00	0,1	1,0	1		0	1
36	taça_300ml		0,90	0,90	0,99	0,1	1,0	0,14	0,14	1,00	0,1	1,0	0		1	1
37	taça_263ml		0,81	0,81	1,00	0,1	1,0	0,12	0,12	0,85	0,1	1,0	1		0	1
38	taça_263ml		0,81	0,81	1,00	0,1	1,0	0,12	0,12	0,85	0,1	1,0	0		1	1
39	taça_181ml		0,96	0,96	1,00	0,1	1,0	0,15	0,15	0,97	0,1	1,0	1		0	1

40	taça_181ml		0,96	0,96	1,00	0,1	1,0	0,15	0,15	0,97	0,1	1,0	0		1	1
41	taça_6		0,55	0,55	0,80	0,1	1,0	0,15	0,15	1,00	0,1	1,0	1		0	1
42	taça_6		0,55	0,55	0,80	0,1	1,0	0,15	0,15	1,00	0,1	1,0	0		1	1
43	taça_7		0,62	0,62	0,85	0,1	1,0	0,15	0,15	1,00	0,1	1,0	1		0	1
44	taça_7		0,62	0,62	0,85	0,1	1,0	0,15	0,15	1,00	0,1	1,0	0		1	1
45	taça_8		0,50	0,50	0,88	0,1	1,0	0,17	0,17	1,00	0,1	1,0	1		0	1
46	taça_8		0,50	0,50	0,88	0,1	1,0	0,17	0,17	1,00	0,1	1,0	0		1	1
47	taça_9		0,65	0,65	1,00	0,1	1,0	0,13	0,13	0,94	0,1	1,0	1		0	1
48	taça_9		0,65	0,65	1,00	0,1	1,0	0,13	0,13	0,94	0,1	1,0	0		1	1
49	taça_10		0,78	0,78	1,00	0,1	1,0	0,16	0,16	1,00	0,1	1,0	1		0	1
50	taça_10		0,78	0,78	1,00	0,1	1,0	0,16	0,16	1,00	0,1	1,0	0		1	1
51	garrafa_Lyon		0,37	0,37	1,00	0,1	1,0	0,11	0,11	0,45	0,1	1,0	1		1	0
52	garrafa_Lyon		0,37	0,37	1,00	0,1	1,0	0,11	0,11	0,45	0,1	1,0	0		0	1
53	garrafa_Lyon_2		0,39	0,39	1,00	0,1	1,0	0,08	0,08	0,40	0,1	1,0	1		1	0
54	garrafa_Lyon_2		0,39	0,39	1,00	0,1	1,0	0,08	0,08	0,40	0,1	1,0	0		0	1
55	garrafa_Lyon_3		0,42	0,42	1,00	0,1	1,0	0,10	0,10	0,38	0,1	1,0	1		1	0
56	garrafa_Lyon_3		0,42	0,42	1,00	0,1	1,0	0,10	0,10	0,38	0,1	1,0	0		0	1
57	garrafa_600ml		0,41	0,41	1,00	0,1	1,0	0,11	0,11	0,34	0,1	1,0	1		1	0
58	garrafa_600ml		0,41	0,41	1,00	0,1	1,0	0,11	0,11	0,34	0,1	1,0	0		0	1
59	garrafa_4		0,41	0,41	1,00	0,1	1,0	0,14	0,14	0,20	0,1	1,0	1		1	0
60	garrafa_4		0,41	0,41	1,00	0,1	1,0	0,14	0,14	0,20	0,1	1,0	0		0	1
61	garrafa_5		0,42	0,42	1,00	0,1	1,0	0,09	0,09	0,17	0,1	1,0	1		1	0

62	garrafa_5		0,42	0,42	1,00	0,1	1,0	0,09	0,09	0,17	0,1	1,0	0		0	1
63	garrafa_6		0,38	0,38	1,00	0,1	1,0	0,13	0,13	0,15	0,1	1,0	1		1	0
64	garrafa_6		0,38	0,38	1,00	0,1	1,0	0,13	0,13	0,15	0,1	1,0	0		0	1
65	garrafa_7		0,39	0,39	1,00	0,1	1,0	0,16	0,16	0,30	0,1	1,0	1		1	0
66	garrafa_7		0,39	0,39	1,00	0,1	1,0	0,16	0,16	0,30	0,1	1,0	0		0	1
67	garrafa_drink		0,43	0,43	1,00	0,1	1,0	0,23	0,23	0,49	0,1	1,0	1		1	0
68	garrafa_drink		0,43	0,43	1,00	0,1	1,0	0,23	0,23	0,49	0,1	1,0	0		0	1
69	garrafa_seduction		0,31	0,31	1,00	0,1	1,0	0,16	0,16	0,50	0,1	1,0	1		1	0
70	garrafa_seduction		0,31	0,31	1,00	0,1	1,0	0,16	0,16	0,50	0,1	1,0	0		0	1
71	garrafa_cachaça		0,34	0,34	1,00	0,1	1,0	0,15	0,15	0,33	0,1	1,0	1		1	0
72	garrafa_cachaça		0,34	0,34	1,00	0,1	1,0	0,15	0,15	0,33	0,1	1,0	0		0	1
73	pote_azeitona_1		0,64	0,64	1,00	0,1	1,0	0,55	0,55	0,15	0,1	1,0	1		1	0
74	pote_azeitona_1		0,64	0,64	1,00	0,1	1,0	0,55	0,55	0,15	0,1	1,0	0		0	1
75	pote_azeitona_2		0,69	0,69	1,00	0,1	1,0	0,63	0,63	0,10	0,1	1,0	1		1	0
76	pote_azeitona_2		0,69	0,69	1,00	0,1	1,0	0,63	0,63	0,10	0,1	1,0	0		0	1
77	pote_azeitona_3		0,59	0,59	1,00	0,1	1,0	0,57	0,57	0,10	0,1	1,0	1		1	0
78	pote_azeitona_3		0,59	0,59	1,00	0,1	1,0	0,57	0,57	0,10	0,1	1,0	0		0	1
79	pote_azeitona_4		0,52	0,52	1,00	0,1	1,0	0,46	0,46	0,12	0,1	1,0	1		1	0
80	pote_azeitona_4		0,52	0,52	1,00	0,1	1,0	0,46	0,46	0,12	0,1	1,0	0		0	1
81	pote_azeitona_5		0,39	0,39	1,00	0,1	1,0	0,37	0,37	0,06	0,1	1,0	1		1	0
82	pote_azeitona_5		0,39	0,39	1,00	0,1	1,0	0,37	0,37	0,06	0,1	1,0	0		0	1
83	pote_azeitona_6		0,34	0,34	1,00	0,1	1,0	0,34	0,34	0,08	0,1	1,0	1		1	0

84	pote_azeitona_6		0,34	0,34	1,00	0,1	1,0	0,34	0,34	0,08	0,1	1,0	0		0	1
85	pote_azeitona_7		0,54	0,54	1,00	0,1	1,0	0,44	0,44	0,09	0,1	1,0	1		1	0
86	pote_azeitona_7		0,54	0,54	1,00	0,1	1,0	0,44	0,44	0,09	0,1	1,0	0		0	1
87	pote_azeitona_8		0,82	0,82	1,00	0,1	1,0	0,73	0,73	0,17	0,1	1,0	1		1	0
88	pote_azeitona_8		0,82	0,82	1,00	0,1	1,0	0,73	0,73	0,17	0,1	1,0	0		0	1
89	pote_azeitona_9		0,66	0,66	1,00	0,1	1,0	0,63	0,63	0,16	0,1	1,0	1		1	0
90	pote_azeitona_9		0,66	0,66	1,00	0,1	1,0	0,63	0,63	0,16	0,1	1,0	0		0	1
91	pote_azeitona_10		0,67	0,67	1,00	0,1	1,0	0,65	0,65	0,19	0,1	1,0	1		1	0
92	pote_azeitona_10		0,67	0,67	1,00	0,1	1,0	0,65	0,65	0,19	0,1	1,0	0		0	1
93	chave_fenda_1		0,18	0,18	0,75	0,1	1,0	0,03	0,03	1,00	0,1	1,0	1		1	1
94	chave_fenda_2		0,14	0,14	0,73	0,1	1,0	0,04	0,04	1,00	0,1	1,0	1		1	1
95	chave_fenda_3		0,12	0,12	0,48	0,1	1,0	0,02	0,02	1,00	0,1	1,0	1		1	1
96	chave_fenda_4		0,12	0,12	0,36	0,1	1,0	0,04	0,04	1,00	0,1	1,0	1		1	1
97	chave_fenda_5		0,13	0,13	0,56	0,1	1,0	0,04	0,04	1,00	0,1	1,0	1		1	1
98	chave_fenda_6		0,18	0,18	0,54	0,1	1,0	0,04	0,04	1,00	0,1	1,0	1		1	1
99	chave_fenda_7		0,16	0,16	0,50	0,1	1,0	0,02	0,02	1,00	0,1	1,0	1		1	1
100	chave_fenda_8		0,22	0,22	1,00	0,1	1,0	0,03	0,03	0,76	0,1	1,0	1		1	1
101	chave_fenda_9		0,23	0,23	1,00	0,1	1,0	0,04	0,04	0,85	0,1	1,0	1		1	1
102	chave_fenda_10		0,21	0,21	1,00	0,1	1,0	0,04	0,04	0,73	0,1	1,0	1		1	1
103	chave_fenda_11		0,20	0,20	0,98	0,1	1,0	0,05	0,05	1,00	0,1	1,0	1		1	1
104	chave_fenda_12		0,19	0,19	0,81	0,1	1,0	0,04	0,04	1,00	0,1	1,0	1		1	1
105	chave_fenda_13		0,21	0,21	0,83	0,1	1,0	0,02	0,02	1,00	0,1	1,0	1		1	1

106	chave_fenda_14		0,38	0,38	1,00	0,1	1,0	0,05	0,05	0,63	0,1	1,0	1		1	1
107	panela_1		1,00	1,00	0,26	0,1	1,0	0,15	0,15	0,76	0,1	0,1	1		0	0
108	panela_2		1,00	1,00	0,78	0,1	1,0	0,15	0,15	0,51	0,1	0,1	1		0	0
109	panela_3		0,97	0,97	0,81	0,1	1,0	0,17	0,17	1,00	0,1	0,1	1		0	0
110	panela_4		1,00	1,00	0,08	0,1	1,0	0,14	0,14	0,77	0,1	0,1	1		0	0
111	panela_5		1,00	1,00	0,84	0,1	1,0	0,22	0,22	0,78	0,1	0,1	1		0	0
112	panela_6		1,00	1,00	0,63	0,1	1,0	0,08	0,08	0,43	0,1	0,1	1		0	0
113	panela_7		1,00	1,00	0,78	0,1	1,0	0,18	0,18	0,95	0,1	0,1	1		0	0
114	panela_8		1,00	1,00	0,71	0,1	1,0	0,13	0,13	0,85	0,1	0,1	1		0	0
115	panela_9		1,00	1,00	0,82	0,1	1,0	0,16	0,16	0,82	0,1	0,1	1		0	0
116	panela_10		0,86	0,86	0,75	0,1	1,0	0,15	0,15	1,00	0,1	0,1	1		0	0
117	cutelo_1		0,08	0,11	0,53	0,1	1,0	0,01	0,47	1,00	0,1	0,1	1		1	0
118	cutelo_2		0,10	0,10	0,84	0,1	0,1	0,02	0,74	1,00	0,1	0,1	1		1	0
119	cutelo_3		0,09	0,09	0,63	0,1	1,0	0,02	0,34	1,00	0,1	0,1	1		1	0
120	cutelo_4		0,11	0,14	0,80	0,1	0,1	0,01	0,57	1,00	0,1	0,1	1		1	0
121	cutelo_5		0,08	0,11	0,81	0,1	0,1	0,03	0,38	1,00	0,1	0,1	1		1	0
122	cutelo_6		0,15	0,15	0,71	0,1	0,1	0,02	0,50	1,00	0,1	0,1	1		1	0
123	cutelo_7		0,13	0,13	0,93	0,1	1,0	0,02	0,47	1,00	0,1	0,1	1		1	0
124	cutelo_8		0,15	0,15	0,86	0,1	0,1	0,02	0,57	1,00	0,1	0,1	1		1	0
125	cutelo_9		0,09	0,12	0,67	0,1	0,1	0,02	0,47	1,00	0,1	0,1	1		1	0
126	cutelo_10		0,17	0,20	0,73	0,1	0,1	0,02	0,53	1,00	0,1	0,1	1		1	0
127	pincel_1		0,10	0,10	1,00	0,1	0,1	0,44	0,44	0,06	0,1	0,1	1		1	1

128	pincel_2		0,11	0,18	1,00	0,1	0,1	0,73	0,27	0,11	0,1	0,1	1		1	1
129	pincel_3		0,05	0,10	1,00	0,1	0,1	0,05	0,25	0,58	0,1	0,1	1		1	1
130	pincel_4		0,05	0,10	1,00	0,1	0,1	0,04	0,20	0,45	0,1	0,1	1		1	1
131	pincel_5		0,06	0,10	1,00	0,1	0,1	0,06	0,25	0,50	0,1	0,1	1		1	1
132	pincel_6		0,10	0,17	1,00	0,1	0,1	0,08	0,20	0,40	0,1	0,1	1		1	1
133	pincel_7		0,05	0,19	1,00	0,1	0,1	0,05	0,19	0,44	0,1	0,1	1		1	1
134	pincel_8		0,06	0,14	1,00	0,1	0,1	0,04	0,11	0,12	0,1	0,1	1		1	1
135	pincel_9		0,05	0,07	1,00	0,1	0,1	0,04	0,07	0,09	0,1	0,1	1		1	1
136	pincel_10		0,04	0,14	1,00	0,1	0,1	0,05	0,13	0,14	0,1	0,1	1		1	1

**Apêndice C - Conjuntos de dados normalizados e
ALEATORIAMENTE distribuídos para o processo de
aprendizagem da RNA.**

C1: Base de dados para o treinamento (95 objetos).

OBJETO	ENTRADA											SAÍDAS DESEJADAS	
	a1	a2	a3	E1	E2	a1	a2	a3	E1	E2	TAREFA	PRIMITIVA	FORMA PREËNSIL
chave_fenda_4	0,12	0,12	0,36	0,1	1,0	0,04	0,04	1,00	0,1	1,0	1	1	1
pote_azeitona_2	0,69	0,69	1,00	0,1	1,0	0,63	0,63	0,10	0,1	1,0	1	1	0
xícara_60ml	0,71	0,71	1,00	0,1	1,0	0,16	0,36	0,61	0,1	0,1	1	0	1
taça_181ml	0,96	0,96	1,00	0,1	1,0	0,15	0,15	0,97	0,1	1,0	1	0	1
caneca_160ml	0,81	0,81	1,00	0,1	1,0	0,23	0,31	0,50	0,1	0,1	0	1	1
chave_fenda_7	0,16	0,16	0,50	0,1	1,0	0,02	0,02	1,00	0,1	1,0	1	1	1
taça_250ml	1,00	1,00	0,84	0,1	1,0	0,13	0,13	0,87	0,1	1,0	0	1	1
cutelo_9	0,09	0,12	0,67	0,1	0,1	0,02	0,47	1,00	0,1	0,1	1	1	0
cutelo_10	0,17	0,20	0,73	0,1	0,1	0,02	0,53	1,00	0,1	0,1	1	1	0
pincel_4	0,05	0,10	1,00	0,1	0,1	0,04	0,20	0,45	0,1	0,1	1	1	1
garrafa_6	0,38	0,38	1,00	0,1	1,0	0,13	0,13	0,15	0,1	1,0	1	1	0
caneca_350ml	0,82	0,82	1,00	0,1	0,1	0,17	0,30	0,35	0,1	0,1	1	0	1
garrafa_5	0,42	0,42	1,00	0,1	1,0	0,09	0,09	0,17	0,1	1,0	1	1	0
pote_azeitona_5	0,39	0,39	1,00	0,1	1,0	0,37	0,37	0,06	0,1	1,0	0	0	1
pincel_7	0,05	0,19	1,00	0,1	0,1	0,05	0,19	0,44	0,1	0,1	1	1	1
taça_263ml	0,81	0,81	1,00	0,1	1,0	0,12	0,12	0,85	0,1	1,0	1	0	1

cutelo_1	0,08	0,11	0,53	0,1	1,0	0,01	0,47	1,00	0,1	0,1	1	1	0
pote_azeitona_7	0,54	0,54	1,00	0,1	1,0	0,44	0,44	0,09	0,1	1,0	1	1	0
caneca_350ml	0,82	0,82	1,00	0,1	0,1	0,17	0,30	0,35	0,1	0,1	0	1	1
pote_azeitona_6	0,34	0,34	1,00	0,1	1,0	0,34	0,34	0,08	0,1	1,0	1	1	0
caneca_465ml	0,71	0,71	1,00	0,1	1,0	0,12	0,38	0,54	0,1	0,1	0	1	1
garrafa_drink	0,43	0,43	1,00	0,1	1,0	0,23	0,23	0,49	0,1	1,0	1	1	0
caneca_475ml	0,56	0,56	1,00	0,1	1,0	0,10	0,27	0,44	0,1	0,1	0	1	1
taça_300ml	0,90	0,90	0,99	0,1	1,0	0,14	0,14	1,00	0,1	1,0	1	0	1
garrafa_Lyon	0,37	0,37	1,00	0,1	1,0	0,11	0,11	0,45	0,1	1,0	0	0	1
pote_azeitona_5	0,39	0,39	1,00	0,1	1,0	0,37	0,37	0,06	0,1	1,0	1	1	0
garrafa_cachaça	0,34	0,34	1,00	0,1	1,0	0,15	0,15	0,33	0,1	1,0	0	0	1
panela_3	0,97	0,97	0,81	0,1	1,0	0,17	0,17	1,00	0,1	0,1	1	0	0
garrafa_4	0,41	0,41	1,00	0,1	1,0	0,14	0,14	0,20	0,1	1,0	1	1	0
panela_7	1,00	1,00	0,78	0,1	1,0	0,18	0,18	0,95	0,1	0,1	1	0	0
pote_azeitona_4	0,52	0,52	1,00	0,1	1,0	0,46	0,46	0,12	0,1	1,0	0	0	1
caneca_160ml	0,81	0,81	1,00	0,1	1,0	0,23	0,31	0,50	0,1	0,1	1	0	1
caneca_410ml	0,61	0,61	1,00	0,1	1,0	0,10	0,24	0,39	0,1	0,1	1	0	1
garrafa_seduction	0,31	0,31	1,00	0,1	1,0	0,16	0,16	0,50	0,1	1,0	0	0	1
taça_300ml	0,90	0,90	0,99	0,1	1,0	0,14	0,14	1,00	0,1	1,0	0	1	1
panela_5	1,00	1,00	0,84	0,1	1,0	0,22	0,22	0,78	0,1	0,1	1	0	0
garrafa_4	0,41	0,41	1,00	0,1	1,0	0,14	0,14	0,20	0,1	1,0	0	0	1

chave_fenda_13	0,21	0,21	0,83	0,1	1,0	0,02	0,02	1,00	0,1	1,0	1	1	1
cutelo_2	0,10	0,10	0,84	0,1	0,1	0,02	0,74	1,00	0,1	0,1	1	1	0
xícara_60ml	0,71	0,71	1,00	0,1	1,0	0,16	0,36	0,61	0,1	0,1	0	1	1
pote_azeitona_8	0,82	0,82	1,00	0,1	1,0	0,73	0,73	0,17	0,1	1,0	0	0	1
caneca_220ml	0,88	0,88	1,00	0,1	1,0	0,20	0,26	0,53	0,1	0,1	1	0	1
taça_8	0,50	0,50	0,88	0,1	1,0	0,17	0,17	1,00	0,1	1,0	0	1	1
pincel_9	0,05	0,07	1,00	0,1	0,1	0,04	0,07	0,09	0,1	0,1	1	1	1
xícara_200ml	1,00	1,00	1,00	0,1	1,0	0,15	0,34	0,54	0,1	0,1	0	1	1
chave_fenda_14	0,38	0,38	1,00	0,1	1,0	0,05	0,05	0,63	0,1	1,0	1	1	1
garrafa_7	0,39	0,39	1,00	0,1	1,0	0,16	0,16	0,30	0,1	1,0	0	0	1
xícara_70ml	1,00	1,00	0,83	0,1	1,0	0,17	0,42	0,75	0,1	0,1	1	0	1
taça_263ml	0,81	0,81	1,00	0,1	1,0	0,12	0,12	0,85	0,1	1,0	0	1	1
pincel_1	0,10	0,10	1,00	0,1	0,1	0,44	0,44	0,06	0,1	0,1	1	1	1
taça_10	0,78	0,78	1,00	0,1	1,0	0,16	0,16	1,00	0,1	1,0	1	0	1
garrafa_drink	0,43	0,43	1,00	0,1	1,0	0,23	0,23	0,49	0,1	1,0	0	0	1
panela_8	1,00	1,00	0,71	0,1	1,0	0,13	0,13	0,85	0,1	0,1	1	0	0
caneca_300ml	0,89	0,89	1,00	0,1	1,0	0,17	0,38	0,56	0,1	0,1	0	1	1
xícara_140	1,00	1,00	0,75	0,1	1,0	0,14	0,33	0,53	0,1	0,1	1	0	1
panela_10	0,86	0,86	0,75	0,1	1,0	0,15	0,15	1,00	0,1	0,1	1	0	0
pote_azeitona_3	0,59	0,59	1,00	0,1	1,0	0,57	0,57	0,10	0,1	1,0	0	0	1
pote_azeitona_10	0,67	0,67	1,00	0,1	1,0	0,65	0,65	0,19	0,1	1,0	1	1	0

taça_9	0,65	0,65	1,00	0,1	1,0	0,13	0,13	0,94	0,1	1,0	1	0	1
garrafa_cachaça	0,34	0,34	1,00	0,1	1,0	0,15	0,15	0,33	0,1	1,0	1	1	0
caneca_410ml	0,61	0,61	1,00	0,1	1,0	0,10	0,24	0,39	0,1	0,1	0	1	1
garrafa_seduction	0,31	0,31	1,00	0,1	1,0	0,16	0,16	0,50	0,1	1,0	1	1	0
cutelo_3	0,09	0,09	0,63	0,1	1,0	0,02	0,34	1,00	0,1	0,1	1	1	0
pote_azeitona_6	0,34	0,34	1,00	0,1	1,0	0,34	0,34	0,08	0,1	1,0	0	0	1
chave_fenda_1	0,18	0,18	0,75	0,1	1,0	0,03	0,03	1,00	0,1	1,0	1	1	1
pincel_3	0,05	0,10	1,00	0,1	0,1	0,05	0,25	0,58	0,1	0,1	1	1	1
chave_fenda_11	0,20	0,20	0,98	0,1	1,0	0,05	0,05	1,00	0,1	1,0	1	1	1
caneca_465ml	0,71	0,71	1,00	0,1	1,0	0,12	0,38	0,54	0,1	0,1	1	0	1
taça_186ml	0,36	0,36	1,00	0,1	1,0	0,10	0,10	0,67	0,1	1,0	1	0	1
garrafa_Lyon_3	0,42	0,42	1,00	0,1	1,0	0,10	0,10	0,38	0,1	1,0	0	0	1
garrafa_Lyon_3	0,42	0,42	1,00	0,1	1,0	0,10	0,10	0,38	0,1	1,0	1	1	0
taça_7	0,62	0,62	0,85	0,1	1,0	0,15	0,15	1,00	0,1	1,0	1	0	1
caneca_500ml	1,00	1,00	0,74	0,1	1,0	0,12	0,39	0,57	0,1	0,1	0	1	1
panela_6	1,00	1,00	0,63	0,1	1,0	0,08	0,08	0,43	0,1	0,1	1	0	0
chave_fenda_8	0,22	0,22	1,00	0,1	1,0	0,03	0,03	0,76	0,1	1,0	1	1	1
xícara_200ml	1,00	1,00	1,00	0,1	1,0	0,15	0,34	0,54	0,1	0,1	1	0	1
pote_azeitona_7	0,54	0,54	1,00	0,1	1,0	0,44	0,44	0,09	0,1	1,0	0	0	1
garrafa_Lyon_2	0,39	0,39	1,00	0,1	1,0	0,08	0,08	0,40	0,1	1,0	1	1	0
garrafa_6	0,38	0,38	1,00	0,1	1,0	0,13	0,13	0,15	0,1	1,0	0	0	1

taça_9	0,65	0,65	1,00	0,1	1,0	0,13	0,13	0,94	0,1	1,0	0	1	1
pincel_5	0,06	0,10	1,00	0,1	0,1	0,06	0,25	0,50	0,1	0,1	1	1	1
panela_1	1,00	1,00	0,26	0,1	1,0	0,15	0,15	0,76	0,1	0,1	1	0	0
pote_azeitona_9	0,66	0,66	1,00	0,1	1,0	0,63	0,63	0,16	0,1	1,0	1	1	0
taça_181ml	0,96	0,96	1,00	0,1	1,0	0,15	0,15	0,97	0,1	1,0	0	1	1
chave_fenda_9	0,23	0,23	1,00	0,1	1,0	0,04	0,04	0,85	0,1	1,0	1	1	1
pote_azeitona_3	0,59	0,59	1,00	0,1	1,0	0,57	0,57	0,10	0,1	1,0	1	1	0
xícara_140	1,00	1,00	0,75	0,1	1,0	0,14	0,33	0,53	0,1	0,1	0	1	1
pote_azeitona_10	0,67	0,67	1,00	0,1	1,0	0,65	0,65	0,19	0,1	1,0	0	0	1
xícara_70ml	1,00	1,00	0,83	0,1	1,0	0,17	0,42	0,75	0,1	0,1	0	1	1
pincel_8	0,06	0,14	1,00	0,1	0,1	0,04	0,11	0,12	0,1	0,1	1	1	1
caneca_220ml	0,88	0,88	1,00	0,1	1,0	0,20	0,26	0,53	0,1	0,1	0	1	1
caneca_430ml	0,50	0,50	1,00	0,1	1,0	0,09	0,23	0,27	0,1	0,1	1	0	1
cutelo_7	0,13	0,13	0,93	0,1	1,0	0,02	0,47	1,00	0,1	0,1	1	1	0
taça_7	0,62	0,62	0,85	0,1	1,0	0,15	0,15	1,00	0,1	1,0	0	1	1
panela_9	1,00	1,00	0,82	0,1	1,0	0,16	0,16	0,82	0,1	0,1	1	0	0

C2: Base de dados para a validação (41 objetos).

taça_250ml		1,00	1,00	0,84	0,1	1,0	0,13	0,13	0,87	0,1	1,0	1		0	1
xícara_210ml		1,00	1,00	0,89	0,1	1,0	0,11	0,39	0,49	0,1	0,1	1		0	1
pincel_6		0,10	0,17	1,00	0,1	0,1	0,08	0,20	0,40	0,1	0,1	1		1	1
chave_fenda_10		0,21	0,21	1,00	0,1	1,0	0,04	0,04	0,73	0,1	1,0	1		1	1
panela_4		1,00	1,00	0,08	0,1	1,0	0,14	0,14	0,77	0,1	0,1	1		0	0
caneca_500ml		1,00	1,00	0,74	0,1	1,0	0,12	0,39	0,57	0,1	0,1	1		0	1
xícara_80ml		1,00	1,00	1,00	0,1	1,0	0,21	0,42	0,63	0,1	0,1	0		1	1
garrafa_600ml		0,41	0,41	1,00	0,1	1,0	0,11	0,11	0,34	0,1	1,0	0		0	1
cutelo_5		0,08	0,11	0,81	0,1	0,1	0,03	0,38	1,00	0,1	0,1	1		1	0
taça_186ml		0,36	0,36	1,00	0,1	1,0	0,10	0,10	0,67	0,1	1,0	0		1	1
taça_6		0,55	0,55	0,80	0,1	1,0	0,15	0,15	1,00	0,1	1,0	0		1	1
garrafa_5		0,42	0,42	1,00	0,1	1,0	0,09	0,09	0,17	0,1	1,0	0		0	1
taça_6		0,55	0,55	0,80	0,1	1,0	0,15	0,15	1,00	0,1	1,0	1		0	1
chave_fenda_3		0,12	0,12	0,48	0,1	1,0	0,02	0,02	1,00	0,1	1,0	1		1	1
garrafa_600ml		0,41	0,41	1,00	0,1	1,0	0,11	0,11	0,34	0,1	1,0	1		1	0
chave_fenda_5		0,13	0,13	0,56	0,1	1,0	0,04	0,04	1,00	0,1	1,0	1		1	1
cutelo_6		0,15	0,15	0,71	0,1	0,1	0,02	0,50	1,00	0,1	0,1	1		1	0
caneca_475ml		0,56	0,56	1,00	0,1	1,0	0,10	0,27	0,44	0,1	0,1	1		0	1
pincel_10		0,04	0,14	1,00	0,1	0,1	0,05	0,13	0,14	0,1	0,1	1		1	1
panela_2		1,00	1,00	0,78	0,1	1,0	0,15	0,15	0,51	0,1	0,1	1		0	0

taça_10		0,78	0,78	1,00	0,1	1,0	0,16	0,16	1,00	0,1	1,0	0		1	1
chave_fenda_6		0,18	0,18	0,54	0,1	1,0	0,04	0,04	1,00	0,1	1,0	1		1	1
xícara_80ml		1,00	1,00	1,00	0,1	1,0	0,21	0,42	0,63	0,1	0,1	1		0	1
pote_azeitona_2		0,69	0,69	1,00	0,1	1,0	0,63	0,63	0,10	0,1	1,0	0		0	1
garrafa_7		0,39	0,39	1,00	0,1	1,0	0,16	0,16	0,30	0,1	1,0	1		1	0
pote_azeitona_1		0,64	0,64	1,00	0,1	1,0	0,55	0,55	0,15	0,1	1,0	1		1	0
garrafa_Lyon_2		0,39	0,39	1,00	0,1	1,0	0,08	0,08	0,40	0,1	1,0	0		0	1
caneca_430ml		0,50	0,50	1,00	0,1	1,0	0,09	0,23	0,27	0,1	0,1	0		1	1
xícara_210ml		1,00	1,00	0,89	0,1	1,0	0,11	0,39	0,49	0,1	0,1	0		1	1
cutelo_8		0,15	0,15	0,86	0,1	0,1	0,02	0,57	1,00	0,1	0,1	1		1	0
taça_8		0,50	0,50	0,88	0,1	1,0	0,17	0,17	1,00	0,1	1,0	1		0	1
chave_fenda_2		0,14	0,14	0,73	0,1	1,0	0,04	0,04	1,00	0,1	1,0	1		1	1
pincel_2		0,11	0,18	1,00	0,1	0,1	0,73	0,27	0,11	0,1	0,1	1		1	1
cutelo_4		0,11	0,14	0,80	0,1	0,1	0,01	0,57	1,00	0,1	0,1	1		1	0
chave_fenda_12		0,19	0,19	0,81	0,1	1,0	0,04	0,04	1,00	0,1	1,0	1		1	1
caneca_300ml		0,89	0,89	1,00	0,1	1,0	0,17	0,38	0,56	0,1	0,1	1		0	1
pote_azeitona_1		0,64	0,64	1,00	0,1	1,0	0,55	0,55	0,15	0,1	1,0	0		0	1
pote_azeitona_4		0,52	0,52	1,00	0,1	1,0	0,46	0,46	0,12	0,1	1,0	1		1	0
garrafa_Lyon		0,37	0,37	1,00	0,1	1,0	0,11	0,11	0,45	0,1	1,0	1		1	0
pote_azeitona_9		0,66	0,66	1,00	0,1	1,0	0,63	0,63	0,16	0,1	1,0	0		0	1
pote_azeitona_8		0,82	0,82	1,00	0,1	1,0	0,73	0,73	0,17	0,1	1,0	1		1	0

Apêndice D - Conjuntos de dados utilizados para o TESTE da RNA envolvendo objetos pertencentes à mesma categoria daqueles utilizados no processo de aprendizagem.

D1: Dados NÃO normalizados

ENTRADA												SAÍDAS DESEJADAS		
	a1	a2	a3	E1	E2	a1	a2	a3	E1	E2	TAREFA		PRIMITIVA	MODO PREÊNSIL
OBJETO	Bocal/Cabo/Corpo: 1					Gargalo/lâmina/Tampa: 0								
garrafa_térmica	80,0	80,0	250,0	0,1	1,0	20,0	20,0	20,0	0,1	1,0	1		1	0
garrafa_térmica	80,0	80,0	250,0	0,1	1,0	20,0	20,0	20,0	0,1	1,0	0		0	1
faca_1	25	28	85	0,1	0,1	2,5	45	200	0,1	0,1	1		1	0
pote_azeitona_5	62,0	62,0	157,0	0,1	1,0	58,0	58,0	10,0	0,1	1,0	1		1	0
pote_azeitona_5	62,0	62,0	157,0	0,1	1,0	58,0	58,0	10,0	0,1	1,0	0		0	1
garrafa_cerveja	73,0	73,0	200,0	0,1	1,0	40,0	40,0	84,0	0,1	1,0	1		1	0
garrafa_cerveja	73,0	73,0	200,0	0,1	1,0	40,0	40,0	84,0	0,1	1,0	0		0	1
panela	150,0	150,0	130,0	0,1	1,0	30,0	30,0	120,0	0,1	0,1	1		0	0
faca_2	21	30	80	0,1	0,1	2	40	220	0,1	0,1	1		1	0
Pincel	6,0	12,0	120,0	0,1	0,1	6,0	30,0	70,0	0,1	0,1	1		1	1
chave_fenda_toco	25	25	45	0,1	1,0	3	3	80	0,1	1,0	1		1	1
caneca_330ml	73	73	120	0,1	1,0	12,8	31	46	0,1	0,1	1		0	1
caneca_330ml	73	73	120	0,1	1,0	12,8	31	46	0,1	0,1	0		1	1
taça_vinho_utilizar	62,0	62,0	95,0	0,1	1,0	12,5	12,5	85,0	0,1	1,0	1		0	1
taça_vinho_guardar	62,0	62,0	95,0	0,1	1,0	12,5	12,5	85,0	0,1	1,0	0		1	1

D2: Dados normalizados

ENTRADA												SAÍDAS DESEJADAS	
Parâmetros	a1	a2	a3	E1	E2	a1	a2	a3	E1	E2	TAREFA	PRIMITIVA	MODO PREËNSIL
OBJETO	Bocal/Corpo/cabo: 1					Gargalo/lâmina/cabeça/tampa/cerdas: 0							
garrafa_térmica	0,32	0,32	1,00	0,1	1,0	0,08	0,08	0,08	0,1	1,0	1	1	0
garrafa_térmica	0,32	0,32	1,00	0,1	1,0	0,08	0,08	0,08	0,1	1,0	0	0	1
faca_1	0,13	0,14	0,43	0,1	0,1	0,01	0,23	1,00	0,1	0,1	1	1	0
pote_azeitona_5	0,39	0,39	1,00	0,1	1,0	0,37	0,37	0,06	0,1	1,0	1	1	0
pote_azeitona_5	0,39	0,39	1,00	0,1	1,0	0,37	0,37	0,06	0,1	1,0	0	0	1
garrafa_cerveja	0,37	0,37	1,00	0,1	1,0	0,20	0,20	0,42	0,1	1,0	1	1	0
garrafa_cerveja	0,37	0,37	1,00	0,1	1,0	0,20	0,20	0,42	0,1	1,0	0	0	1
panela	1,00	1,00	0,87	0,1	1,0	0,20	0,20	0,80	0,1	0,1	1	0	0
faca_2	0,10	0,14	0,36	0,1	0,1	0,01	0,18	1,00	0,1	0,1	1	1	0
Pincel	0,05	0,10	1,00	0,1	0,1	0,05	0,25	0,58	0,1	0,1	1	1	1
chave_fenda_toco	0,31	0,31	0,56	0,1	1,0	0,04	0,04	1,00	0,1	1,0	1	1	1
caneca_330ml	0,61	0,61	1,00	0,1	1,0	0,11	0,26	0,38	0,1	0,1	1	0	1
caneca_330ml	0,61	0,61	1,00	0,1	1,0	0,11	0,26	0,38	0,1	0,1	0	1	1
taça_vinho_utilizar	0,65	0,65	1,00	0,1	1,0	0,13	0,13	0,89	0,1	1,0	1	0	1
taça_vinho_guardar	0,65	0,65	1,00	0,1	1,0	0,13	0,13	0,89	0,1	1,0	0	1	1

Apêndice E – Arquivo de resultados para o teste envolvendo objetos pertencentes à mesma categoria daqueles adotados na base de aprendizagem da RNA.

SNNS result file V1.4-3D
generated at Sun Sep 12 19:52:10 2010

No. of patterns : 14
No. of input units : 11
No. of output units : 2

startpattern : 1
endpattern : 14

input patterns included
teaching output included

#1.1
0.32 0.32 1 0.1 1 0.08 0.08 0.08 0.1 1 1
1 0 ← Escolha humana
0.99995 0.01513 RNA

Tarefa: Utilizar

#2.1
0.32 0.32 1 0.1 1 0.08 0.08 0.08 0.1 1 0
0 1
0.00026 1

Tarefa: Guardar

#3.1
0.13 0.14 0.43 0.1 0.1 0.01 0.23 1 0.1 0.1 1
1 0
0.95111 0.04132

#4.1
0.39 0.39 1 0.1 1 0.37 0.37 0.06 0.1 1 0
0 1
0.00018 0.99998

#5.1
0.37 0.37 1 0.1 1 0.2 0.2 0.42 0.1 1 1
1 0
0.99931 0.00082

#6.1

0.37 0.37 1 0.1 1 0.2 0.2 0.42 0.1 1 0

0 1

0.00035 1

#7.1

1 1 0.87 0.1 1 0.2 0.2 0.8 0.1 0.1 1

0 0

0.00018 0.00355

#8.1

0.1 0.14 0.36 0.1 0.1 0.01 0.18 1 0.1 0.1 1

1 0

0.94629 0.09848

#9.1

0.05 0.1 1 0.1 0.1 0.05 0.25 0.58 0.1 0.1 1

1 1

1 0.99255

#10.1

0.31 0.31 0.56 0.1 1 0.04 0.04 1 0.1 1 1

1 1

0.97332 0.99995

#11.1

0.61 0.61 1 0.1 1 0.11 0.26 0.38 0.1 0.1 1

0 1

0.00008 0.99941

#12.1

0.61 0.61 1 0.1 1 0.11 0.26 0.38 0.1 0.1 0

1 1

0.99238 0.99975

#13.1

0.65 0.65 1 0.1 1 0.13 0.13 0.89 0.1 1 1

0 1

0.00066 0.99962

#14.1

0.65 0.65 1 0.1 1 0.13 0.13 0.89 0.1 1 0

1 1

0.99623 0.99994

Apêndice F – Arquivo de saída dos resultados para o TESTE da RNA envolvendo itens de categorias diferentes daquelas utilizadas no processo de aprendizagem.

Observação: A inclusão das saídas desejadas é utilizada somente para facilitar a interpretação dos resultados. Elas não são consideradas na execução do teste.

SNNS result file V1.4-3D
generated at Sun Sep 12 19:56:35 2010

No. of patterns : 36
No. of input units : 11
No. of output units : 2

startpattern : 1
endpattern : 36

input patterns included
teaching output included

#1.1
0.67 0.67 0.13 0.1 1 0.11 0.11 1 0.1 1 **1**
0 1 ← Escolha humana
0.00015 0.96317 RNA

Tarefa: Utilizar

#2.1
0.67 0.67 0.13 0.1 1 0.11 0.11 1 0.1 1 **0**
0 1
0.99973 0.99933

Tarefa: Guardar

#3.1
0.21 0.21 1 0.1 1 0.29 0.29 0.29 0.1 1 1
1 0
0.99977 0.02619

#4.1
0.17 0.2 0.73 0.1 0.1 0.02 0.37 1 0.1 0.1 1
1 0
0.97877 0.25694

#5.1

0.17 0.2 0.73 0.1 0.1 0.02 0.37 1 0.1 0.1 0

1 1

0.85422 1

#6.1

0.14 0.23 0.83 0.1 0.1 0.02 0.73 1 0.1 0.1 1

1 0

1 0.00001

#7.1

0.14 0.23 0.83 0.1 0.1 0.02 0.73 1 0.1 0.1 0

1 1

0.97586 1

#8.1

0.07 0.11 0.52 0.1 0.1 0.3 0.87 1 0.1 0.1 1

1 0

0.99979 0.00006

#9.1

0.05 0.11 1 0.1 0.1 0.02 0.44 0.59 0.1 0.1 1

1 0

1 0.15468

#10.1

0.2 0.2 1 0.1 1 0.07 0.07 0.73 0.1 1 1

1 1

0.9997 0.99992

#11.1

0.18 0.18 1 0.1 1 0.29 0.29 0.65 0.1 1 1

1 0

0.99973 0.296

#12.1

0.14 0.14 0.61 0.1 0.1 0.04 0.84 1 1 1 1

1 0

0.99999 0.00007

#13.1

0.06 0.15 1 0.1 0.1 0.15 0.23 0.42 1 1 1

1 1

0.9996 0.99999

#14.1
0.06 0.15 1 0.1 0.1 0.15 0.23 0.42 1 1 0
1 1
0.98681 1

#15.1
0.1 0.1 1 0.1 1 0.1 0.67 0.75 1 1 1
1 0
0.99946 0.00092

#16.1
0.1 0.1 1 0.1 1 0.1 0.67 0.75 1 1 0
1 1
0.59812 1

#17.1
0.12 0.15 1 0.1 0.1 0.15 0.5 0.5 0.1 1
1
1 0
0.99936 0.46884

#18.1
0.12 0.15 1 0.1 0.1 0.15 0.5 0.5 0.1 1 0
1 1
0.03675 1

#19.1
0.16 0.16 1 0.1 1 0.05 0.25 0.84 0.1 0.1 1
1 0
1 0.02603

#20.1
0.16 0.16 1 0.1 1 0.05 0.25 0.84 0.1 0.1 0
1 1
0.9909 1

#21.1
0.18 0.15 1 0.1 0.1 0.15 0.33 0.7 0.1 0.1 1
1 0
1 0.04491

#22.1
0.13 0.13 0.7 0.1 0.1 0.03 0.37 1 0.1 0.1 1
1 0
1 0.0226

#23.1
0.13 0.13 0.7 0.1 0.1 0.03 0.37 1 0.1 0.1 0
1 1
0.88819 1

#24.1
0.06 0.12 0.43 0.1 0.1 0.06 0.83 1 0.1 0.1 1
1 0
1 0

#25.1
0.16 0.32 1 0.1 0.1 0.06 0.15 0.18 0.1 0.1 1
1 1
0.99986 0.67858

#26.1
0.16 0.32 1 0.1 0.1 0.06 0.15 0.18 0.1 0.1 0
1 1
0.9863 1

#27.1
0.44 1 1 0.1 1 0.23 0.4 0.4 0.1 1 1
1 1
0.00028 0.9992

#28.1
0.44 1 1 0.1 1 0.23 0.4 0.4 0.1 1 0
1 1
0.00031 1

#29.1
0.08 0.11 1 0.1 0.1 0.12 0.15 0.3 0.1 0.1 1
1 0
0.99999 0.99991

#30.1
0.08 0.11 1 0.1 0.1 0.12 0.15 0.3 0.1 0.1 0
1 1
0.98725 1

#31.1

0.23 0.23 1 0.1 1 0.35 0.35 0.35 1 1 1

1 0

0.99962 0.00439

#32.1

0.23 0.23 1 0.1 1 0.35 0.35 0.35 1 1 0

1 1

0.00016 1

#33.1

0.15 0.3 1 0.1 0.1 0.11 0.11 0.13 0.1 1 1

1 0

0.99953 0.99992

#34.1

0.15 0.3 1 0.1 0.1 0.11 0.11 0.13 0.1 1 0

1 1

0.01443 1

#35.1

0.39 0.86 1 0.1 0.1 0.05 0.16 0.98 0.1 0.1 1

1 1

0.00018 0.00034

#36.1

0.39 0.86 1 0.1 0.1 0.05 0.16 0.98 0.1 0.1 0

1 1

0.06282 0.99814