

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**RECONSTRUÇÃO 3D DE OBJETOS SIMÉTRICOS A PARTIR DE
NUVENS DE PONTOS UTILIZANDO MODELOS PARAMÉTRICOS DE
SUPERQUÁDRICAS PARA RECONHECIMENTO POR VISÃO
ROBÓTICA**

DIANA CAROLINA MONTILLA PERAFÁN

**ORIENTADOR: JOSÉ MAURÍCIO SANTOS TORRES DA MOTTA
DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS**

**PUBLICAÇÃO:
BRASÍLIA/DF: AGOSTO – 2013**

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**RECONSTRUÇÃO 3D DE OBJETOS SIMÉTRICOS A PARTIR DE NUVENS DE
PONTOS UTILIZANDO MODELOS PARAMÉTRICOS DE SUPERQUÁDRICAS
PARA RECONHECIMENTO POR VISÃO ROBÓTICA**

DIANA CAROLINA MONTILLA PERAFÁN

**DISSERTAÇÃO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA
MECÂNICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE
BRASÍLIA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO
DO GRAU DE MESTRE EM SISTEMAS MECATRÔNICOS.**

APROVADA POR:

**Prof. Dr. José Maurício Santos Torres da Motta, (ENM/FT/UnB)
(Orientador)**

**Prof. Dr. Cristiano Jacques Miosso Rodrigues Mendes, (FGA/UnB)
(Membro Externo)**

**Prof. Dr. Díbio Leandro Borges, (CIC/IE/UnB)
(Membro Interno)**

BRASÍLIA/DF, 22 AGOSTO 2013

FICHA CATALOGRÁFICA

PERAFÁN, DIANA CAROLINA MONTILLA

RECONSTRUÇÃO 3D DE OBJETOS SIMÉTRICOS A PARTIR DE NUVENS DE PONTOS UTILIZANDO MODELOS PARAMÉTRICOS DE SUPERQUÁDRICAS PARA RECONHECIMENTO POR VISÃO ROBÓTICA

xviii, 89p., 210 x 297mm (ENM/FT/UnB, Mestre, Sistemas Mecatrônicos, 2013).

Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.
Departamento de Engenharia Mecânica

- | | |
|----------------------------|-----------------------|
| 1. Superquadricas | 2. Levenber-Marquardt |
| 3. Superelipsóides | 4. Nuvem de Pontos |
| 5. Sistema <i>Kinect</i> ® | |

REFERÊNCIA BIBLIOGRÁFICA

PERAFÁN, D. C. M. (2013). RECONSTRUÇÃO 3D DE OBJETOS SIMÉTRICOS A PARTIR DE NUVENS DE PONTO UTILIZANDO MODELOS PARAMÉTRICOS DE SUPERQUÁDRICAS PARA RECONHECIMENTOPOR VISÃO ROBÓTICA. Dissertação de Mestrado em Sistemas Mecatrônicos, Publicação, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 79p.

CESSÃO DE DIREITOS

AUTOR: Diana Carolina Montilla Perafán

RECONSTRUÇÃO 3D DE OBJETOS SIMÉTRICOS A PARTIR DE NUVENS DE PONTOS UTILIZANDO MODELOS PARAMÉTRICOS DE SUPERQUÁDRICAS PARA RECONHECIMENTO POR VISÃO ROBÓTICA:

GRAU: Mestre

ANO: 2013

É concedida à Universidade de Brasília permissão para reproduzir copias desta Dissertação de Mestrado e para emprestar ou vender tais copias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

Diana Carolina Montilla Perafán
SCLN 407 Bloco A, Sala 207.
34.318.618 – Brasília – DF- Brasil.

AGRADECIMENTOS

Agradeço, primeiramente, ao meu orientador, Prof. Dr. José Maurício Santos Torres da Motta, que conseguiu entender todas as minhas dificuldades.

Agradeço a meus pais pela confiança e pelo apoio incondicional, agradeço a meu marido e à minha filha, pois, nos momentos em que tudo que eu precisei foi de uma manifestação de carinho e compreensão, eles me fortaleceram e me encorajaram até eu chegar à conclusão deste trabalho.

RESUMO

RECONSTRUÇÃO 3D DE OBJETOS SIMÉTRICOS A PARTIR DE NUVENS DE PONTOS UTILIZANDO MODELOS PARAMÉTRICOS DE SUPERQUÁDRICAS PARA RECONHECIMENTO POR VISÃO ROBÓTICA

Uma das aplicações de sistemas de visão robótica em intensa pesquisa atualmente é a reconstrução 3D de objetos a partir de nuvens de pontos para realizar o reconhecimento e localização desses objetos. Este trabalho tem como objetivo desenvolver um método para reconstruir objetos sólidos representados por nuvem de pontos utilizando um modelo paramétrico de superquádricas com o fim de usar em sistemas de visão robótica.

A metodologia implementada consiste em obter 5 superelipsoides geradas artificialmente mais 4 objetos reais com nuvem de pontos extraída usando uma câmera comercial com sensor de profundidade - o *Kinect*®. A partir da nuvem de pontos de objetos simétricos que possam ter descrição matemática por superquádricas foram utilizados algoritmos especialmente desenvolvidos para a identificação dos onze parâmetros que descrevem tamanho, posição, rotação e forma que caracterizem o objeto-alvo. Por último, a tarefa de minimizar a função implícita da superquádrica com o algoritmo iterativo de Levenberg-Marquardt até chegar aos verdadeiros parâmetros e construir o modelo tridimensional do objeto-alvo.

A análise dos resultados se fez de duas formas. Primeiramente, foi feita a avaliação de cada etapa do algoritmo através da comparação dos resultados obtidos com os resultados esperados. Neste caso, os resultados demonstraram um erro padrão assintótico de 1% para cada parâmetro, o que favorece a validação da metodologia. Em seguida, foram usadas nuvens de pontos de objetos reais, cuja análise experimental demonstrou resultados do vetor de parâmetros característicos do objeto com valor de erro padrão assintótico para cada parâmetro também menor do que 1%. Os resultados também puderam ser avaliados graficamente, mostrando a aproximação dos pontos estimados com os pontos das imagens reais durante as iterações.

Os resultados desta pesquisa demonstraram não só a eficácia do método com nuvens de pontos simuladas, mas também sua adequação à nuvem de pontos de objetos reais, onde se incluem ruído, erros de calibração do sistema de aquisição e descrição do objeto.

ABSTRACT

3D RECONSTRUCTION OF SYMMETRIC OBJECTS FROM POINT CLOUDS USING PARAMETRIC MODELS OF SUPERQUADRICS FOR ROBOTIC VISION

One of the applications of robotic vision systems intensively researched today is the 3D reconstruction of objects from point clouds to achieve the recognition and location of these objects. This work aims at developing a method to reconstruct solid objects represented by point clouds using a parametric model of superquadrics in order to use in robot vision systems.

The implemented methodology makes use of 5 synthetic superellipsoids and 4 real objects with point clouds extracted by using a commercial camera with a depth sensor – The Microsoft Kinect®. With point clouds of symmetrical objects that may have mathematical description for superquadrics, algorithms were developed to identify the eleven parameters that describe size, position, rotation and shape that characterize the target object. Finally, the task of minimizing the implicit function of the superquadric by using the iterative algorithm of Levenberg-Marquardt to get the true parameters and to construct the 3D model of the target object.

The analysis was carried out in two ways. First, each step of the algorithm was assessed by comparing the results with the expected results. In this case, the results showed an average error of 1% for each parameter, which promotes the validation methodology. Then, point clouds were used from real objects, whose experimental analysis demonstrated results for the characteristic parameter vector with mean square errors for each parameter also less than 1%. The results also could be evaluated graphically, showing the approach of the estimated points to the points of the actual images during the iterations.

These results demonstrate not only the effectiveness of the method with simulated point clouds, but also its suitability for point clouds of real objects, which include noise, calibration errors of the acquisition system and object description.

LISTA DE FIGURAS

Figura 1. Reconstrução de objetos por entalhes. Adaptado de Alan H., 1998.	3
Figura 2. Processo de representação do ser humano com superelipsóides a) Capturar cena. b) Segmentação da cena e c) Modelagem por superelipsóides. Adaptado de Strand e Dillmann (2009).	4
Figura 3. Modelagem simples de Geometria estéreo. Adaptado de Trucco et. al (1998)	8
Figura 4. Esquema físico para um sensor de profundidade.....	9
Figura 5. Dispositivo <i>Kinect</i> ®. Adaptado de Teixeira et. al. (2011).....	10
Figura 6. Curva planar da superfície, no ponto (u,v)	13
Figura 7. Mapa de garrafa: a) Imagem de profundidade e b) localização de descontinuidade de profundidade e orientação.	14
Figura 8. Segmentação de objetos com superfícies lisas. a) Imagem de profundidade, b) Mapa H-K, c) Mapa do método <i>step</i> , d) mapa do método <i>roof</i>). Segmentação final. Yokoya e Levine (1989)	15
Figura 9. Ordem hierárquica dos modelos geométricos. Adaptado de Biegelbauer, 2006.	17
Figura 10. Família de superquádricas. (a). Elipsóides, (b). Hiperbolóides de dupla folha, (c). Hiperbolóides de uma folha e (d). Toró ide. Adaptado de Barr, 1981.	18
Figura 11. Processo de ajuste com a função e energia entre o objeto e o modelo de recuperação. Adaptado de Solina e Bajcsy, 1990.....	20
Figura 12. Método iterativo para reconstrução 3D. Adaptado de Leonardis <i>et al.</i> , 1997.	22
Figura 13. Reconhecimento de um objeto a) objeto, b) descrição das partes do objeto por superquádricas e c) Nível estrutural das partes do objeto: Juntas. d), e) e f), sequência do reconhecimento. Adaptado de Krivic e Solina, 2001.	23
Figura 14. Detecção de uma tigela. (a)Imagem real, (b) Imagem do objeto rotulado. Adaptado de Biegelbauer e Vincze, 2007.	24
Figura 15. Produto esférico de duas curvas. Adaptado de Jaklic A., et. al. 2000.	25
Figura 16: Exemplo de família de superelipsóides. Adaptada de Barr, 1981.	26
Figura 17. Família de superquádricas. a) Superelipsóides, b) superhiperbolóides de uma folha, c) superhiperbolóides de duas folhas e d) supertoróides. Adaptado de Jaclik A., et. al., 2000.	29
Figura 18. Etapas de desenvolvimento	36

Figura 19. Localização do objeto e o Sistema Kinect para capturar imagens de profundidade	39
Figura 20. Translação do sistema de coordenadas local ao sistema de coordenadas global de um objeto no espaço	41
Figura 21. Objetos em 3D.....	47
Figura 22. Objetos geométricos (a) cilindro, (b) octaedro (c) esfera (d) cubo.....	48
Figura 23. Nuvens de pontos para cada objeto da figura 21.....	51
Figura 24. Representação dos eixos de inércia, posição e estimativa do tamanho das nuvens.	53
Figura 25. Nuvens de pontos para as Superelipsóides1 e 3. a) Nuvens de pontos de entrada e b) Nuvens de pontos no sistema de coordenadas geral feito pelo algoritmo 3.....	57
Figura 26. Resposta da função F com respeito a o numero de pontos (N). a) Superelipsóide1 b) Superelipsóide 3.	58
Figura 27. Resultados da superelipsóide 1 a). Gráfico de uma iteração do método iterativo Levenberg-Marquardt, comparando a aproximação da Função F estimada (Azul) com a real (verde). Gráfico de $F(x,y,z)$ Vs o valor da coordenada X b) Gráfico em 3D dos pontos x, y, z reais com a cor azul e os pontos estimados com a cor verde.....	60
Figura 28. Resultados da superelipsóide 2 a). Gráfico de uma iteração do método iterativo Levenberg-Marquardt, comparando a aproximação da Função F estimada (Azul) com a real (verde). Gráfico de $F(x,y,z)$ Vs o valor da coordenada X b) Gráfico em 3D dos pontos x, y, z reais com a cor azul e os pontos estimados com a cor verde.....	62
Figura 29. Resultados da superelipsóide 3. a). Gráfico de uma iteração do método iterativo Levenberg-Marquardt, comparando a aproximação da Função F estimada (Azul) com a real (verde). Gráfico de $F(x,y,z)$ Vs o valor da coordenada X b) Gráfico em 3D dos pontos x, y, z reais com a cor azul e os pontos estimados com a cor verde.....	64
Figura 30. Resultados da superelipsóide 4 a). Gráfico de uma iteração do método iterativo Levenberg-Marquardt, comparando a aproximação da Função F estimada (Azul) com a real (verde). Gráfico de $F(x,y,z)$ Vs o valor da coordenada X b) Gráfico em 3D dos pontos x, y, z reais com a cor azul e os pontos estimados com a cor verde.....	66
Figura 31. Resultados da superelipsóide 5 a). Gráfico de uma iteração do método iterativo Levenberg-Marquardt, comparando a aproximação da Função F estimada (Azul) com a real (verde). Gráfico de $F(x,y,z)$ Vs o valor da coordenada X b) Gráfico em 3D dos pontos x, y, z reais com a cor azul e os pontos estimados com a cor verde.....	68

Figura 32. Nuvens de pontos dos objetos . (a) cilindro, (b) octaedro, (c) esfera e (d) cubo. ...	69
Figura 33. Resultados da superelipsóide em forma de cilindro (a) Coordenadas dos pontos ajustados (verdes) sobrepostos à nuvem de pontos (azuis) e (c), reconstrução da nuvem de pontos com o vetor de parâmetros ajustado β	71
Figura 34. Resultados da superelipsóide em forma de octaedro (a) Coordenadas dos pontos ajustados (verdes) sobrepostos à nuvem de pontos (azuis) e (c), reconstrução da nuvem de pontos com o vetor de parâmetros ajustado β	73
Figura 35. Resultados da superelipsóide em forma de esfera (a) Coordenadas dos pontos ajustados (verdes) sobrepostos à nuvem de pontos (azuis) e (c), reconstrução da nuvem de pontos com o vetor de parâmetros ajustado β	75
Figura 36. Resultados da elipsóide 1 em forma de cubo(a) Coordenadas dos pontos ajustados (verdes) sobrepostos à nuvem de pontos (azuis) e (c), reconstrução da nuvem de pontos com o vetor de parâmetros ajustado β	77

LISTA DE TABELAS

Tabela 1. Equações para cada tipo de Superquádricas	28
Tabela 2. Parâmetros característicos para gerar a nuvem de pontos do objeto alvo	48
Tabela 3. Valores reais e estimados dos ângulos de Euler.	54
Tabela 4. Dados do vetor β com o algoritmo 2 para cada nuvem de pontos.....	54
Tabela 5. Variação dos parâmetros entre os valores reais e estimados	56
Tabela 6. Resultados e erro padrão assintótico para a superelipsóide 1	59
Tabela 7. Resultados e erro padrão assintótico para a superelipsóide 2.....	61
Tabela 8. . Resultados e erro padrão assintótico para a superelipsóide 3.....	63
Tabela 9. Resultados e erro padrão assintótico para a superelipsóide 4.....	65
Tabela 10. Resultados e erro padrão assintótico para a superelipsóide 5.....	67
Tabela 11. Resultados para a superelipsóide 1	70
Tabela 12. Resultado para superelipsóide 2	72
Tabela 13. Resultado para superelipsóide 3	74
Tabela 14. Resultado para superelipsóide 4	76

SUMÁRIO

1.	INTRODUÇÃO.....	1
1.1	MOTIVAÇÃO.....	2
1.2	OBJETIVOS.....	4
1.2.1	Objetivo geral.....	4
1.2.2	Objetivos específicos.....	4
1.3	JUSTIFICATIVA.....	5
1.4	ORGANIZAÇÃO.....	6
2	REVISÃO BIBLIOGRÁFICA.....	7
2.1	SISTEMAS DE AQUISIÇÃO DE OBJETOS EM 3D.....	7
2.1.1	Sensor de imagens de profundidade.....	7
2.1.2	Sistema de aquisição de imagens de profundidade: <i>Kinect</i> ®.....	10
2.1.3	Segmentação da cena tridimensional.....	12
2.2	REPRESENTAÇÃO DE OBJETOS POR SUPERQUÁDRICAS.....	16
2.2.1	Conceitos das superelipsóides.....	24
2.2.2	Propriedades geométricas das superelipsóides.....	29
2.2.3	Função objetivo.....	31
2.3	ALGORITMO LEVENBERG-MARQUARDT.....	33
2.3.1	Modelo de minimização.....	33
2.3.2	Critérios de convergência.....	35
3	MÉTODO DE RECONSTRUÇÃO TRIDIMENSIONAL DE OBJETOS.....	36
3.1	GERAÇÃO DA NUVEM DE PONTOS.....	37
3.2	ESTIMATIVA DOS PARÂMETROS INICIAIS DA NUVEM DE PONTOS.....	39
3.2.1	Posição global da nuvem.....	40
3.2.2	Orientação da nuvem.....	41
3.2.3	Tamanho da nuvem.....	43

3.2.4	Declaração das condições iniciais dos parâmetros da função objetivo	44
3.3	ESTIMATIVA DA FUNÇÃO OBJETIVO.....	44
3.4	ALGORITMO DE MINIMIZAÇÃO: MÉTODO DE LEVENBERG-MARQUARDT	46
4	VALIDAÇÃO E TESTES DO MÉTODO DE RECONSTRUÇÃO 3D	50
4.1	Validação do método de reconstrução 3D com os objetos sintéticos.....	50
4.1.1	Parte 1: Estimativa dos parâmetros iniciais da nuvem de pontos.....	52
4.1.2	Parte 2. Definição da função objetivo	57
4.1.3	Parte 3: O Algoritmo de Levenberg-Marquardt	58
4.2	RECONSTRUÇÃO 3D COM OS OBJETOS SINTÉTICOS	69
4.2.1	Superelipsóide 1: Cilindro.....	69
4.2.2	Superelipsóide 2: Octaedro.....	72
4.2.3	Superelipsóide 3: Esfera	74
4.2.4	Superelipsóide 4: Cubo.....	76
4.2.5	Análise dos resultados	78
5	CONCLUSÕES	79
5.1	Trabalhos Futuros	80
6	REFERÊNCIAS BIBLIOGRÁFICAS	81

1. INTRODUÇÃO

O problema discutido neste trabalho refere-se à reconstrução tridimensional de objetos, mediante o modelo matemático de superelipsóides. O desenvolvimento se faz sucessivamente pelas etapas de aquisição da imagem de profundidade do objeto, escolha do modelo matemático, aplicação de um algoritmo de ajuste e, finalmente, a execução do processo para gerar a representação 3D do objeto.

As imagens de profundidade dos objetos podem ser obtidas por meio de diversas técnicas de visão robótica e conter diferentes tipos de informação, no caso informações tridimensionais na forma de uma nuvem de pontos. Neste trabalho, serão utilizadas coordenadas geométricas de objetos feitos por *software*, o que significa que são conhecidos os valores verdadeiros dos parâmetros característicos, como sua forma, tamanho e posição. Também serão obtidas imagens de profundidade de objetos reais (totalmente desconhecidos), as imagens serão capturadas a partir de um sistema de visão configurado para obter a maior quantidade de pontos que descreve ao objeto da cena.

O sistema de visão é composto principalmente pelo sensor do *Kinect*¹ que pode capturar as informações da cena e formar a nuvem de pontos dos objetos da cena. Este dispositivo tem entre seus componentes um sensor de profundidade que faz uma varredura do ambiente à sua volta em três dimensões, que também pode ser conectado ao computador e ter acessibilidade a inúmeras aplicações de softwares divulgadas em diversas comunidades e em *web sites* especializados. Conseqüentemente, as informações da cena com este sistema de aquisição são obtidas na forma de coordenadas espaciais chamadas nuvem de pontos.

Uma vez obtida a nuvem de pontos da cena, a tarefa é representar os objetos da imagem utilizando-se modelos matemáticos não paramétricos (como triangulação entre pontos) ou paramétricos (como superquádricas). O último tipo tem mais vantagens que os outros, entre elas a facilidade de representação, de indexação e também a compactação dos dados. Além disso, o modelo paramétrico sobre superfícies quádricas e sólidas produz uma variedade de formas comuns no mundo real. O modelo matemático unificado chamado de superquádricas

¹O sistema *Kinect@Microsoft* é um sensor de movimentos desenvolvido para jogos eletrônicos.

consiste em definir a superfície do objeto mediante uma equação paramétrica, segundo sua simetria.

Com a descrição matemática da geometria do objeto por intermédio de um vetor de parâmetros, é possível encontrar um conjunto de parâmetros que se ajustam por meio de um algoritmo iterativo de minimização. Dentre eles, existe o algoritmo de Levenberg-Marquardt, encarregado de buscar o mínimo global da função do objeto. Sua resposta depende da definição precisa das variáveis de entrada e os critérios de convergência para ajustar e representar o objeto em 3D.

1.1 MOTIVAÇÃO

O procedimento para reconstruir um objeto, basicamente, é modelagem da forma e das propriedades geométricas do objeto a partir de imagens, baseados em conceitos matemáticos e de engenharia capazes de descrever, analisar e interpretar o contexto do ambiente físico. Os avanços necessários para conseguir uma representação mais eficaz do objeto exigem a integração com algoritmos que se executam em tempo real, métodos robustos e uma reconstrução 3D densa, gerando questões de pesquisa em aberto que motivam o desenvolvimento de novas técnicas e a integração destas com outras já existentes.

Dentro das técnicas de reconstrução pode-se estudar o método de representação do objeto a partir de nuvem de pontos e, para desenvolver este método, existem diversos algoritmos propostos, como por exemplo, o proposto por Rusenkiewicz et. al. (2000), onde a nuvem de pontos de objetos é reconstruída diretamente do modelo poliédrico, mas a desvantagem principal é que, na maioria das aplicações, os objetos convertem-se em sólidos limitados pelas superfícies de polígonos ou em malhas triangulares irregulares que compartilham suas bordas e seus vértices, tornando a representação muito trabalhosa e os erros ainda mais evidentes. Outro modo de reconstrução do objeto é mediante a representação por subdivisão paramétrica, quer dizer, uma superfície paramétrica pode ser representada através de pedaços de superfícies denominados entalhes (*patches*). Um exemplo dessa técnica é observado na Figura 1. No entanto, verificou-se que, para que a nuvem de pontos do objeto corresponda

exatamente à reconstrução, o formato do objeto real deve ser definido por muitos pontos, ainda mais quando o objeto tem curvaturas.

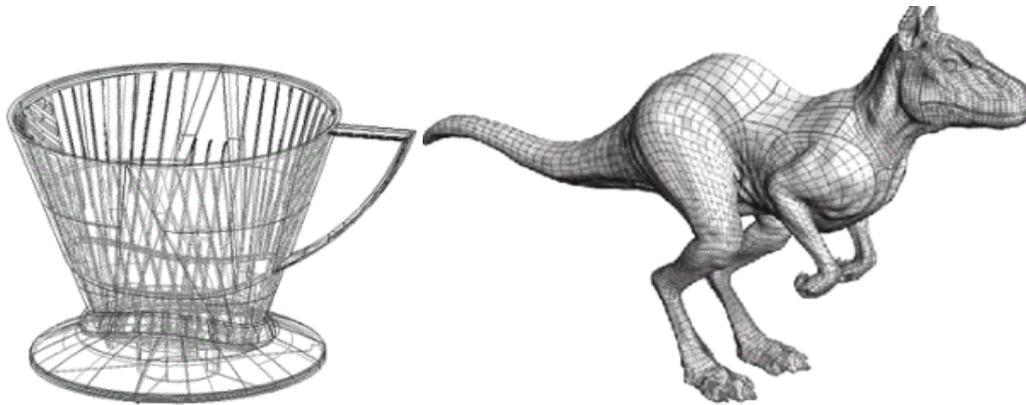


Figura 1. Reconstrução de objetos por entalhes. Adaptado de Alan H., 1998.

Um método alternativo, que contorne estes problemas, pode ser a representação geométrica de objetos por superfícies paramétricas chamadas de superquádricas. A representação do objeto é compacta e robusta, além da reduzida quantidade de parâmetros necessários para descrever o objeto e a possibilidade de controlar sua forma geométrica pelo método matemático unificado apresentado por Barr (1981).

No entanto, sua suposição de simetria intrínseca falha na modelagem de inúmeros exemplos do mundo real, tais como o corpo humano, animais e outros objetos naturais. Assim, Barr (1984) apresenta o método matemático com uma abordagem nova, que é chamada superquádricas estendidas, ampliando os expoentes da superquádricas e gerando formas assimétricas para a modelagem.

Uma aplicação desses métodos matemáticos é vista no trabalho de Strand e Dillmann (2009) (Figura 2), em que se descreve o ser humano com modelos compactos de representação volumétrica, de geometrias simples, ou seja, com uma família de superfícies paramétricas chamadas de superelipsóides.

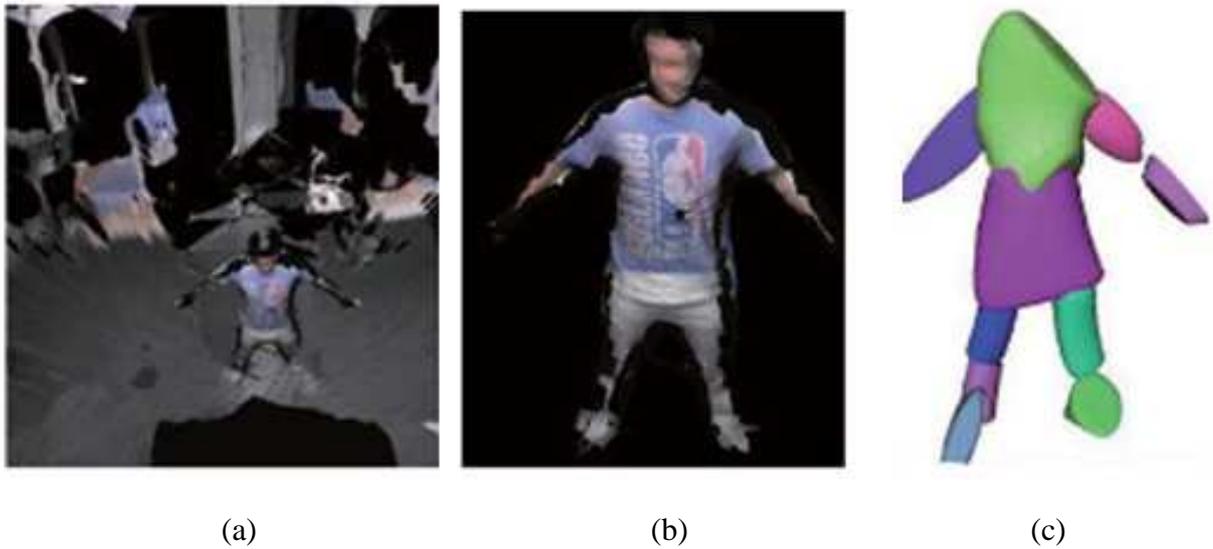


Figura 2. Processo de representação do ser humano com superelipsóides a) Capturar cena. b) Segmentação da cena e c) Modelagem por superelipsóides. Adaptado de Strand e Dillmann (2009).

Este trabalho propõe e programa um algoritmo para reconstrução de objetos simétricos e reais a partir de nuvens de pontos capturadas por um sistema de visão. O algoritmo é baseado no modelo matemático de superquádricas, e verificado com objetos construídos em software e com objetos reais com imagens de profundidade reais, utilizando o programa Matlab®. Esta técnica é aparentemente simples, mas seu fundamento tem fortes conceitos matemáticos que o fazem um método viável para a modelagem, reconstrução ou recuperação tridimensional de um objeto. Além disso, os resultados deste processo garantem e aumentam as alternativas de reconstrução de um objeto de uma cena real.

1.2 OBJETIVOS

1.2.1 Objetivo geral

O objetivo deste trabalho é implementar modelos matemáticos baseados em superquádricas para a reconstrução tridimensional de objetos geométricos usando, para isso, sua nuvem de pontos.

1.2.2 Objetivos específicos

- Implementar os algoritmos para encontrar o vetor de parâmetros que poderá descrever a nuvem de pontos mediante a equação da superquádrica.
- Adequar o algoritmo para determinar o valor da função de minimização. A função será ajustada iterativamente pelo algoritmo de minimização de Levenberg-Marquardt.
- Integrar as etapas anteriores e testar o desempenho do modelo com a família de superelipsóides usando dois tipos de dados de entrada:
 1. Descrição do objeto com nuvem de pontos ideais.
 2. Descrição do objeto com nuvem de pontos reais.
- Validar o modelo matemático por meio das análises de erro e pela apreciação visual dos resultados.

1.3 JUSTIFICATIVA

É claro que, para se fazer a reconstrução 3D deve-se ter um ferramental matemático e recursos computacionais que garantam a eficácia do processo, para depois aplicar a técnica em um sistema de visão robótica. Desta forma, é possível identificar necessidades contínuas impostas tanto pelas demandas da crescente atividade de pesquisa como também em decorrência dos avanços tecnológicos que possibilitam o desenvolvimento de projetos mais avançados.

No laboratório de robótica do Grupo de Automação e Controle - GRACO, da Universidade de Brasília- UnB, vem-se trabalhando a implantação de modelos matemáticos para reconstrução 3D. Sua incursão na modelagem tridimensional de uma cena começa com a construção de um sensor de profundidade, baseado em varredura por luz laser para um robô manipulador, o que fomenta estudos para aperfeiçoar a captura das imagens e testes de algoritmos mais velozes, identificando todas as variáveis que influenciam na precisão, velocidade e eficiência de um sistema de construção de modelos de superfícies.

Como consequência, o laboratório tem construído o suporte para a viabilização de novos modelos de reconstrução 3D que demandam recursos de visão robótica. A isto, soma-se o grau de confiabilidade que gera e a pesquisa científica da diversificação de métodos que tem o mesmo fim de modelar o objeto em 3D.

1.4 ORGANIZAÇÃO

Os capítulos deste trabalho são apresentados da seguinte maneira: no primeiro, a introdução, abrange o desenvolvimento geral, motivação, objetivos e justificativa do trabalho. No segundo capítulo é apresentada uma revisão teórica quanto à contextualização frente ao campo da visão robótica, a fundamentação teórica na descrição de um objeto por equações paramétricas mediante o modelo de superquádricas, a geometria das superquádricas para avaliar seu sistema de coordenadas e, por último, a apresentação do algoritmo de minimização Levenberg-Marquardt como método de ajuste de duas funções. O capítulo terceiro apresenta o modelo executado em diagramas de blocos e cada etapa do processo é descrita em subitens. Em seguida, no capítulo dos resultados, é avaliada cada uma das etapas em dois tipos de experimentos, ou seja, com os objetos construídos pelo *software* e com os objetos reais obtidos pelo sistema de visão. Finalmente, temos as conclusões e as referências bibliográficas.

2 REVISÃO BIBLIOGRÁFICA

2.1 SISTEMAS DE AQUISIÇÃO DE OBJETOS EM 3D

Em geral, o sistema de aquisição volumétrica por visão proporciona informação sobre os atributos da cena (modelagem do ambiente) ou dos objetos, (posição, orientação e suas inter-relações funcionais) e talvez, em determinadas aplicações, um subconjunto dessa informação seria suficiente para iniciar um processo de controle pelo robô. Os sistemas de aquisição volumétrica basicamente contém três etapas: sensoreamento, processamento de imagens e a reconstrução da cena ou o objeto. Cada uma dessas etapas é desenvolvida com diversas técnicas que, dependendo da precisão dos resultados podem ter diversas aplicações em diferentes áreas.

2.1.1 Sensor de imagens de profundidade.

Os sensores CCD² e CMOS³ de imagem são compostos por milhões de semicondutores que captam a intensidade de luz e a carga elétrica dos fótons que partem da cena. No entanto, o aumento da demanda da tecnologia de automação, que cada vez mais depende do conteúdo 3D da cena, fez evoluir os sensores para que adquiram tanto informação de brilho ou cor da cena como também a profundidade em que o raio de luz associado a cada pixel intercepta a cena observada por uma câmera.

Existem vários tipos de sensores de imagens usados para reconstrução 3D de objetos, os comumente usados são os sensores de intensidade e os sensores de profundidade. Um sensor de profundidade determina a distância $z(x, y)$ desde um ponto visível na superfície do objeto até o sensor e pode controlar sua própria iluminação. A distância pode ser medida por tempo de voo do iluminador ou por triangulação.

² CCD (*charge-coupled device*) sensor para captação de imagens

³ CMOS (*complementary metal-oxide-semiconductor*),

Os sistemas de luz estruturada são sensores ativos baseados em triangulação. Seu funcionamento parte dos mesmos princípios do sistema da Figura 3. Essa modelagem simples de detecção de um ponto da cena se faz com duas câmeras localizadas nos pontos A e B na mesma linha horizontal, mas uma delas capta o lado direito do ponto W e a outra o lado esquerdo do ponto W. C1 e C2 são os centros das lentes da câmera esquerda e direita, f é o comprimento focal, z é a medida de profundidade e x1, x2 correspondem às coordenadas detectadas do ponto W no lado esquerdo e direito. A profundidade z então é definida por semelhança dos triângulos WAB e WC1C2, assim:

$$z = \frac{Bf}{x_1 + x_2} \quad (1)$$

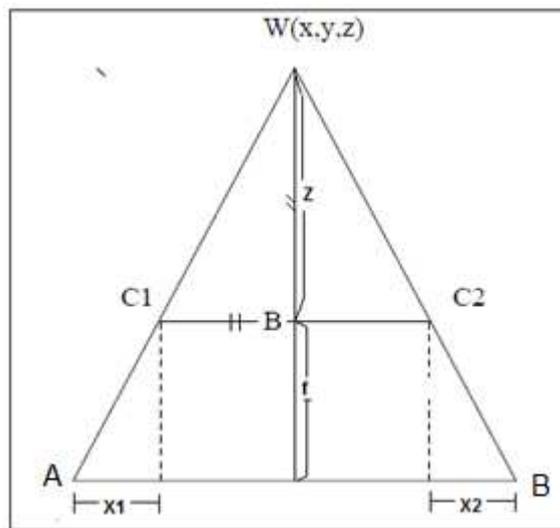


Figura 3. Modelagem simples de Geometria estéreo. Adaptado de Trucco et. al (1998)

A construção de um sistema de luz estruturada consiste em substituir uma das câmeras por uma fonte de iluminação controlada (luz estruturada), seja uma luz na forma de ponto, malha ou grade. A luz é capturada pela outra câmera e diminui o problema de correspondência do método estéreo. Na Figura 4, um projetor é configurado para emitir o feixe do laser em um ângulo α em relação à linha de referência e é composto por uma lente cilíndrica e um elemento refletor da luz (espelho), e o raio refletido pelo objeto da cena é detectado pelo sensor. A distância entre o projetor e a câmera é a linha de base B. A câmera possui comprimento focal f_c tem suas lentes localizadas na origem. Um ponto (x, y, z) no espaço real é capturado pelo sensor e a profundidade z_0 é dada por:

$$Z_o = \frac{B}{\frac{x_o}{f} + \tan \alpha} \quad (2)$$

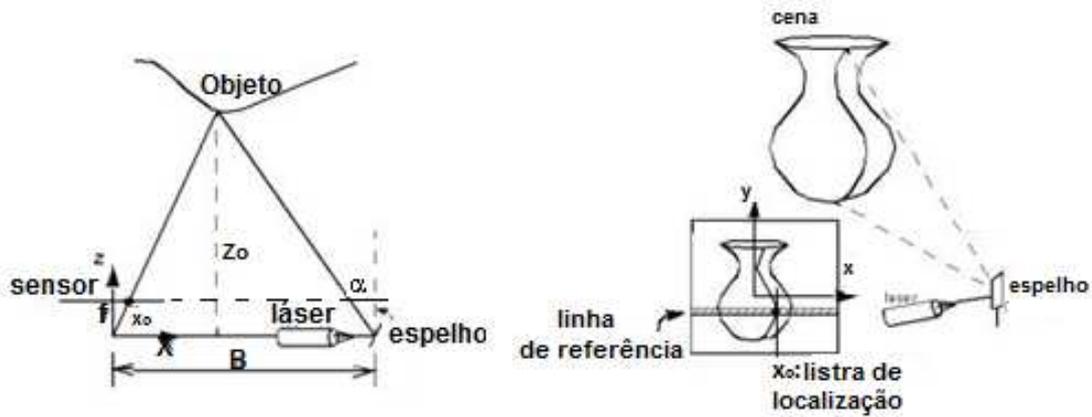


Figura 4. Esquema físico para um sensor de profundidade.

Sendo o espelho rotativo, então o ângulo α varia com o tempo (sendo $\alpha = \omega t$, ω *velocidade angular* e t o tempo) e a superfície será digitalizada sequencialmente para fornecer o mapa de profundidade da cena. Assim, o tempo t é determinado pela distância em que o laser percorre uma distância entre a coordenada x_o da lista de localização e a coordenada x ao longo da linha de referência (Figura 4). O ângulo α é então determinado através da medição do tempo decorrido a partir de um ponto de referência. Deste modo a equação (2) é definida novamente como:

$$Z = \frac{B}{\frac{x}{f} + \tan \alpha} \quad (3)$$

Os valores f , B , $\tan \alpha$ sendo $\alpha = \omega t$ são conhecidos pela configuração do equipamento e a localização dos pixels é encontrada por detecção de pico percorrendo a linha de varredura.

Uma importante característica dessa configuração é o custo, no entanto o sistema não é muito preciso, já que sua resolução é limitada pela resolução do pixel.

2.1.2 Sistema de aquisição de imagens de profundidade: *Kinect*®

O *Kinect*® da Microsoft (2010) é um dispositivo (Figura 5) que possui uma câmera colorida com um sensor (RGB) e um sensor de profundidade infravermelho (IR), além de um processador personalizado que executa o software de propriedade da Microsoft. O sensor de profundidade é um projetor de infravermelho combinado com o sensor (CMOS) monocromático que permite obter uma imagem em 3D para qualquer condição de iluminação.



Figura 5. Dispositivo *Kinect*®. Adaptado de Teixeira et. al. (2011)

Um estudo detalhado do *Kinect*® foi desenvolvido no ano 2013 (Smisek et.al., 2013): o sensor de profundidade é composto por uma fonte de luz Infravermelha (IR - *InfraRed*) e um sensor CMOS monocromático que permite ao *Kinect*® obter a cena 3D em qualquer condição de luz ambiental. Então, o dispositivo *Kinect*® captura, em tempo real, a imagem de profundidade do ambiente através da luz infravermelha estruturada, fazendo o método de triangulação com a imagem capturada. O intervalo de detecção da profundidade do sensor é ajustável pelo *software*, que é capaz de calibrar automaticamente o sensor.

O sensor de profundidade e o sensor CMOS, dispostos horizontalmente a uma distância de aproximadamente 2,5cm entre si (Figura 5), registram duas imagens levemente díspares, por isso é necessário executar o processo de referenciar um pixel de cor exatamente com o pixel da imagem de profundidade respectivo. Este processo matemático é baseado no modelo de projeção geométrica dos sistemas de luz estruturada. Primeiro, a equação três é definida como

a função de profundidade $P(x, y)$, e as coordenadas reais dos pontos capturados pelo sensor de profundidade do *Kinect*® são dados por:

$$\begin{aligned} X &= (x_{im} - O_x) * P(x_{im}, y_{im}) / f_{xim} \\ Y &= (y_{im} - O_y) * P(x_{im}, y_{im}) / f_{yim} \\ Z &= P(x_{im}, y_{im}) \end{aligned} \quad (4)$$

Sendo (x_{im}, y_{im}) as coordenadas 2D da imagem em píxeis, (O_x, O_y) a origem do sistema, (f_{xim}, f_{yim}) o comprimento focal da câmera, $P(x_{im}, y_{im})$ a função de profundidade e (X, Y, Z) as coordenadas 3D do ponto projetado.

Uma vez obtidas às coordenadas (X, Y, Z) de cada ponto no sistema geral de coordenadas, é feita a transformação geométrica ao sistema de coordenadas da câmera RGB $(X_{RGB}, Y_{RGB}, Z_{RGB})$ por meio dos parâmetros extrínsecos, a saber: vetor de translação T e matriz de rotação R , de acordo com a equação:

$$\begin{bmatrix} X_{RGB} \\ Y_{RGB} \\ Z_{RGB} \end{bmatrix} = R * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T \quad (5)$$

Finalmente, esses pontos são novamente projetados geometricamente na imagem RGB para definir as coordenadas de corem píxeis (x_c, y_c) ligadas aos parâmetros intrínsecos da câmera RGB $(O_{xc}, O_{yc}, f_{xc}, f_{yc})$. Então, um ponto X, Y, Z no espaço real é projetado na câmera no pixel (x_c, y_c) de forma que:

$$\begin{aligned} x_c &= X_{RGB} * f_{xc} / Z_{RGB} + O_{xc} \\ y_c &= Y_{RGB} * f_{yc} / Z_{RGB} + O_{yc} \end{aligned} \quad (6)$$

Deste modo, tem-se a nuvem de pontos da cena e as coordenadas de cor, no entanto, o procedimento matemático já está disponível em algumas bibliotecas que fornecem suporte ao *Kinect*, como o *Kinect for Windows SDK beta* e o *OpenNI*.

2.1.3 Segmentação da cena tridimensional

A segmentação serve para destacar os objetos ou ambientes de interesse que serão estudados em tempo real. Uma imagem de profundidade representa uma superfície de uma cena e pode ser descrita por uma função implícita, explícita ou paramétrica, segundo sua geometria. A imagem pode conter várias superfícies representadas por diferentes funções, mas o processo de segmentação pode encontrar as descontinuidades entre as superfícies e as uniões entre objetos similares.

O método de segmentação para este tipo de imagens se faz sobre o princípio da geometria analítica diferencial encarregada da medição de distâncias, ângulos e outros parâmetros, além da detecção de bordas e extremidades.

2.1.3.1 Princípio de geometria analítica diferencial

A geometria diferencial de uma superfície pode fornecer informação sobre as propriedades intrínsecas da morfologia do espaço e suas medidas, porém estuda as estruturas geométricas. A partir de noções básicas de cálculo e álgebra linear faz-se um estudo de curvas e superfícies, mas não somente considerando o lugar geométrico que as curvas e superfícies ocupam no espaço, senão também a parametrização delas. Estes conceitos serão adaptados para identificar numa superfície partes que tenham características similares e descontinuidades para assim conseguir a segmentação e, conseqüentemente, a reconstrução 3D baseada em imagens.

Considerando o domínio da superfície paramétrica da Figura 6, como o local de análises de pequenas variações de sua superfície, encontra-se o vetor tangente e a superfície normal que definem o sistema de coordenadas ortogonais de um ponto P sobre a superfície. No momento em que se faz a interseção do plano normal com a superfície que contém o ponto P (Figura 6) representa-se a variável para descrever a forma local da superfície nesse ponto P , chamada de curva planar.

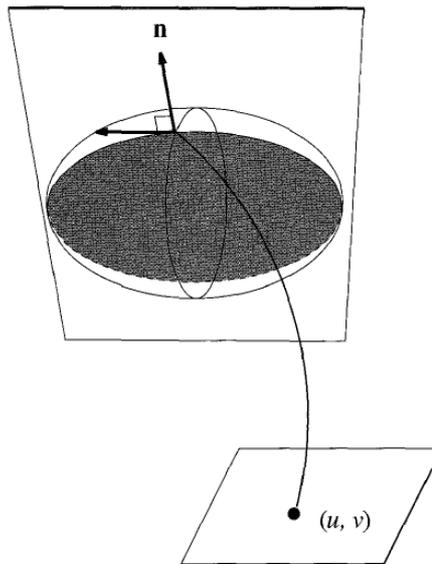


Figura 6. Curva planar da superfície, no ponto (u,v).

Para cada curva planar, tem-se os máximos e mínimos raios de curvatura k_1 e k_2 e a curvatura Gaussiana K e a curvatura média H são definidas como:

$$K = k_1 \cdot k_2 \quad (7)$$

$$H = \frac{k_1 + k_2}{2} \quad (8)$$

Os parâmetros K e H são os que descrevem uma imagem de profundidade em regiões de forma homogênea chamadas de “entalhes de superfícies”. Por exemplo, a segmentação de um objeto com várias formas pode ser feita distinguindo-se a mudança de suas formas pelas discontinuidades usando H e K para segmentar; para isso, apresenta-se o método (Ponce e Brady, 1987), que combina as ferramentas da geometria diferencial e assim localizar intersecções na superfície. A Figura 7 descreve uma garrafa com as diferentes instâncias de separação entre elas, dois tipos de discontinuidades: *step*, onde a profundidade atual é descontínua, e *roof*, onde a profundidade é contínua, mas a orientação muda abruptamente.

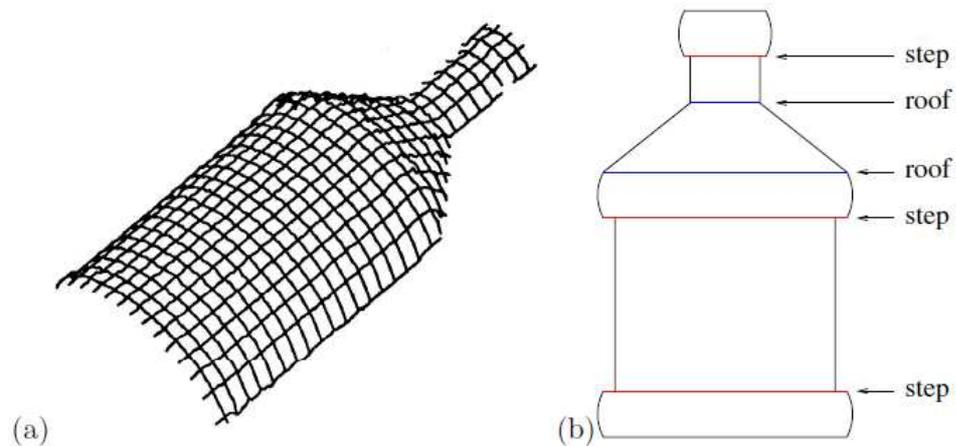


Figura 7. Mapa de garrafa: a) Imagem de profundidade e b) localização de descontinuidade de profundidade e orientação.

Basicamente, a detecção de borda apresentado por Ponce e Brady (1987) observado no mapa da garrafa (Figura 7b) consiste em limitar as descontinuidades dos modelos cilíndricos que estão alinhados com a direção de um mesmo plano. Como resultado, tem-se a detecção de bordas sem problemas de detecção de curvatura artificial.

Yokoya e Levine (1989) fizeram uma análise de dados na segmentação de uma imagem 3D para objetos que tem superfícies lisas diferenciáveis. (Figura 8a). Para isso, criaram uma cena de superfícies primitivas homogêneas e, em seguida, obtiveram o mapa de curvatura pelo cálculo das curvaturas (Figura 8b), dois mapas obtidos pelo método descrito anterior de *step* e *roof* (Figura 8c e 8d) e como resultado tem-se a segmentação final. (Figura 8e).

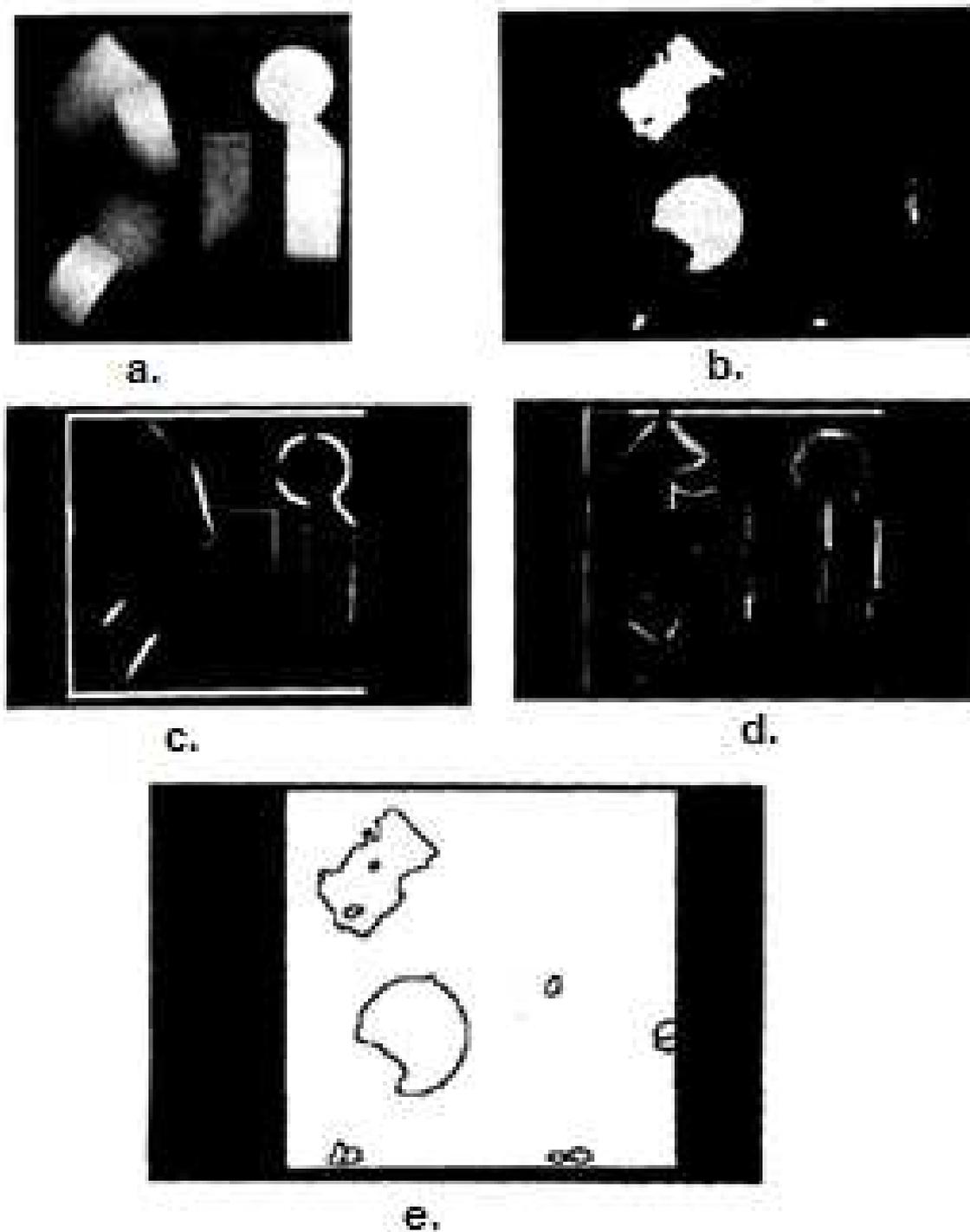


Figura 8. Segmentação de objetos com superfícies lisas. a) Imagem de profundidade, b) Mapa H-K, c) Mapa do método *step*, d) mapa do método *roof*. Segmentação final. Yokoya e Levine (1989)

No presente trabalho não se aplicou um algoritmo de segmentação das imagens de profundidade, mas a segmentação se fez manualmente. O *software* MeshLab® e 3DObject® foram utilizados para manipular a nuvem de pontos e assim, obter o arquivo 'txt' com as coordenadas espaciais.

Com os objetos de interesse podem-se extrair os seus parâmetros característicos e representar os objetos a partir de um modelo matemático. O objeto pode ser uma superfície que se supõe plana ou aberta, ou um sólido de superfície fechada que ocupa certo volume. O objetivo pode ser visualizar o aspecto externo do objeto ressaltando-se os detalhes de sua estrutura interna.

2.2 REPRESENTAÇÃO DE OBJETOS POR SUPERQUÁDRICAS

Existem diferentes modelos paramétricos em 3D usados para descrever possíveis objetos tendo como entrada nuvens de pontos que são usadas em visão computacional. Biegelbauer (2006) expõe a ordem hierárquica dos modelos geométricos (Figura 9) segundo o grau de complexidade e o número de parâmetros. Com menor grau de complexidade, estão as primitivas que se referem àqueles modelos com um único parâmetro de forma, como a esfera, cilindro, cone, toroide e caixa. Ascendendo, encontram-se os *Geons*, as Superquádricas e os Cilindros generalizados, que têm um limitado número de parâmetros. Por último, os modelos paramétricos de alta complexidade que representam um objeto arbitrário descritos por vários parâmetros, como as hiperquádricas, polinomiais implícitos e superfícies harmônicas esféricas.



Figura 9. Ordem hierárquica dos modelos geométricos. Adaptado de Biegelbauer, 2006.

Neste contexto, identifica-se a superquádrlica como uma família de formas paramétricas sobre superfícies quádrlicas e sólidas, produzindo-se uma variedade de formas úteis, baseadas num modelo matemático unificado (Barr, 1981). As superquádrlicas (Figura 10) foram divididas em superelipsóides (Figura 10a), superhiperbolóides de duas folhas (Figura 10b), superhiperbolóides de uma folha (Figura 10c) e supertoróides (Figura 10d). Cada um tem sua equação paramétrica e nela existem parâmetros de forma encarregados de gerar ainda objetos dentro de cada tipo de superquádrlicas. Por isso, na Figura 10, observa-se cada família de superquádrlicas descrita por uma equação paramétrica, mas com parâmetros de forma diferentes.

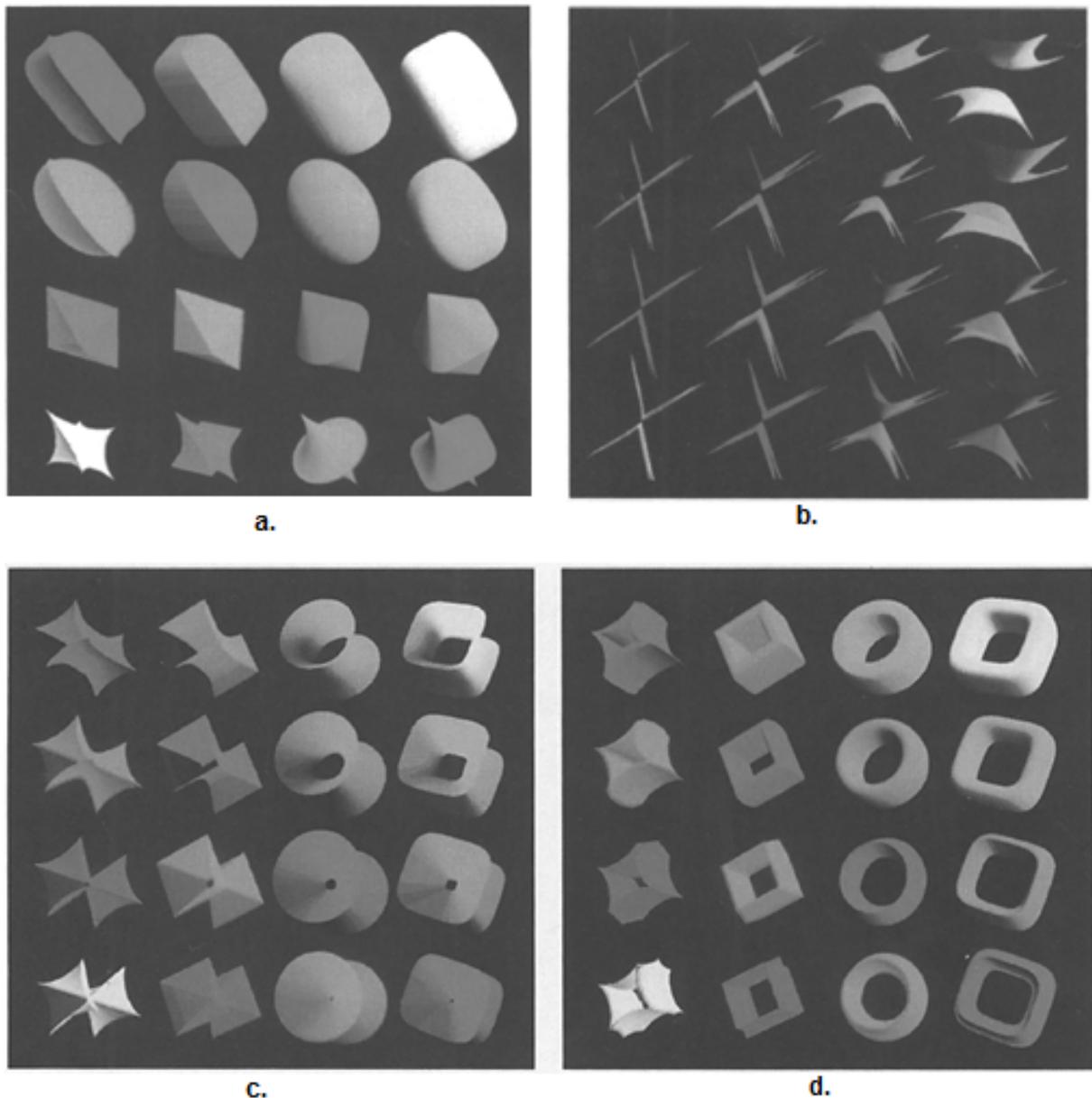


Figura 10. Família de superquádricas. (a). Elipsóides, (b). Hiperbolóides de dupla folha, (c). Hiperbolóides de uma folha e (d). Toró ide. Adaptado de Barr, 1981.

No entanto, o modelo matemático de Barr (1984) acrescentou os parâmetros para descrever os objetos geométricos deformáveis, que podem ser modelados por modelos de superquádricas quando se adiciona os parâmetros de deformações *tapering* e *bending*. Assim, a família de superquádricas foi aumentando, ao mesmo tempo em que se aumentaram também o estudo geométrico e matemático delas. (Barr, 1984)

O desenvolvimento do modelo matemático das superquádricas, feito por Barr (1981), surgiu e acabou por envolver a área de Visão Computacional com a reconstrução 3D de objetos. O

reflexo disso encontra-se nas aplicações e trabalhos como: o esboço tridimensional da estrutura de uma peça, descrito por Pentland (1986); a demonstração dos argumentos qualitativos e quantitativos sobre os critérios de análises experimentais da qualidade, apresentados por Gupta et al. (1989); o desenvolvimento de um método para ajustar superquádricas, por Solina e Bajcsy (1990); como também o reconhecimento de um conjunto de peças geométricas usando-se a técnica de segmentação de uma imagem de profundidade, realizadas por Leonardis et al. (1997); e a definição das propriedades geométricas, dadas no livro de Jaklic et al. (2000); entre outros.

O modelo matemático das superquádricas também foi estudado para objetos isolados (Pentland, 1991 e Solina e Bajcsy, 1990). O desenvolvimento deste trabalho foi a partir da construção de um objeto chamado “*objeto de recuperação*” com algumas características do objeto real. Depois se executa o processo de ajuste, com o método de minimização da soma dos quadrados, entre as funções definidas para os dois objetos. Por último, Pentland (1991) e Solina e Bajcsy (1990) conseguem encontrar as características do objeto real e reconstruir tridimensionalmente o objeto. Na Figura 11, observa-se o processo iterativo do ajuste de um objeto de recuperação com forma de elipsóide até a representação do objeto real. A primeira imagem é o objeto alvo, a imagem seguinte corresponde a sua nuvem de pontos, na imagem da letra E da figura é a representação entre o objeto e o modelo de recuperação e depois, observa-se as imagens para as iterações 1, 3, 7, 15 e 30. Sendo a iteração 30 o ajuste da nuvem de pontos do objeto com o modelo de recuperação.

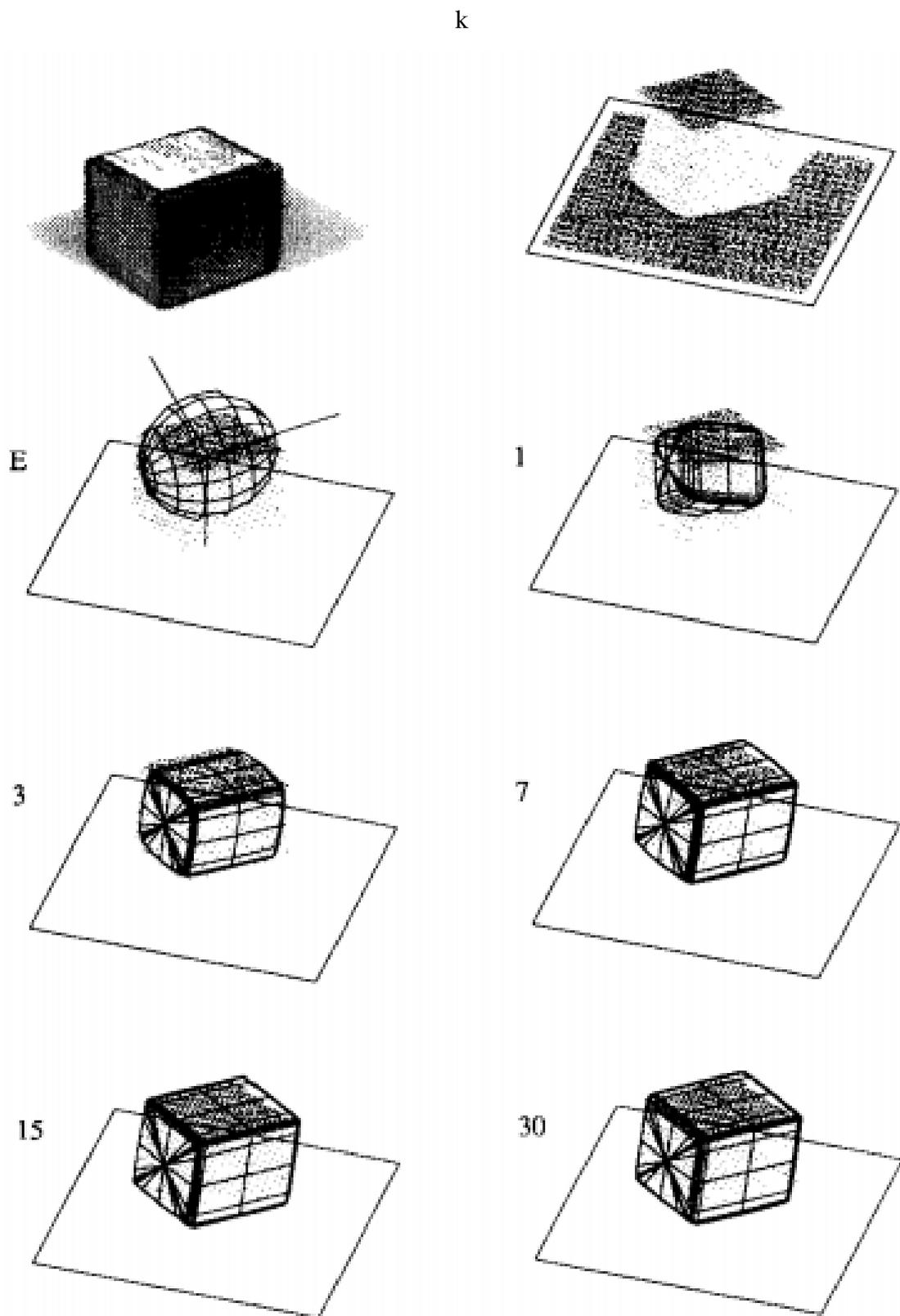


Figura 11. Processo de ajuste com a função e energia entre o objeto e o modelo de recuperação. Adaptado de Solina e Bajcsy, 1990.

Entretanto, a segmentação das imagens de profundidade foi avaliada com o fim de modelar objetos mais complexos. Os trabalhos de Pentland (1987), Pentland (1990), Ferrie *et al.* (1993) e Gupta e Bajcsy (1993) têm em comum a aplicação em modo hierárquico para estimar as propriedades de superfícies, as curvaturas, simetria rotacional, entre outras. Mas Leonardis *et al.* (1997) combinam os procedimentos de segmentação e recuperação de objetos e descrevem cada peça que forma um objeto complexo. Na Figura 12, observa-se o procedimento de Leonardis *et al.* (1997). A imagem de entrada é uma nuvem de pontos de vários objetos simétricos, em seguida o algoritmo escolhe pontos sementes para criar uma forma paramétrica padrão tal que o ponto semente vai crescendo até coincidir com os pontos do objeto real. O algoritmo tem em conta as descontinuidades da nuvem de pontos para identificar a nuvem de pontos de outro objeto e consegue chegar à descrição das formas de cada objeto presentes na cena.

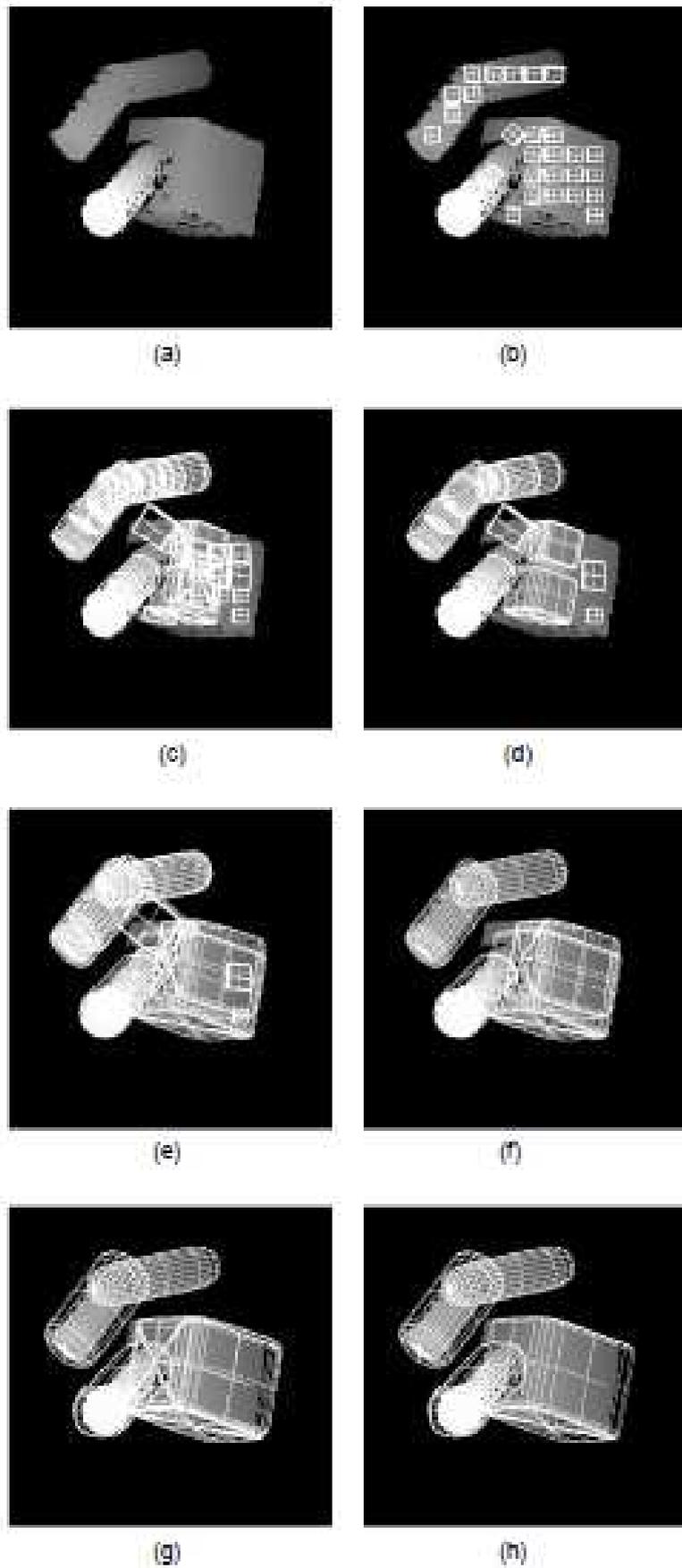


Figura 12. Método iterativo para reconstrução 3D. Adaptado de Leonardis *et al.*, 1997.

Outro tipo de proposta estudada, entre outras, é a de Krivic e Solina (2001), em que eles recuperam um objeto padrão com juntas articuladas dentro de um cenário complexo. Esse trabalho definiu primeiro um objeto padrão (Figura 13 a) composto de diferentes objetos geométricos, depois identificou as juntas de cada objeto e o representou com superquádricas (Figura 13b, 13c). A cena complexa é um conjunto de peças (Figura 13 d) que contem tanto as peças que pertençam e as que não pertencem ao objeto padrão. Então, foi criado o algoritmo combinando a técnica de segmentação (*Segmentor*), o modelo de correspondência (*Matching*) e a técnica de árvore de interpretação para reconhecer os objetos padrões na cena complexa (Figura 13e, 13f).

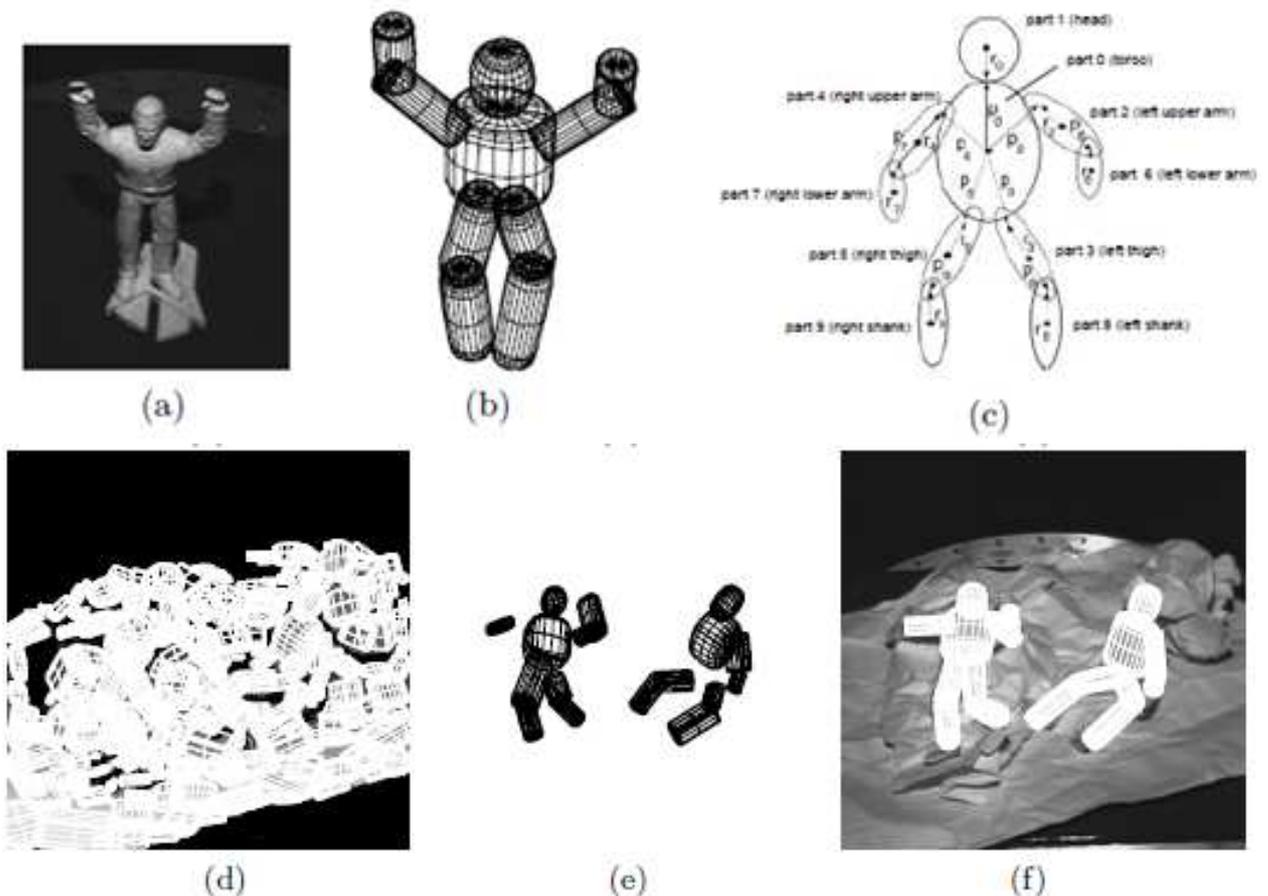


Figura 13. Reconhecimento de um objeto a) objeto, b) descrição das partes do objeto por superquádricas e c) Nível estrutural das partes do objeto: Juntas. d), e) e f), sequência do reconhecimento. Adaptado de Krivic e Solina, 2001.

Biegelbauer e Vincze (2007) apresentam o ajuste automático de diferentes objetos obtidos com um sensor de profundidade a laser numa única vista (Figura 14a). A modelagem de uma tigela é feita com superquádricas e a detecção é baseada no processo hierárquico RANSAC,

chegando a resultados rápidos e ótimos critérios de qualidade para encontrar uma tigela, que é rotulada de cor verde, quando é encontrada (Figura 14b).



Figura 14. Detecção de uma tigela. (a) Imagem real, (b) Imagem do objeto rotulado. Adaptado de Biegelbauer e Vincze, 2007.

Em geral, todos os estudos foram direcionados a objetos sintéticos e reais tendo em conta a 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. No entanto, a simulação de uma cena como um conjunto de primitivas é mais difícil do que uma representação genérica, tal como as descritas por triangulação, mas apresenta maior eficiência na armazenagem (Strand e Dillmann, 2009) e vantagens em aplicações de visão robótica que implique a descrição de formas básicas.

2.2.1 Conceitos das superelipsóides

O modelo matemático de superquádrica é definido pelo produto esférico de duas curvas em 2D, mediante equações implícitas, explícitas ou paramétricas. Por exemplo, considerem-se duas curvas: o círculo $S_1(n)$ no plano (x, y) e um semicírculo $S_2(w)$ ortogonal a ele (Figura 15). A esfera é obtida a partir do produto esférico delas (curva seno-cosseno). De maneira geral, a equação que descreve a operação é:

$$r(n, w) = S_1(n) \otimes S_2(w) = \begin{bmatrix} \cos^{\varepsilon_1} n \\ a_3 \sin^{\varepsilon_1} n \end{bmatrix} \otimes \begin{bmatrix} a_1 \cos^{\varepsilon_2} w \\ a_2 \sin^{\varepsilon_2} w \end{bmatrix} = \begin{bmatrix} a_1 \cos^{\varepsilon_1} n \cos^{\varepsilon_2} w \\ a_2 \cos^{\varepsilon_1} n \sin^{\varepsilon_2} w \\ a_3 \sin^{\varepsilon_1} n \end{bmatrix} \quad \begin{matrix} -\frac{\pi}{2} < n < \frac{\pi}{2} \\ -\pi < w < \pi \end{matrix}$$

(9)

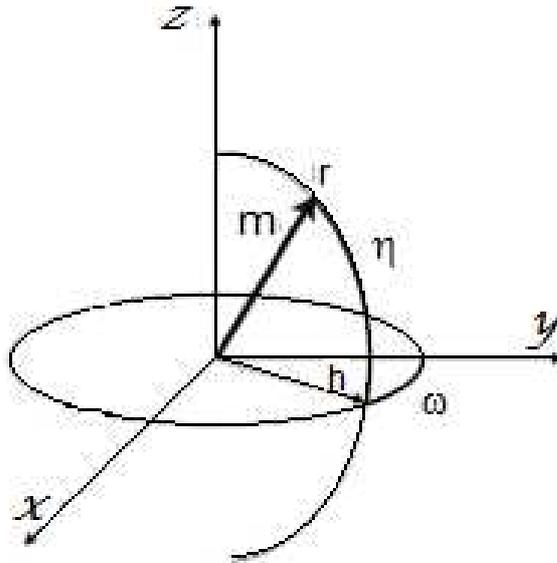


Figura 15. Produto esférico de duas curvas. Adaptado de Jaklic A., et. al. 2000.

O símbolo \otimes representa o produto esférico, os parâmetros n e w correspondem à latitude e longitude do vetor x em coordenadas esféricas, a_1, a_2, a_3 são constantes escalares correspondentes aos três eixos x, y, z , respectivamente. Os dois expoentes ε_1 e ε_2 são parâmetros que descrevem a forma ao longo da longitude e latitude do objeto que têm como objetivo comprimir, arredondar e ajustar partes que estão fora das formas sólidas, para suavizar a nitidez da peça e para produzir bordas e filetes arredondados com qualquer raio e são denominados parâmetros de forma (ε_1 : longitude e ε_2 : latitude), de forma que suas variações fornecem diferentes superelipsóides.

O modelo pode representar diferentes objetos a partir de poucos parâmetros e, na equação 9, são observados cinco parâmetros capazes de gerar uma família de formas geométricas. Na

Figura 16 tem-se uma família de superelipsóides, a forma do objeto é marcada pelos parâmetros ε_1 e ε_2 , que variam entre 0.1 até 2.

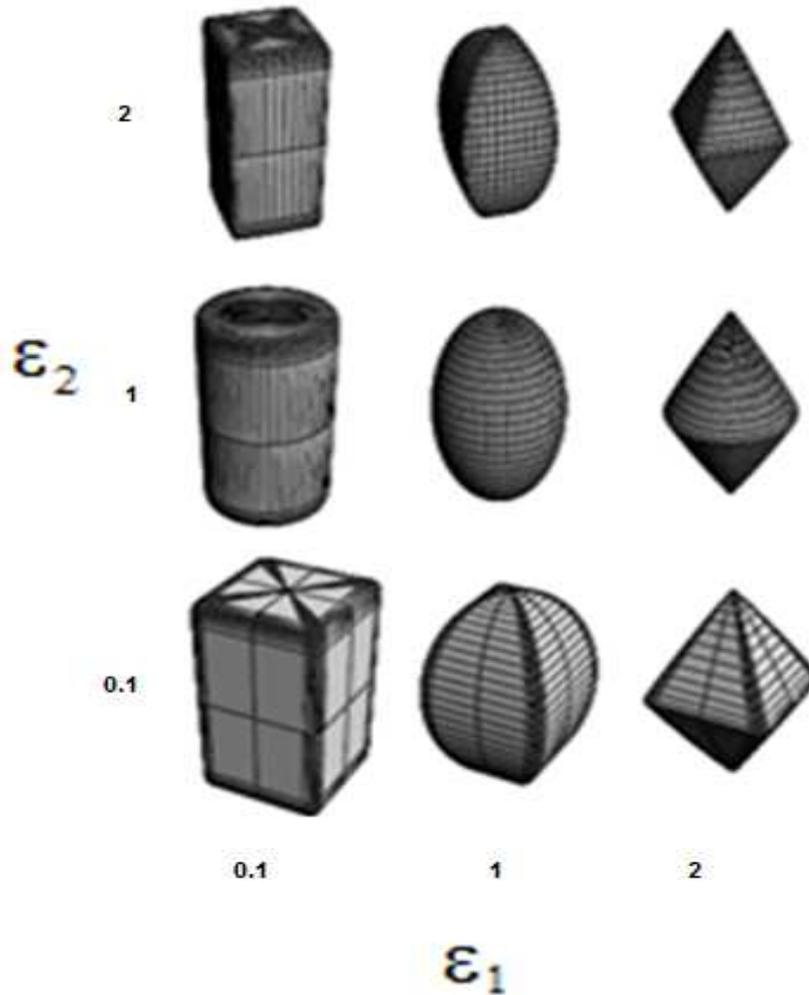


Figura 16: Exemplo de família de superelipsóides. Adaptada de Barr, 1981.

Com a notação paramétrica da equação 9, pode-se obter a equação implícita da superquádrlica. Para o caso das superelipsóides, em primeiro lugar definem-se cada coordenada (x, y, z) com as funções trigonométricas da equação 9 e depois são elevadas ao quadrado:

$$\left(\frac{x}{a_1}\right)^2 = \cos^{2\varepsilon_1} n \cos^{2\varepsilon_2} w \quad (10)$$

$$\left(\frac{y}{a_2}\right)^2 = \cos^{2\varepsilon_1} n \sin^{2\varepsilon_2} w \quad (11)$$

$$\left(\frac{z}{a_3}\right)^2 = \text{sen}^{2\varepsilon_1} n \quad (12)$$

Elevando as duas primeiras equações (10 e 11) a potência $1/\varepsilon_2$, somando-as e substituindo-se a equação trigonométrica $\cos^2\alpha + \text{sen}^2\alpha = 1$, tem se:

$$\left(\frac{x}{a_1}\right)^{2/\varepsilon_2} + \left(\frac{y}{a_2}\right)^{2/\varepsilon_2} = \text{cos}^{2\varepsilon_1/\varepsilon_2} n \quad (13)$$

Agora, eleva-se a terceira equação (12) a potência $1/\varepsilon_1$:

$$\left(\frac{z}{a_3}\right)^{2/\varepsilon_1} = \text{sen}^2 n \quad (14)$$

Se elevar a equação (13) à potência $\varepsilon_2/\varepsilon_1$, e somar (14), então se têm a equação implícita das superelipsóides:

$$\left(\left(\frac{x}{a_1}\right)^{2/\varepsilon_2} + \left(\frac{y}{a_2}\right)^{2/\varepsilon_2}\right)^{\varepsilon_2/\varepsilon_1} + \left(\frac{z}{a_3}\right)^{2/\varepsilon_1} = 1 \quad (15)$$

Barr (1981) descreve três regiões geométricas de uma superelipsóide mediante sua função paramétrica:

- A região onde as coordenadas estão dentro do contorno da superfície.
- A região onde as coordenadas estão na superfície.
- A região onde as coordenadas estão fora do contorno da superfície.

Assim, define-se a equação paramétrica (15), reescrita e denominada de função dentro-fora (equação 16), que fornece um teste simples da região em que um ponto arbitrário se encontra com respeito à superfície definida pelo vetor de parâmetros característicos β .

$$F(\beta; x, y, z) = \left(\left(\frac{x}{a_1}\right)^{2/\varepsilon_2} + \left(\frac{y}{a_2}\right)^{2/\varepsilon_2}\right)^{\varepsilon_2/\varepsilon_1} + \left(\frac{z}{a_3}\right)^{2/\varepsilon_1} \quad (16)$$

tal que,

$F < 1$, então (X, Y, Z) esta dentro da superquádrica.

$F = 1$, então (X, Y, Z) esta no contorno da superquádrica.

$F > 1$, então (X, Y, Z) esta fora da superquádrica.

sendo β o vetor de parâmetros característicos que representam o objeto e será descrito no resultado deste trabalho.

O mesmo procedimento é feito para as outras superquádricas e, na Tabela seguinte, são descritas as equações implícitas e paramétricas para as quatro famílias de superquádricas (superelipsóide, superhiperbolóides de uma folha, superhiperbolóides de duas folhas e supertoróides) e seus modelos representativos correspondentes (Figura 17).

Tabela 1. Equações para cada tipo de Superquádricas

Família de Superquádrica	Curva em 2D	Equação paramétrica	Equação implícita
Superelipsóide	$S_1 = \begin{bmatrix} \cos^{\varepsilon_1} n \\ a_3 \sin^{\varepsilon_1} n \end{bmatrix}$ $S_2 = \begin{bmatrix} a_1 \cos^{\varepsilon_2} n \\ a_2 \sin^{\varepsilon_2} n \end{bmatrix}$	$r(n, w) = \begin{bmatrix} a_1 \cos^{\varepsilon_1} n \cos^{\varepsilon_2} w \\ a_2 \cos^{\varepsilon_1} n \sin^{\varepsilon_2} w \\ a_3 \sin^{\varepsilon_2} w \end{bmatrix} \begin{matrix} -\frac{\pi}{2} < n < \frac{\pi}{2} \\ -\pi < w < \pi \end{matrix}$	$F = \left(\left(\frac{X}{a_1} \right)^{2/\varepsilon_2} + \left(\frac{Y}{a_2} \right)^{2/\varepsilon_2} \right)^{\varepsilon_2/\varepsilon_1} + \left(\frac{Z}{a_3} \right)^{2/\varepsilon_1}$
Superhiperbolóides de uma folha	$S_1 = \begin{bmatrix} \sec^{\varepsilon_1} n \\ a_3 \tan^{\varepsilon_1} n \end{bmatrix}$ $S_2 = \begin{bmatrix} a_1 \cos^{\varepsilon_2} n \\ a_2 \sin^{\varepsilon_2} n \end{bmatrix}$	$r(n, w) = \begin{bmatrix} a_1 \sec^{\varepsilon_1} n \cos^{\varepsilon_2} w \\ a_2 \sec^{\varepsilon_1} n \sin^{\varepsilon_2} w \\ a_3 \tan^{\varepsilon_2} w \end{bmatrix} \begin{matrix} -\frac{\pi}{2} < n < \frac{\pi}{2} \\ -\pi < w < \pi \end{matrix}$	$F = \left(\left(\frac{X}{a_1} \right)^{2/\varepsilon_2} + \left(\frac{Y}{a_2} \right)^{2/\varepsilon_2} \right)^{\varepsilon_2/\varepsilon_1} - \left(\frac{Z}{a_3} \right)^{2/\varepsilon_1}$
Superhiperbolóides de duas folhas	$S_1 = \begin{bmatrix} \sec^{\varepsilon_1} n \\ a_3 \tan^{\varepsilon_1} n \end{bmatrix}$ $S_2 = \begin{bmatrix} a_1 \sec^{\varepsilon_2} n \\ a_2 \tan^{\varepsilon_2} n \end{bmatrix}$	$r(n, w) = \begin{bmatrix} a_1 \sec^{\varepsilon_1} n \sec^{\varepsilon_2} w \\ a_2 \sec^{\varepsilon_1} n \tan^{\varepsilon_2} w \\ a_3 \tan^{\varepsilon_2} w \end{bmatrix} \begin{matrix} -\frac{\pi}{2} < n < \frac{\pi}{2} \\ -\frac{\pi}{2} < w < \frac{\pi}{2} \text{ folha 1} \\ \frac{\pi}{2} < w < \frac{3\pi}{2} \text{ folha 2} \end{matrix}$	$F = \left(\left(\frac{X}{a_1} \right)^{2/\varepsilon_2} - \left(\frac{Y}{a_2} \right)^{2/\varepsilon_2} \right)^{\varepsilon_2/\varepsilon_1} - \left(\frac{Z}{a_3} \right)^{2/\varepsilon_1}$
Supertoróides	$S_1 = \begin{bmatrix} a_4 + \cos^{\varepsilon_1} n \\ a_3 \sin^{\varepsilon_1} n \end{bmatrix}$ $S_2 = \begin{bmatrix} a_1 \cos^{\varepsilon_2} n \\ a_2 \sin^{\varepsilon_2} n \end{bmatrix}$	$r(n, w) = \begin{bmatrix} a_1(a_4 + \cos^{\varepsilon_1} n) \cos^{\varepsilon_2} w \\ a_2(a_4 + \cos^{\varepsilon_1} n) \sin^{\varepsilon_2} w \\ a_3 \tan^{\varepsilon_2} w \end{bmatrix} \begin{matrix} -\pi < n < \pi \\ -\pi < w < \pi \end{matrix}$	$F = \left(\left(\frac{X}{a_1} \right)^{2/\varepsilon_2} + \left(\frac{Y}{a_2} \right)^{2/\varepsilon_2} \right)^{\varepsilon_2/2} - a_4 \left(\frac{Z}{a_3} \right)^{2/\varepsilon_1} + \left(\frac{Z}{a_3} \right)^{2/\varepsilon_1}$

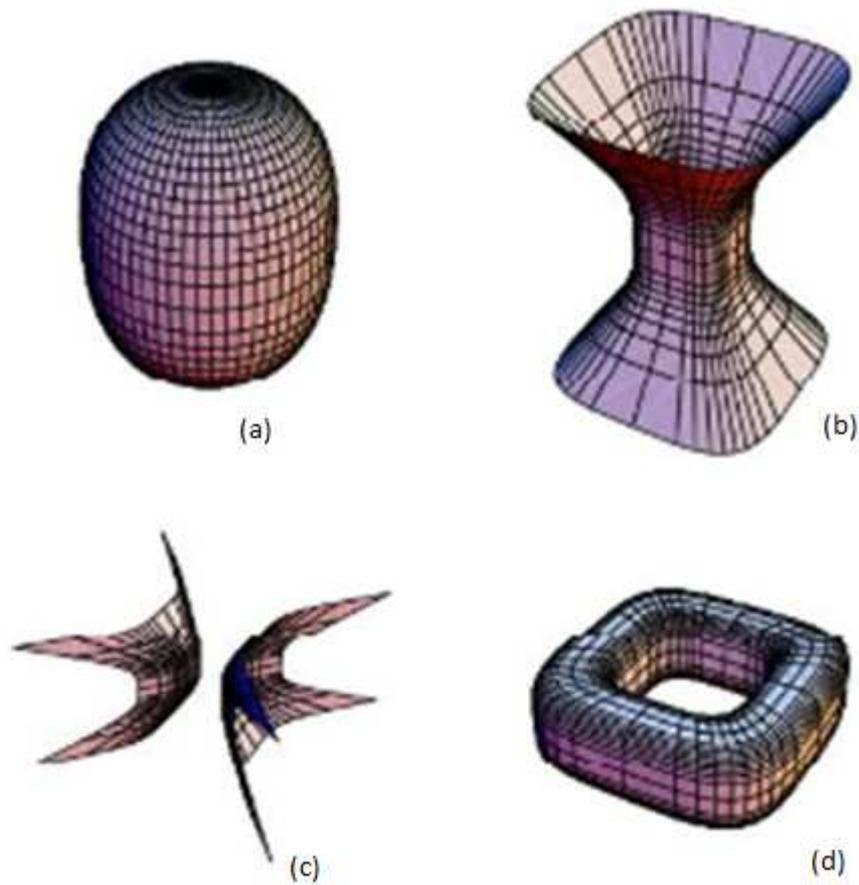


Figura 17. Família de superquádricas. a) Superelipsóides, b) superhiperbolóides de uma folha, c) superhiperbolóides de duas folhas e d) supertoróides. Adaptado de Jaclik A., et. al., 2000.

A representação de um objeto por estas superfícies é dada pela descrição do vetor de parâmetros e depende da complexidade do objeto alvo para se conhecer a quantidade de parâmetros necessários. Por isso, é necessário também estudar as propriedades geométricas das superquádricas e, neste caso, das superelipsóides.

2.2.2 Propriedades geométricas das superelipsóides

No método das superquádricas, as superelipsóides são definidas num espaço local (X, Y, Z) com a origem no centro da superelipsóide (equação 16). Para mudar para um sistema global (x, y, z) , as coordenadas dependem de um deslocamento definido pela matriz inversa de rotação e pelo vetor de translação (X_c, Y_c, Z_c) , que, ao longo do processo, este vetor de translação se reconhece como o centróide da nuvem de pontos. A representação matricial dos pontos (x, y, z) esta dada pela equação (17):

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & Xc \\ \cdot & R & \cdot & Yc \\ \cdot & \cdot & \cdot & Zc \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (17)$$

E a representação matricial dos pontos (X, Y, Z) será:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & 1 \\ \cdot & R^{-1} & \cdot & 1 \\ \cdot & \cdot & \cdot & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - Xc \\ y - Yc \\ z - Zc \\ 1 \end{bmatrix} \quad (18)$$

Considerando que as rotações são em torno dos eixos de coordenadas, a matriz de rotação se define como o produto das três matrizes R_x , R_y e R_z

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\text{sen} \phi \\ 0 & \text{sen} \phi & \cos \phi \end{bmatrix} \quad (19)$$

$$R_y = \begin{bmatrix} \cos \theta & 0 & \text{sen} \theta \\ 0 & 1 & 0 \\ -\text{sen} \theta & 0 & \cos \theta \end{bmatrix} \quad (20)$$

$$R_z = \begin{bmatrix} \cos \psi & -\text{sen} \psi & 0 \\ \text{sen} \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (21)$$

e a inversa R^{-1} se define segundo a equação 22:

$$R^{-1} = \begin{bmatrix} \cos \phi \cos \theta & -\cos \theta \text{sen} \phi & -\text{sen} \theta \\ \cos \psi \text{sen} \phi - \cos \phi \text{sen} \psi \text{sen} \theta & \cos \phi \cos \psi + \text{sen} \phi \text{sen} \psi \text{sen} \theta & -\cos \theta \text{sen} \psi \\ \text{sen} \phi \text{sen} \psi + \cos \phi \cos \psi \text{sen} \theta & \cos \phi \text{sen} \psi - \cos \psi \text{sen} \phi \text{sen} \theta & \cos \psi \cos \theta \end{bmatrix} \quad (22)$$

sendo ϕ , θ e ψ os ângulos de Euler que expressam as três rotações consecutivas em torno dos eixos x, y e z, respectivamente. Portanto, as coordenadas dos pontos do objeto no espaço local X, Y, Z são definidas assim:

$$X = \cos \theta \cos \phi (x - Xc) + (-\cos \theta \text{sen} \phi)(y - Yc) + \text{sen} \theta (z - Zc) \quad (23)$$

$$Y = (\cos\psi\sin\phi - \cos\phi\sin\psi\sin\theta)(x - X_c) + (\cos\phi\cos\psi + \sin\phi\sin\psi\sin\theta)(y - Y_c) + (-\cos\theta\sin\psi)(z - Z_c) \quad (24)$$

$$Z = (\sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta)(x - X_c) + (\cos\phi\sin\psi - \cos\psi\sin\phi\sin\theta)(y - Y_c) + (\cos\psi\cos\theta)(z - Z_c) \quad (25)$$

Finalmente, substituindo-se as equações anteriores (23, 24 e 25) na equação implícita (equação 16) a superelipsóide é descrita no espaço global. Então, identifica-se o número de parâmetros para representar a forma paramétrica que, neste caso, tem onze parâmetros característicos representados pelo vetor β : os de tamanho a_1, a_2 e a_3 ; os de forma ε_1 e ε_2 ; e os de posição (rotação e translação), $\phi, \theta, \varphi, X_c, Y_c, Z_c$. Assim, o vetor é representado por:

$$\beta = (a_1, a_2, a_3, \varepsilon_1, \varepsilon_2, \phi, \theta, \varphi, X_c, Y_c, Z_c) \quad (26)$$

Tem-se, então, uma função que é uma combinação não linear de parâmetros que dependem de onze variáveis independentes. O objetivo é encontrar o vetor β , o valor dos parâmetros em que os pontos da nuvem representem o objeto. Portanto, a eficiência do modelo depende do método para encontrar os parâmetros e da otimização do algoritmo de aproximações sucessivas, descrito adiante.

2.2.3 Função objetivo

O conjunto dos parâmetros da superelipsóide expressos no vetor da equação (26) para o qual o modelo se ajuste a uma nuvem de pontos deve ser achado pela minimização da função objetivo. A função objetivo, que indica distância euclidiana da nuvem de pontos às superelipsóides, é definida pela distância euclidiana entre as funções implícitas F_i e $\widehat{F}(x, y, z; \beta)$, sendo F_i a função modelo da nuvem de pontos e $\widehat{F}(x, y, z; \beta)$ a superelipsóide expressa pelo vetor β .

O modelo matemático unificado (Barr, 1981) apresenta a definição para a distância euclidiana da seguinte forma: considerando-se uma nuvem de N pontos posicionados no sistema de coordenadas global, o objetivo consiste em variar os onze parâmetros da equação da superelipsóide para obter a menor distância entre esses pontos e o objeto que será recuperado.

Então, o processo de ajuste corresponde à minimização da soma do quadrado da distância dos pontos ao modelo:

$$d(\boldsymbol{\beta}) = \sum_{i=1}^N (F(\boldsymbol{\beta}; X_i, Y_i, Z_i) - 1)^2 \quad (27)$$

No entanto, Solina e Bajcsy (1990) e Jaklic (2000) comentam que essa definição de distância euclidiana não é certa para modelos de menor volume e multiplicam a função implícita por $\sqrt{a_1 \cdot a_2 \cdot a_3}$ para considerar os objetos de maior volume. (Katsoulas, 2004) acredita que a definição da equação (27) só abrange modelos de elevada curvatura na superfície, por isso, elevou a função a uma potência ε_1 . Assim, a função objetivo torna-se independente da forma da superfície controlada e a definição da distância euclidiana não gerou rápido crescimento numérico da função para pequenos valores de ε_1 . Então, Katsoulas (2004) propõe a definição da função objetivo como segue:

$$d1(\boldsymbol{\beta}) = \sum_{i=1}^N (\sqrt{a_1 \cdot a_2 \cdot a_3} (F^{\varepsilon_1}(\boldsymbol{\beta}; X_i, Y_i, Z_i) - 1))^2 \quad (28)$$

Para esses autores, a principal vantagem no uso dessa função objetivo foi à eficiência computacional da modelagem. No entanto, Chevalier *et al.* (2003) fez uma análise desta expressão e afirmou que os resultados obtidos poderiam divergir daqueles esperados quando dois modelos em escalas diferentes foram comparados, sobretudo para objetos alongados. De fato, Jaklic (2000) observou que esta representação não produz valores uniformemente distribuídos para a distância em todas as direções do modelo, bem como apresenta dependência para com os parâmetros de tamanho. Além disso, a minimização do escalar $(F^{\varepsilon_1} - 1)$ ainda favoreceu modelos de maiores volumes, como foi inicialmente demonstrado por Whaite e Ferrie (1991) e Katsoulas (2004). Tal comportamento verificou-se pela análise da distância euclidiana radial, proposta por Whaite e Ferrie (1991). A qualidade dos resultados obtidos usando a distância euclidiana radial não compensou os custos computacionais adicionais introduzidos no processo de ajuste (Katsoulas, 2004). Desse modo, a função objetivo para este processo de ajuste é a versão modificada da equação (28), apresentada por Jacklic e Solina (2003):

$$d(\boldsymbol{\beta}) = \sum_{i=1}^N (F^{\frac{\varepsilon_1}{2}}(\boldsymbol{\beta}; X_i, Y_i, Z_i) - 1)^2 \quad (29)$$

Sendo X_i, Y_i, Z_i a coordenada espacial de um ponto do objeto e β o vetor de n parâmetros. O processo de ajuste desta função não linear de onze parâmetros se faz com o algoritmo de minimização iterativo de Levenberg-Marquardt (1963), (Solina e Bajcsy, 1990), (Dop e Regtien 1996), (Leonardiset al., 1997), (Jaklicet al., 2000), (Jaklic e Solina, 2003), (Chevalier et al., 2003 e 2004), (El-Khoury e Sahbani, 2008), (Zhang, 2003)

2.3 ALGORITMO LEVENBERG-MARQUARDT

O algoritmo Levenberg-Marquardt (publicado primeiramente por Kenneth Levenberg em 1944 e aperfeiçoado por Donald Marquardt em 1963) é um algoritmo de regressão não linear. O problema de otimização envolvida na minimização da função da superquádrica é geralmente resolvido pelo algoritmo de Levenberg-Marquardt, sendo que o desempenho e a eficiência do algoritmo de representação dependem principalmente da estimação da função objetivo (Zhang 2003). O processo consiste em ajustar uma função objetivo vetorial $F(t, \mathbf{p})$, com t variáveis independentes e um vetor de n parâmetros \mathbf{p} , para um conjunto de pontos de dados (t_i, F_i) . Para isso, usa a combinação dos métodos “Steepest Descent e Gauss Newton” da seguinte forma: quando o resultado se encontra muito longe do valor correto, então se trabalha com o segundo método. A sua maximização na eficiência é obtida pela restrição numérica dos parâmetros a serem ajustados.

2.3.1 Modelo de minimização

O algoritmo Levenberg-Marquardt (Lourakis, 2005) define uma função de minimização representada pela soma dos quadrados da diferença entre as funções vetoriais do modelo F_i e o conjunto de dados que vai se ajustar, $\hat{F}(t_i, \mathbf{P})$ (equação 30). Essa diferença é chamada de erro residual ou erro ponderado. A função de minimização é o critério de erro chi-quadrado, descrito na equação (31) e de modo matricial na equação (33):

$$e = F_i - \hat{F}(t_i, \mathbf{P}) \quad (30)$$

$$\gamma^2(\mathbf{P}) = \frac{1}{2} \sum_{i=1}^m \left[\frac{F_i - \hat{F}(t_i, \mathbf{P})}{w_i} \right]^2 \quad (31)$$

$$\gamma^2(\mathbf{P}) = \frac{1}{2} (F_i - \hat{F}(t, \mathbf{P}))^T W (F_i - \hat{F}(t, \mathbf{P})) \quad (32)$$

sendo w_i o erro de medida e F_i e W a matriz de ponderação diagonal com dados $W_{im} = 1/w_i^2$.

O objetivo do algoritmo é, mediante uma minimização iterativa, encontrar uma perturbação h para os parâmetros P que reduzam o valor de γ^2 .

No método de Gradiente descendente (*SteepestDescent*), a perturbação $h = h_{gd}$ que movimenta os parâmetros na direção do passo descendente é dada por:

$$h_{gd} = \alpha J^T W (F_i - \hat{F}(t, \mathbf{P})) \quad (33)$$

Sendo J a matriz jacobiana e α um escalar positivo que determina o comprimento do passo na direção descendente.

No método de Gauss Newton, tem-se a equação normal resultante para a perturbação h_{gn} :

$$(J^T W J) h_{gn} = J^T W (F_i - \hat{F}(t, \mathbf{P})) \quad (34)$$

Então, o algoritmo de Levenberg-Marquardt é a combinação destes métodos anteriores, dada pela seguinte equação:

$$(J^T W J + \lambda I) h_{lm} = J^T W (F_i - \hat{F}(t, \mathbf{P})) \quad (35)$$

sendo λ o parâmetro amortecedor que influencia tanto na direção quanto no tamanho do passo. Por isso, se o valor de λ é muito pequeno, então, utiliza-se o algoritmo Gauss Newton, caso contrário, o resultado se atualiza com o algoritmo gradiente descendente.

2.3.2 Critérios de convergência

O processo de ajustar uma função até outra função que descreve uma nuvem de pontos desconhecida estabelece os critérios de convergência, que devem ser verificados em cada iteração:

- Quando a magnitude do gradiente de $|J^T W(F_i - \hat{F}(t, \mathbf{P}))|$ é menor que um limite k_1 ;
- A mudança relativa na magnitude de $|\frac{h_{lm}}{\mathbf{P}}|$ abaixo de um k_2 ;
- O erro $\gamma^2(\mathbf{P})$ é menor que o limiar k_3 ;
- A quantidade máxima de iterações foi atingida.

3 MÉTODO DE RECONSTRUÇÃO TRIDIMENSIONAL DE OBJETOS

O seguinte capítulo é o método de reconstrução 3D que será implementado e validado para as superelipsóides. Observa-se o processo geral do método no diagrama da Figura 18, que consiste no desenvolvimento de cinco etapas sequenciais.

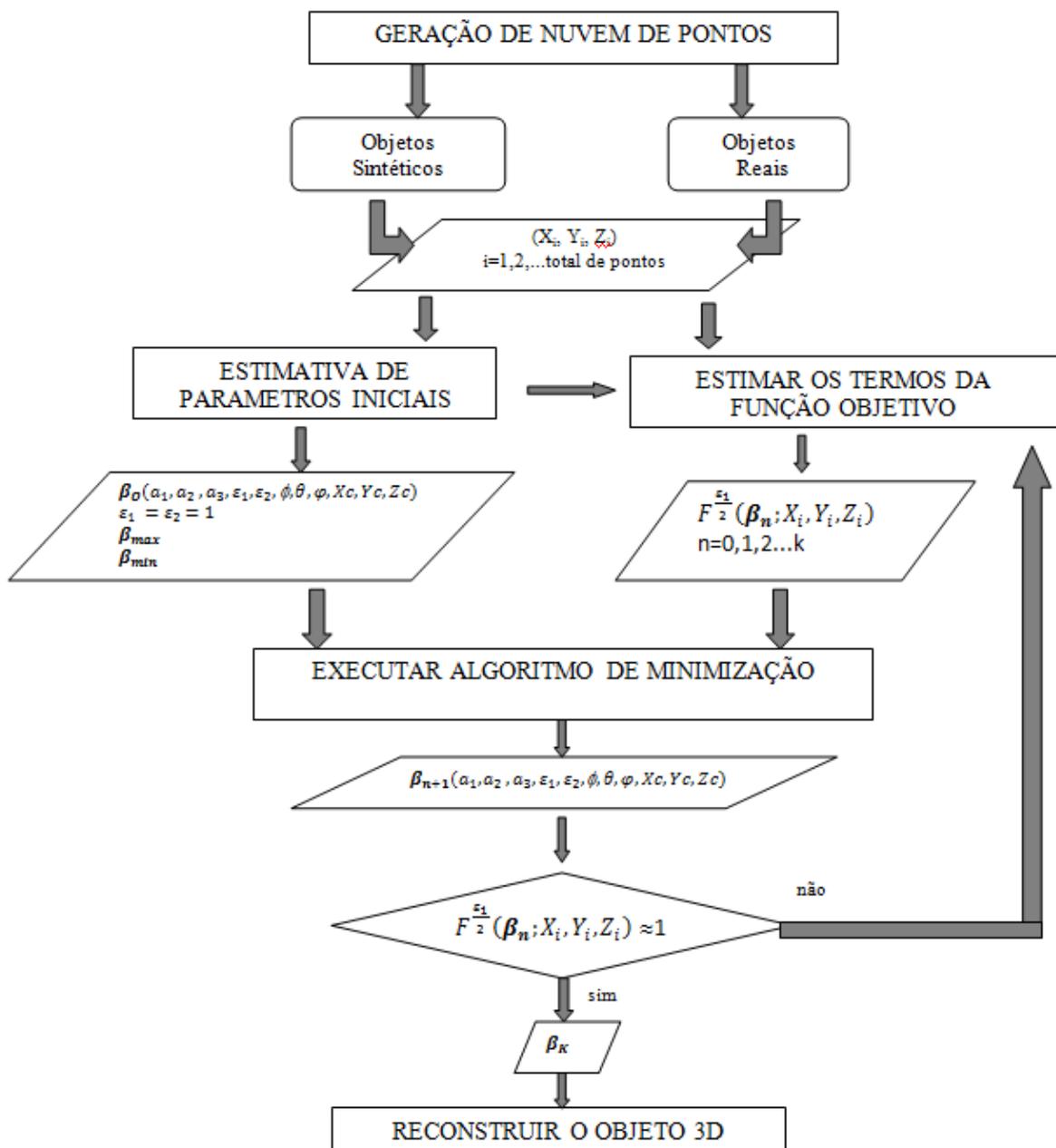


Figura 18. Etapas de desenvolvimento

A primeira etapa, “Geração da nuvem de pontos”, consiste em obter os pontos espaciais de um objeto. Neste trabalho, as nuvens de pontos são obtidas 1). Implementando o algoritmo do modelo matemático da superelipsóide e 2). Usando o sistema de aquisição de imagens de profundidade, o sistema *kinect*.

A etapa de “Estimativa dos parâmetros iniciais da nuvem” é feita para estimar 11 parâmetros característicos de uma superelipsóide: três parâmetros que indicam a rotação, três que estimam a posição, três que definem as dimensões da nuvem de pontos e dois parâmetros que definem a forma da nuvem, com o valor inicial de 1, o que corresponde a uma elipsóide. Esta é uma etapa de maior importância, posto que a resposta do algoritmo de minimização dependa do método para definir os parâmetros e dos mínimos e máximos de cada parâmetro.

Em seguida, tem-se a terceira etapa, “Estimar os termos da função objetivo”, que consiste em implementar um algoritmo que estima a função $F(\boldsymbol{\beta}; x, y, z)$ da equação (16) cujo resultado será ajustado, tantas vezes como seja necessário, pela seguinte etapa de “Executar o algoritmo de minimização”. Essa etapa consiste em ajustar a função de minimização até conseguir o vetor $\boldsymbol{\beta}$ das características da nuvem de pontos do objeto e representar por meio deste vetor o objeto em 3D.

3.1 GERAÇÃO DA NUVEM DE PONTOS

Com o fim de validar o método implementado para a reconstrução 3D dos objetos, implementa-se o algoritmo para gerar uma nuvem de pontos de uma superelipsóide. A superelipsóide é determinada pela equação (9), formada por duas curvas, tal que uma curva é gerada sobre o plano x-y e a outra curva é ortogonal à primeira. Os parâmetros geométricos do objeto, $\boldsymbol{\beta}$, são à entrada da equação (9), e a saída do algoritmo é a nuvem de pontos do objeto. O algoritmo implementado para este processo é o seguinte:

ALGORITMO PARA GERAÇÃO DA NUVEM DE PONTOS (Apêndice 1)

- O usuário fornece os seguintes parâmetros:
 - O tamanho do objeto: a_1, a_2, a_3 que correspondem aos eixos da superelipsóide.
 - A forma para a primeira curva ε_2 e para a segunda ε_1 no intervalo $(0,1;$ dois)
 - Os ângulos de Euler ϕ, θ, φ . Para estimar a rotação e translação da nuvem.
- Definir um vetor no intervalo $(-\pi \leq w \leq \pi)$.
- Criar a superfície $S_2(w) = \begin{bmatrix} a_1 \cos^{\varepsilon_2} w \\ a_2 \sin^{\varepsilon_2} w \end{bmatrix}$
- Definir um vetor n no intervalo $(-\pi/2 \leq n \leq \pi/2)$.
- Criar a superfície $S_1(n) = \begin{bmatrix} \cos^{\varepsilon_1} n \\ a_3 \sin^{\varepsilon_1} n \end{bmatrix}$
- Representar o objeto 3D $r(n, w) = S_1(n) \otimes S_2(w)$.

Também, nesta etapa é obtida a imagem de profundidade com o sistema *Kinect* as aplicações do conjunto de APIs conhecidas como OpenNI (*Open Natural Interaction*), que permitem a configuração para capturar as coordenadas da cena. O modelo da cena (Figura 19) consiste em posicionar a câmera do *Kinect*® em linha com o objeto alvo, separados por uma distância de no máximo 90 cm (distância para que o sistema capture a nuvem de pontos do objeto) e uma altura que depende do tamanho e da posição do objeto. Com essa configuração, capturam-se as imagens de profundidade de, no mínimo, três faces do objeto, ou a maior área dele, num ambiente real, afetados pelo ruído do entorno.

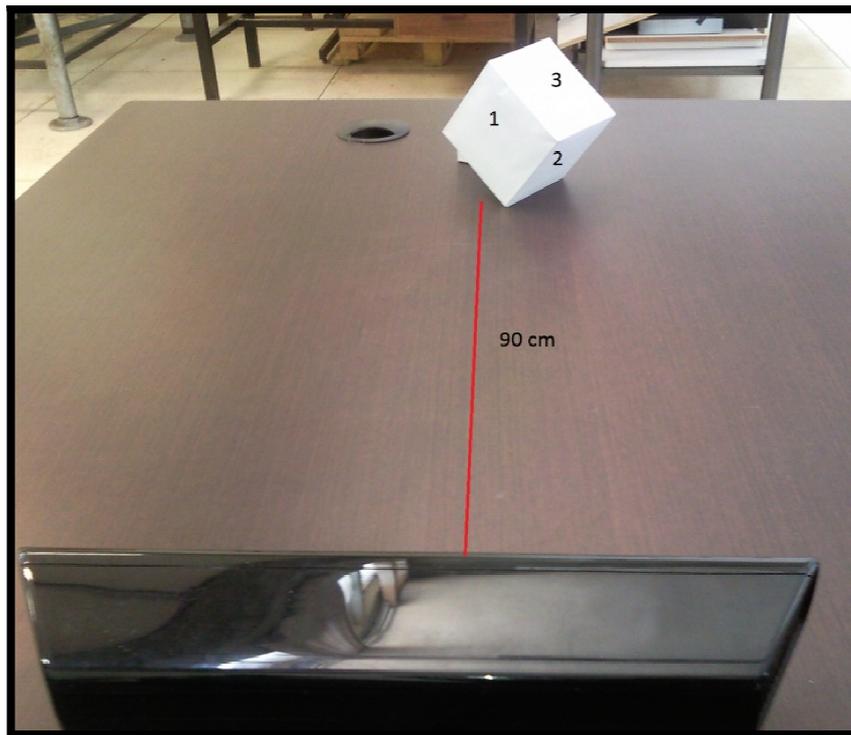


Figura 19. Localização do objeto e o Sistema Kinect para capturar imagens de profundidade

Para se segmentar a imagem, utilizou-se o *software MeshLab*, capaz de editar e reestruturar a imagem capturada pelo sistema e o *software 3D object*, usado para criar objetos 3D e salvar as coordenadas espaciais do objeto em arquivos tipo “txt”.

3.2 ESTIMATIVA DOS PARÂMETROS INICIAIS DA NUVEM DE PONTOS

Os parâmetros iniciais conformam o vetor β inicial e são obtidos a partir das seguintes características da nuvem de pontos do objeto alvo:

- Posição;
- Orientação;
- Tamanho;
- Forma inicial de uma elipsóide.

Além disso, para cada parâmetro são escolhidos os valores mínimos e os máximos para que o algoritmo de minimização tenha limites na busca da função de ajuste.

3.2.1 Posição global da nuvem

Considera-se uma imagem de profundidade para N pontos espaciais $\mathbf{P}_i = (X_i, Y_i, Z_i)$, sua posição global pode ser descrita pelo centróide da nuvem mediante a seguinte equação:

$$(X_c, Y_c, Z_c) = \sum_{i=1}^N \frac{X_i}{N}, \frac{Y_i}{N}, \frac{Z_i}{N} \quad (36)$$

Como mencionado anteriormente, o centróide da nuvem é a origem das coordenadas do vetor de translação. Assim o cálculo do deslocamento da nuvem de pontos à origem do sistema de coordenadas global (Figura 20) foi realizado com o algoritmo seguinte:

ALGORITMO PARA ESTIMATIVA DA POSIÇÃO GLOBAL DA NUVEM (Apêndice 2)

- Separar as coordenadas x, y, z dos pontos da nuvem de pontos do objeto alvo por meio de vetores.
- Achar o número total de pontos N.
- Calcular $X_c = \sum_{i=1}^N \frac{X_i}{N}$, $Y_c = \sum_{i=1}^N \frac{Y_i}{N}$, $Z_c = \sum_{i=1}^N \frac{Z_i}{N}$
- Transladar o centróide da nuvem de pontos para a origem do sistema de coordenadas (X, Y, Z) global $X_i = x_i - X_c$, $Y_i = y_i - Y_c$, $Z_i = z_i - Z_c$
- Salvar o conjunto de coordenadas (X_i, Y_i, Z_i).

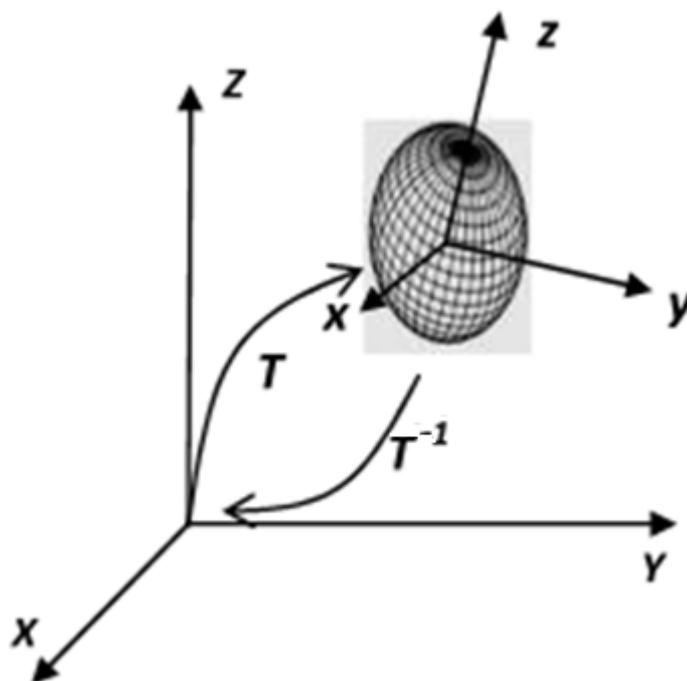


Figura 20. Translação do sistema de coordenadas local ao sistema de coordenadas global de um objeto no espaço

Os valores máximos e mínimos foram calculados pela margem de tolerância obtida nos ensaios experimentais, descrito mais detalhadamente no Capítulo 4.

3.2.2 Orientação da nuvem

A orientação é o parâmetro que fornece informações para a obtenção da matriz de rotação R (sistema da equação (17)). Existem vários procedimentos para determinar esta matriz, entre eles os de Strand e Dillmann (2009), Bardinet (1995) e Chevalier (2004). O método proposto por Strand e Dillmann (2009) define a orientação da nuvem de pontos pelos eixos de inércia da matriz de covariância da equação 37:

$$C = \frac{1}{N} \sum_{i=1}^N \begin{bmatrix} (X_i - X_c)(X_i - X_c) & (Y_i - Y_c)(X_i - X_c) & (Z_i - Z_c)(X_i - X_c) \\ (Y_i - X_c)(X_i - X_c) & (Y_i - Y_c)(Y_i - Y_c) & (Z_i - Z_c)(Y_i - Y_c) \\ (Z_i - X_c)(X_i - X_c) & (Y_i - Y_c)(Z_i - Z_c) & (Z_i - Z_c)(Z_i - Z_c) \end{bmatrix} \quad (37)$$

Essa matriz descreve o objeto matematicamente com os autovalores e autovetores. Os autovetores representam os eixos de inércia obtidos da matriz de rotação e os autovalores são fornecidos pela matriz diagonal da covariância:

$$\mathbf{v} = \mathbf{autovetores}(C) = (\mathbf{v}_1 \ \mathbf{v}_2 \ \mathbf{v}_3) = \begin{bmatrix} \mathbf{v}_{1,1} & \mathbf{v}_{2,1} & \mathbf{v}_{3,1} \\ \mathbf{v}_{1,2} & \mathbf{v}_{2,2} & \mathbf{v}_{3,2} \\ \mathbf{v}_{1,3} & \mathbf{v}_{2,3} & \mathbf{v}_{3,3} \end{bmatrix} \quad (38)$$

Como a matriz de autovetores (equação 38) corresponde à matriz de rotação R da nuvem de pontos e a orientação inicial é determinada pelos ângulos de Euler ϕ , θ e φ , fez-se assim: As matrizes de rotação R_x , R_y e R_z , descritas no item 2.2, representam a rotação de ϕ , θ e φ em relação ao *eixo-x*, *eixo-y* e *eixo-z* respectivamente, o que se conhece como uma combinação xyz. A multiplicação de $R_x * R_y * R_z$ gera a matriz de rotação tridimensional R :

$$\mathbf{R} = \begin{bmatrix} \cos\phi\cos\theta & \cos\varphi\sin\phi - \cos\phi\sin\varphi\sin\theta & \sin\phi\sin\varphi + \cos\phi\cos\varphi\sin\theta \\ -\cos\theta\sin\phi, & \cos\phi\cos\varphi + \sin\phi\sin\varphi\sin\theta & \cos\phi\sin\varphi - \cos\varphi\sin\phi\sin\theta \\ \cos\theta, & -\cos\theta\sin\varphi, & \cos\varphi\cos\theta \end{bmatrix} \quad (39)$$

sendo que as matrizes \mathbf{R} e \mathbf{v} definem a rotação da nuvem de pontos, então sua equivalência dá como resultado os valores para cada ângulo de rotação:

$$\phi = \arctan\left(\frac{\mathbf{v}_{2,3}}{\mathbf{v}_{1,3}}\right) \quad (40)$$

$$\theta = \arccos(\mathbf{v}_{3,3}) \quad (41)$$

$$\varphi = \arctan\left(\frac{\mathbf{v}_{3,2}}{-\mathbf{v}_{3,1}}\right) \quad (42)$$

O algoritmo para construir esta etapa foi o seguinte:

ALGORITMO PARA ESTIMATIVA DA ORIENTAÇÃO DA NUVEM (Apêndice 3)

- Determinar a matriz de covariância C da equação (37)
- Determinar os autovetores $\mathbf{v}_{1,2,3}$ da matriz de covariância e formar a matriz da equação (38).

- Estimar os ângulos de Euler utilizando as equações (40,41 e 42).

O valor máximo e mínimo para cada ângulo depende da definição do ângulo de rotação em relação ao eixo correspondente e da posição espacial onde se encontra. Isso significa que é preciso identificar o quadrante em que se encontra o ângulo de Euler e configurar a orientação do objeto, Strand e Dilmann (2009).

3.2.3 Tamanho da nuvem

Solina e Bajcsy (1990) definem a direção dos eixos no espaço local da seguinte maneira: Se $(\lambda_1, \lambda_2$ e $\lambda_3)$ são os autovalores dos autovetores v_1, v_2 e v_3 da matriz de covariância, então quando: $|\lambda_1 - \lambda_2| < |\lambda_2 - \lambda_3|$; a direção do novo eixo z é definida pelas coordenadas do autovetor v_3 . Caso contrário, a direção deste eixo é indicada pelo autovetor v_1 . Em seguida, podem-se determinar os parâmetros de tamanho, ou seja, o comprimento dos eixos da nuvem de pontos, estimando os pontos máximos a_i de cada um dos eixos de inércia v_i :

$$a_i = \max_{j=1}^N \{((X_j, Y_j, Z_j) - (X_c, Y_c, Z_c)) \cdot v_i\} \quad (43)$$

O método desenvolvido por Solina e Bajcsy (1990) gerou bons resultados na determinação do tamanho. Por isso, o algoritmo é:

ALGORITMO PARA ESTIMATIVA DO TAMANHO (Apêndice 4)

- Obter os dados dos autovalores da matriz de covariância λ_1, λ_2 e λ_3 .
- Achar o eixo principal z segundo a ordem crescente dos autovalores.
- Caso $|\lambda_1 - \lambda_2| < |\lambda_2 - \lambda_3|$ então a_1, a_2 e a_3 correspondem, respectivamente, aos autovetores v_1, v_2 e v_3 , senão a_1, a_2 e a_3 correspondem, respectivamente, aos autovetores v_3, v_2 e v_1 ou v_3, v_1 e v_2 , dependendo da ordem crescente dos autovalores.
- Estimar o tamanho da nuvem de pontos com a equação (40).

3.2.4 Declaração das condições iniciais dos parâmetros da função objetivo

O vetor inicial de onze parâmetros β_o é o vetor de um objeto com a posição, orientação e tamanho da nuvem de pontos do objeto alvo, mas forma inicial de uma elipsóide. Os onze parâmetros são representados assim:

$$\beta_o = (a_1, a_2, a_3, \varepsilon_1, \varepsilon_2, \phi, \theta, \varphi, X_c, Y_c, Z_c), \text{ com } \varepsilon_1, \varepsilon_2 = 1. \quad (44)$$

Para garantir que o método de minimização apresente maior convergência para resultados particularmente desejados, foi necessário restringir o domínio de valores para os parâmetros; por isso, cada parâmetro tem seu valor mínimo e máximo, criando-se um intervalo de variação dos mesmos

Os valores máximo e mínimo foram calculados pela margem de tolerância obtida nos ensaios experimentais, de tal forma que:

- O valor mínimo e o máximo para os parâmetros de tamanho (a_1, a_2, a_3) e posição (X_c, Y_c, Z_c) são determinados pela margem de tolerância dos ensaios experimentais ($\pm 5\%$). Sendo que, o valor mínimo é 0,95 vezes o valor encontrado e o valor máximo é 1,05 vezes o valor encontrado.
- O intervalo para encontrar os parâmetros de forma é: (0.1; 2).

Portanto, nesta seção foram definidos dois vetores: 1). β_{max} , vetor dos valores máximos de cada parâmetro e 2). β_{min} , vetor dos valores mínimos de cada parâmetro.

3.3 ESTIMATIVA DA FUNÇÃO OBJETIVO

A função objetivo é a distância euclidiana entre as funções explícitas da superelipsóide F_i e $\hat{F}(\beta_n; X_i, Y_i, Z_i)$, definida na equação 29. A função F_i é igual a um porque é o valor quando os pontos da nuvem encontram-se na fronteira da superelipsóide; por outro lado, $\hat{F}(\beta_n; X_i, Y_i, Z_i)$ é a função da superquádrica, mas, quando calculada em cada ponto da

nuvem (x_i, y_i, z_i) com um vetor de parâmetros β_n , para $n = 0, 1, 2, \dots, k$. Sendo β_n o vetor de parâmetros ajustado à nuvem de pontos do objeto alvo, e k número máximo de iterações.

O algoritmo para estimar o termo $F^{\frac{\epsilon_1}{2}}(\beta_n; X_i, Y_i, Z_i)$, sendo que (X_i, Y_i, Z_i) são os pontos do sistema de coordenadas global (X, Y, Z) , é o seguinte:

ALGORITMO PARA: ESTIMAR OS TERMOS DA FUNÇÃO OBJETIVO (Apêndice 5)

- Rotar e transladar a nuvem de pontos resolvendo o sistema da equação (18).
- Chamar o vetor de parâmetros β_n , para $n = 0, 1, 2, \dots, k$. Sendo k o número de iterações definido para o método de minimização.
- Determinar o valor da função explícita da superelipsóide $F(\beta; x, y, z)$ com a equação 16.
- Estimar o termo $F(\beta; x, y, z)$ elevada à potência $\frac{\epsilon_1}{2}$ definida para distância euclidiana na equação (29).
- Estimar a função objetivo (equação 29) para cada ponto da nuvem.

Na equação (29), observa-se que os termos exponenciais da função explícita da equação (16) são da forma x^ϵ porém, podem ser obtidos valores pertencentes ao conjunto dos números complexos. Por isso, é fundamental garantir que o resultado não seja um valor descrito pelo conjunto dos números complexos quando x for um valor real negativo, tendo em conta que o cálculo dos termos exponenciais da função objetivo sejam definidos da seguinte forma:

$$x^\epsilon = \text{sin}(x)|x|^\epsilon = \begin{cases} -|x|^\epsilon, & \text{para } x < 0 \\ +|x|^\epsilon, & \text{para } x > 0 \end{cases} \quad (45)$$

3.4 ALGORITMO DE MINIMIZAÇÃO: MÉTODO DE LEVENBERG-MARQUARDT

Por sua estabilidade e eficiência computacional comprovada por (Zhang, 2003), o algoritmo de Levenberg-Marquardt é extensivamente usado para minimizar funções objetivas (Solina e Bajcsy, 1990, Dop e Regtien 1996, Leonardiset *al.*, 1997, Jaklic *et al.*, 2000, Jaklic e Solina, 2003, (Chevalier *et al.*, 2003 e 2004, El-Khoury e Sahbani, 2008).

Basicamente, o algoritmo de Levenberg-Marquardt minimiza a função objetiva adotada, avaliando o vetor de parâmetros β_n , para $n=0, 1, 2, \dots, k$, sendo k o número de iterações do método, até que β_n seja o vetor ajustado que descreve a nuvem de pontos do objeto alvo.

O algoritmo de Levenberg-Marquardt construído em Matlab foi fornecido por Gavin (2011), com resultados satisfatórios para sistemas com uma e duas variáveis e o algoritmo tem as seguintes variáveis de entrada:

- β_i : vetor dos onze parâmetros, sendo i o índice de iteração.
- (x, y, z) : vetor de coordenadas dos pontos da nuvem.
- $F^{\frac{\epsilon_1}{2}}(\beta; X, Y, Z)$: A função objetivo.
- $d\beta$: incremento pelas derivadas numéricas.
- β_{\min} : limite mínimo.
- β_{\max} : limite máximo.

E como variável estimada de saída tem-se:

- β_{fit} : o vetor ajustado do modelo.

Com o vetor β_{fit} conseguiu-se reconstruir o objeto em 3D.

Os testes do método foram realizados para as seguintes categorias:

- 1) Objetos sintéticos: nuvens de pontos simuladas com Matlab®
- 2) Objetos reais: nuvens de pontos adquiridas pelo sistema Kinect®

A amostra para validação do método são os objetos sintéticos da Figura 21. Estes objetos pertencem à família de superelipsóide e foram construídos a partir de seus parâmetros característicos.

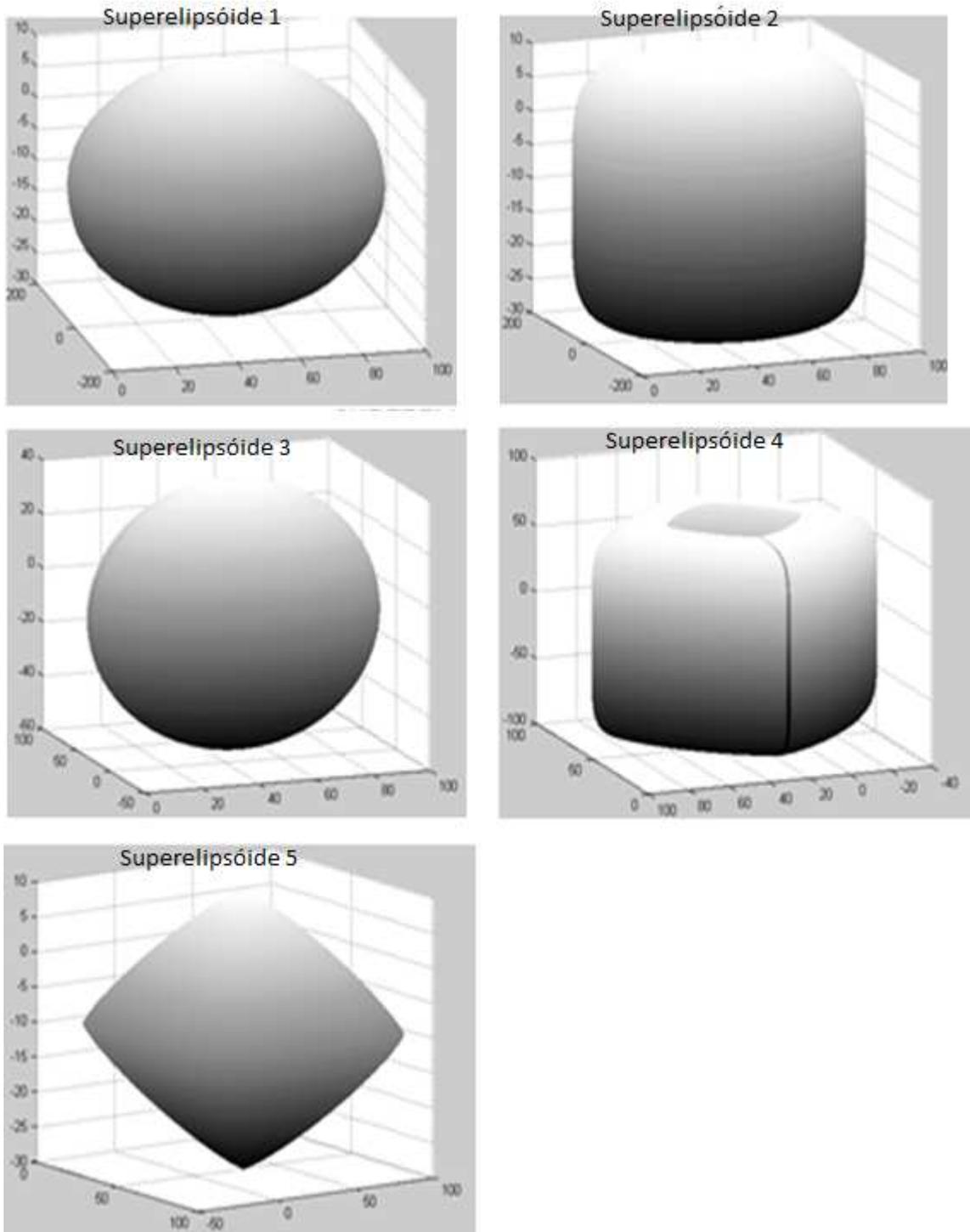


Figura 21. Objetos em 3D

Os parâmetros característicos para cada elipsóide foram escolhidos de tal modo que abrangem diferentes setores da família de superelipsóides, os valores escolhidos para o tamanho (a_1, a_2, a_3), a rotação (φ, θ e ϕ) e os de forma (ϵ_1 e ϵ_2) estão descritos na Tabela 2.

Tabela 2. Parâmetros característicos para gerar a nuvem de pontos do objeto alvo

Superelipsóide	a_1	a_2	a_3	φ	θ	ϕ	ϵ_1	ϵ_2
Superelipsóide 1	170	50	20	0	20	0	1	1
Superelipsóide 2	170	50	20	30	20	0	0.4	1
Superelipsóide 3	50	50	50	30	20	0	1	1
Superelipsóide 4	50	70	80	30	0	0	0.3	1.5
Superelipsóide 5	50	70	20	0	10	0	1.7	0.5

Estes parâmetros são variáveis da função paramétrica da superelipsóide (Equação 16), porém foi implementado o algoritmo em Matlab® tal que a saída fosse às coordenadas x, y, z da superfície do objeto sintético.

Com o fim de abranger o estudo experimental do projeto e avaliar o método construído em uma cena real foram feitos quatro objetos geométricos da família das superelipsóides (Figura 22)

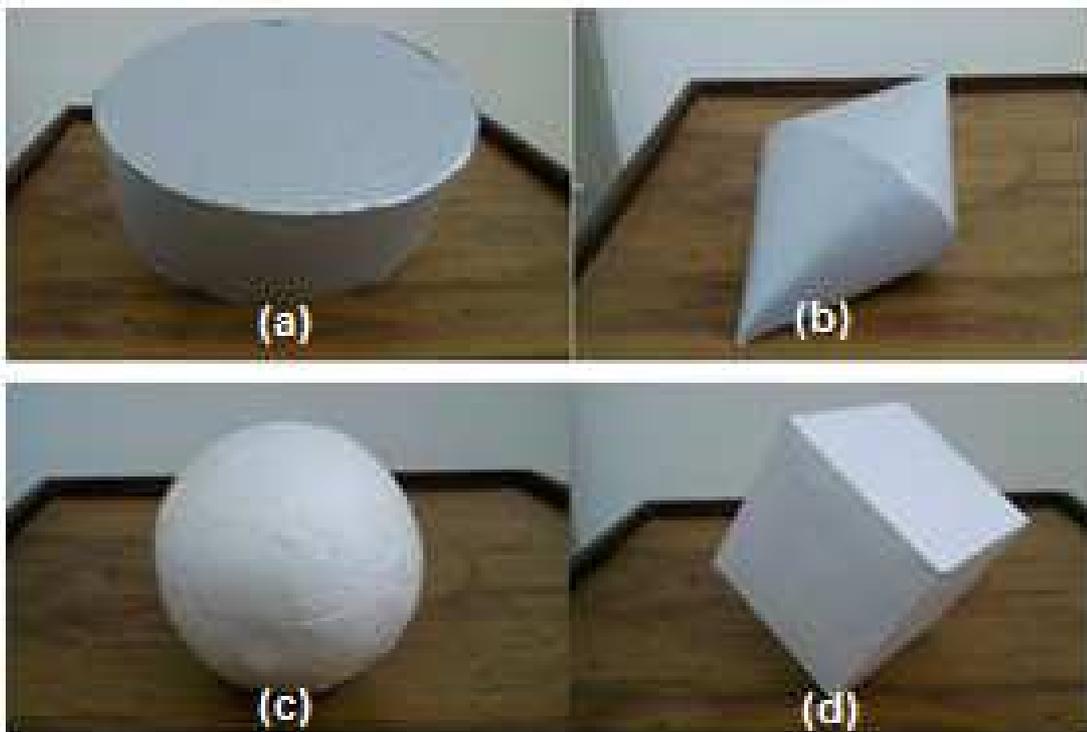


Figura 22. Objetos geométricos (a) cilindro, (b) octaedro (c) esfera (d) cubo.

O processo consistiu em obter as imagens de profundidade dos quatro objetos com o sistema *kinect*. O sistema capturou as coordenadas da cena completa, objeto alvo e fundo, porém, foi necessário o uso de softwares livre “*3d Converter*” e “*MeshLab*” para extrair só as coordenadas do objeto em formato “txt”.

3.5 ANÁLISE DE ERRO

A análise dos erros de ajuste com a metodologia foi realizada a partir dos resultados obtidos em suas etapas, com as nuvens de pontos sintéticas e reais.

Na etapa de validação do algoritmo foram utilizadas as nuvens de pontos sintéticas, de forma que os valores dos parâmetros reais do objeto são conhecidos e os erros são calculados comparando os valores estimados pelo algoritmo para cada parâmetro com os parâmetros reais conhecidos, através do erro médio calculado como:

$$e = |Vr_{\text{real}} - Vr_{\text{estimado}}| / Vr_{\text{real}} \quad (46)$$

sendo Vr_{real} o valor real do parâmetro e Vr_{estimado} o valor obtido pelo método de iteração, e e é o erro que indica a proximidade dos dados reais com os resultados obtidos. Deste modo, a metodologia implementada pode ser avaliada pelo resultado do erro de cada parâmetros do objeto.

A etapa do ajuste das nuvens de pontos permite a estimativa não só do erro dos parâmetros, mas também dos seus erros padrões assintóticos indicando a qualidade do ajuste. Desta forma, um bom ajuste de curva indica que o modelo proposto é adequado para explicar os resultados observados. Os erros padrões assintóticos dos parâmetros δ_p , definido por Gavin (2011) são definidos pela raiz quadrada da matriz diagonal da matriz de covariância $(J^T W J)^{-1}$ com a seguinte equação:

$$\delta_p = \sqrt{\text{diag}(J^T W J)^{-1}} \quad (47)$$

sendo W a matriz de ponderação igual a 1 e J a matriz Jacobiano.

4 VALIDAÇÃO E TESTES DO MÉTODO DE RECONSTRUÇÃO 3D

Este capítulo aborda o desenvolvimento utilizado nos testes para validação do modelo 3D gerado através do método de reconstrução 3D apresentado neste trabalho e aborda também, os resultados obtidos e a análise da qualidade e confiabilidade do método em relação a objetos reais.

4.1 Validação do método de reconstrução 3D com os objetos sintéticos

O desenvolvimento do método de reconstrução 3D do objeto com imagens de profundidade de objetos sintéticos foi feito com o intuito de validar cada uma das etapas do método.

O conjunto de coordenadas colocados em um sistema tridimensional é conhecido como uma nuvem de pontos e por meio da função “*plot3*” de Matlab® pode-se observar a nuvem para cada objeto(Figura 23). A criação da nuvem de pontos de um objeto pelo algoritmo permite ter as coordenadas de toda a superfície do objeto. No entanto, observa-se que para as superelipsóides 1, 3 e 5 a nuvem de pontos apresenta uma distribuição uniforme. Mas, para as superelipsóides 2 e 4, não são detectados os pontos na parte central da superfície, e apresentam concentração de pontos na parte superior e inferior.

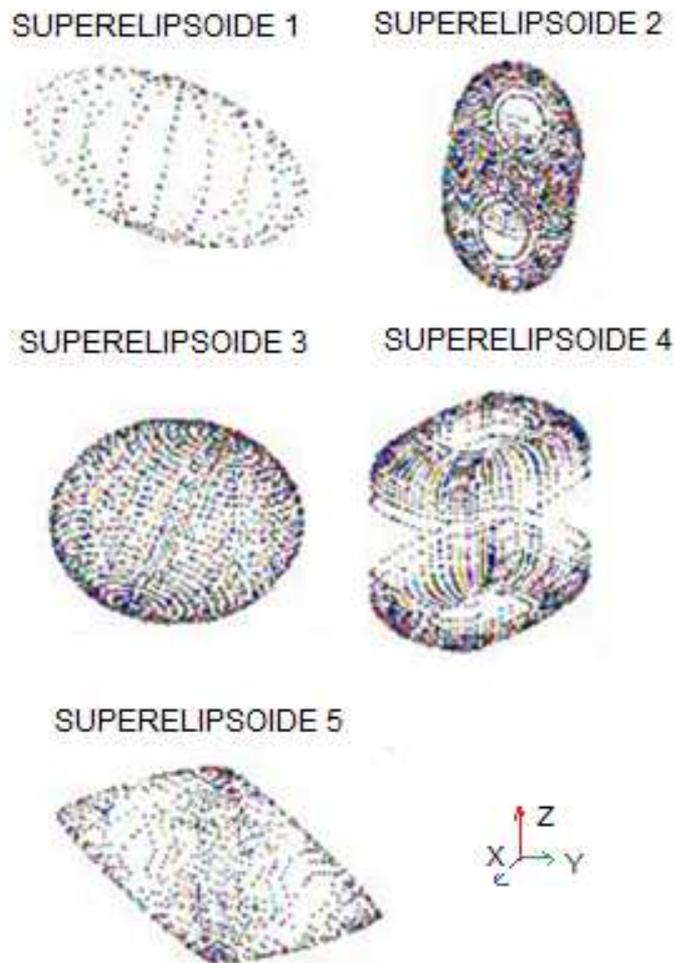


Figura 23. Nuvens de pontos para cada objeto da figura 21.

Uma vez obtida a amostra sintética, começa-se com o processo de validação do método. Este processo foi dividido por partes segundo as etapas de desenvolvimento do método, com o fim de verificar a resposta de cada etapa e garantir a entrada para a seguinte etapa. Dessa forma as partes do processo de validação consistem em comparações entre os resultados esperados e os resultados estimados, da seguinte maneira:

- Parte 1. Encontrar os parâmetros iniciais da nuvem de pontos (β inicial).
Validação: Comparar os parâmetros fornecidos para criar a nuvem de pontos com os parâmetros obtidos que correspondem ao vetor de parâmetros β inicial
- Parte 2. A definição da função objetivo

Validação: Comparar o resultado da função objetivo $F(\beta; x, y, z)$ com o valor de $F=1$. Isto indica se o ponto (x, y, z) esta na superfície do objeto.

- Parte 3. O Algoritmo de Levenberg-Marquardt.

Validação: Comparar o resultado estimado (β final) com o resultado esperado (β real).

4.1.1 Parte 1: Estimativa dos parâmetros iniciais da nuvem de pontos

A validação desta etapa consiste em verificar a resposta do algoritmo da estimativa dos parâmetros iniciais (item 3.2). Este algoritmo encontra a posição, o tamanho e os eixos de inércia da nuvem de pontos a fim de acelerar o processo de minimização e diminuir os erros entre os dados esperados e os dados da função de ajuste da superfície.

- A posição do objeto que basicamente é encontrar o centróide da nuvem de pontos foi verificada localizando a coordenada do centróide na nuvem de pontos. Observa-se, na Figura 24, a localização do centróide para cada uma das superelipsóides como a origem dos eixos coordenados da nuvem de pontos.

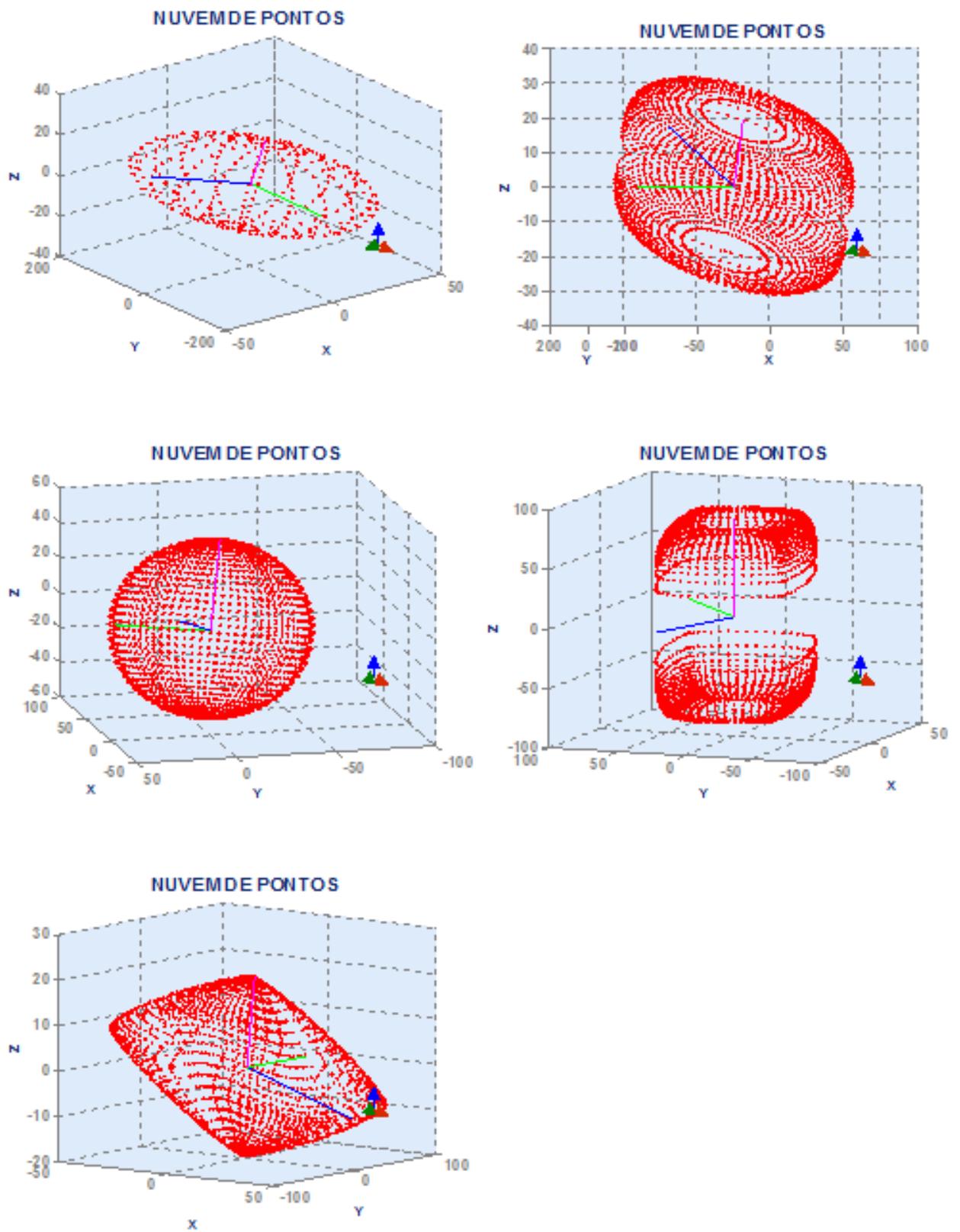


Figura 24. Representação dos eixos de inércia, posição e estimativa do tamanho das nuvens.

- Os eixos de inércia, também vistos na Figura 24 como o eixo x (azul), o eixo y (verde) e o eixo z (rosa) representam a rotação do sistema de coordenadas global ao sistema de coordenadas local. Os parâmetros que descrevem a rotação da nuvem de pontos são os ângulos de Euler. A amostra das nuvens de pontos foi fornecida com uma rotação dada, porém pode-se comparar o resultado entre os ângulos de Euler (φ, θ, ϕ) estimados e os ângulos (φ, θ, ϕ) originais. A comparação dos dados está na Tabela 3

Tabela 3. Valores reais e estimados dos ângulos de Euler.

Ângulos de Euler (rad)	Superelipsóide1		Superelipsode2		Supereipsoide3		Superelipsóide4		Superelipsóide5	
	Vr Real	Vr. Estimado	Vr Real	Vr. Estimado	Vr Real	Vr. Estimado	Vr Real	Vr. Estimado	Vr Real	Vr. Estimado
ϕ	0	0	0,5236	0,52356	0,5236	0,565196	0,5236	0,5223	0	0,001231
θ	0,3491	0,35	0,3491	0,349105	0,3491	0,34716	0	-2,4 E-0,5	0,1745	-0,17456
φ	0,00	0	0	1,02E-5	0	0,01422	0	6,10 E-8	0	0,0002

Pode-se observar que o valor estimado não diferiu muito do valor real. O erro médio da medição é 0,58%. Tem-se que a diferença do valor estimado e o valor real é zero para 12 das 15 leituras dos ângulos. Observa-se uma maior aproximação entre o valor real e o valor verdadeiro, porém a exatidão das medidas valida o algoritmo implementado.

- O tamanho (a_1, a_2 e a_3) da nuvem de pontos é a medida do comprimento dos eixos de inércia. Na Figura 24, pode-se ver que os três eixos têm um comprimento, este valor é fornecido pelo cálculo do ponto máximo da nuvem que se encontra sob o eixo. Os valores estimados são os apresentados na Tabela 4.

Tabela 4. Dados do vetor β com o algoritmo 2 para cada nuvem de pontos.

Parâmetros de tamanho	Superelipsóide1		Superelipsode2		Supereipsoide3		Superelipsóide4		Superelipsóide5	
	Real	Estimado	Real	Estimado	Real	Estimado	Real	Estimado	Real	Estimado
a_1	50	48,56	50	49,94	50	49,95	50	49,94	50	49,95
a_2	170	170	170	169,81	50	49,88	70	69,90	70	69,86
a_3	20	20	20	20,49	50	51,13	80	82,017	20	20,40

Pode-se observar a exatidão das medições por que o valor estimado é muito próximo do valor real. O erro médio da medição é de 0.8% o que significa a aprovação do método para obter a

rotação do objeto. No entanto, o algoritmo de minimização de Levenberg-Marquardt precisa da informação de um intervalo onde seja possível encontrar os parâmetros do vetor característico do objeto. Pelo que se usou o valor do erro médio como referência para encontrar o valor máximo e mínimo dos parâmetros utilizando a relação entre o valor real (Tr) e o valor obtido (To). O intervalo foi definido a partir das seguintes condições:

$$\text{Para } Tr = To \text{ então } \frac{Tr}{To} = 1 \quad (48)$$

$$\text{Para } Tr \leq To \text{ então } \frac{Tr}{To} \leq 1 - \delta \quad (49)$$

$$\text{Para } Tr \geq To \text{ então } \frac{Tr}{To} \geq 1 + \delta \quad (50)$$

Sendo $\pm\delta$ o grau de tolerância para os valores máximos e mínimos da medida. Para encontrar o grau de tolerância, se calculou a relação entre os valores reais e estimados (Tabela 5). Logo, o grau médio de tolerância neste experimento foi de 0,0033. Mas, o resultado do erro médio foi entre $[0, 0,008]$, e com o fim de abranger mais valores dentro do intervalo o grau de tolerância foi escolhido com um valor de 0,05. Assim o intervalo para os parâmetros de tamanho, posição, rotação é:

$$To(1 - \delta) \leq To \leq To(1 + \delta) \quad (51)$$

$$0,95To \leq To \leq 1,05To \quad (52)$$

Tabela 5 .Variação dos parâmetros entre os valores reais e estimados

Superelipsóides	Tamanho (a1, a2, a3)		$\frac{Tr}{To}$
	Tr	To	
Superelipsóide1	50	48,56	0,9712
	170	170	1
	20	20	1
Superelipsóide2	50	49,94027	0,998805
	170	170	1
	20	20,49621	1,024811
Superelipsóide3	50	49,95832	0,999166
	50	49,88584	0,997717
	50	51,13116	1,022623
Superelipsóide4	50	49,94822	0,998964
	70	69,90906	0,998701
	80	82,01767	1,025221
Superelipsóide5	50	49,95043	0,999009
	70	69,85926	0,997989
	20	20,40591	1,020296

A posição do objeto calculada pelo centróide da nuvem de pontos, também requer o grau de tolerância para obter os valores máximos e mínimos. Porém, o intervalo de medida da posição foi:

$$(Xc, Yc, Zc)(1 - \delta) \leq Xc, Yc, Zc \leq (Xc, Yc, Zc)(1 + \delta) \quad (53)$$

$$0,95Xc \leq Xc \leq 1,05Xc \quad (54)$$

$$0,95Yc \leq Yc \leq 1,05Yc \quad (55)$$

$$0,95Zc \leq Zc \leq 1,05Zc \quad (56)$$

Finalmente, o vetor β compõe os parâmetros de posição, tamanho, rotação e forma, da nuvem de pontos do objeto, e os vetores β_{\min} , β_{\max} , correspondem ao vetor de parâmetros mínimos e máximos respectivamente. Os resultados coerentes desta parte do algoritmo foram validados.

4.1.2 Parte 2. Definição da função objetivo

O algoritmo para estimar o termo da função objetivo correspondente ao encontro dos valores da função da superelipsóide (descrita na equação 16) foi validado com as nuvens de pontos da superelipsóide 1 e 3 (a elipsóide e a esfera, respectivamente). O processo descrito no algoritmo é observado nos gráficos da Figura 25. A primeira Figura (25(a)) é a execução da determinação do centróide na nuvem de pontos. Em seguida, fez-se a transformação e a rotação da nuvem de pontos, como demonstrado na Figura 25(b). As nuvens de pontos foram postas no sistema de coordenadas local.

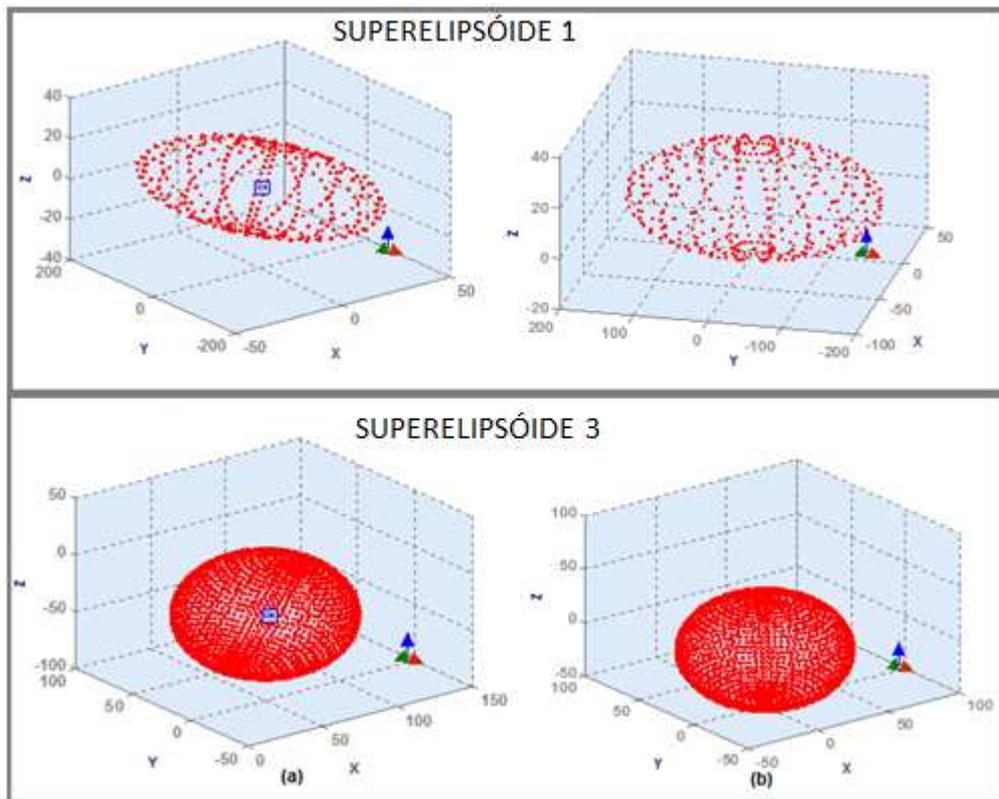


Figura 25. Nuvens de pontos para as Superelipsóides 1 e 3. a) Nuvens de pontos de entrada e b) Nuvens de pontos no sistema de coordenadas local feito pelo algoritmo 3.

O algoritmo estimou o valor da função objetivo e os resultados foram vistos em um gráfico da função $F(\beta; x, y, z)$ contra o ponto $P_i(x_i, y_i, z_i)$, como mostrado na Figura 26. As análises do gráfico (Figura 26) foram feitas tendo-se em conta a definição de que quando $F(\beta; x_i, y_i, z_i) = 1$, então significa que o ponto $P_i(x_i, y_i, z_i)$ está na superfície.

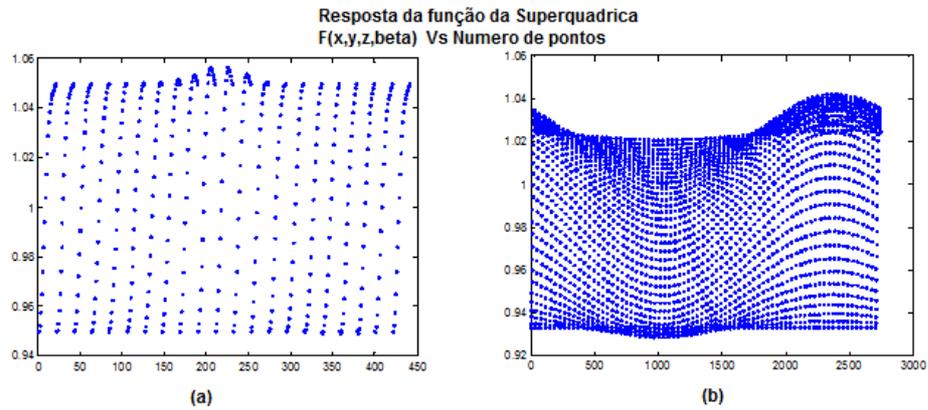


Figura 26. Resposta da função F com respeito a o numero de pontos (N). a) Superelipsóide 1 b) Superelipsóide 3.

Tem-se que a média de F para a Figura 25(a) é 1, e para a Figura 26(b) é 0,98. O erro médio para cada uma das médias é 0,03 e 0,028. Esses valores fornecem confiabilidade na resposta do algoritmo, uma vez que os valores estimados são muito próximos ao valor real da função. Assim, os resultados do algoritmo para encontrar a função objetivo são validados.

4.1.3 Parte 3: O Algoritmo de Levenberg-Marquardt

O último processo do método de reconstrução 3D é a minimização da função objetivo por meio do algoritmo de Levenberg-Marquardt. Como resultado desse processo, tem-se o vetor de parâmetros β com as características estimadas do objeto em ensaio.

Os resultados e os erros padrões assintóticos dos parâmetros estimados para cada nuvem de pontos estão descritos nas Tabelas 6, 7, 8, 9 e 10. O padrão assintótico de cada parâmetro é determinado segundo Gavin (2011) (equação 47) que mediante esta medida apresenta quanto a variabilidade dos dados propaga a variabilidade dos parâmetros e é essencialmente uma medida de erro para os parâmetros.

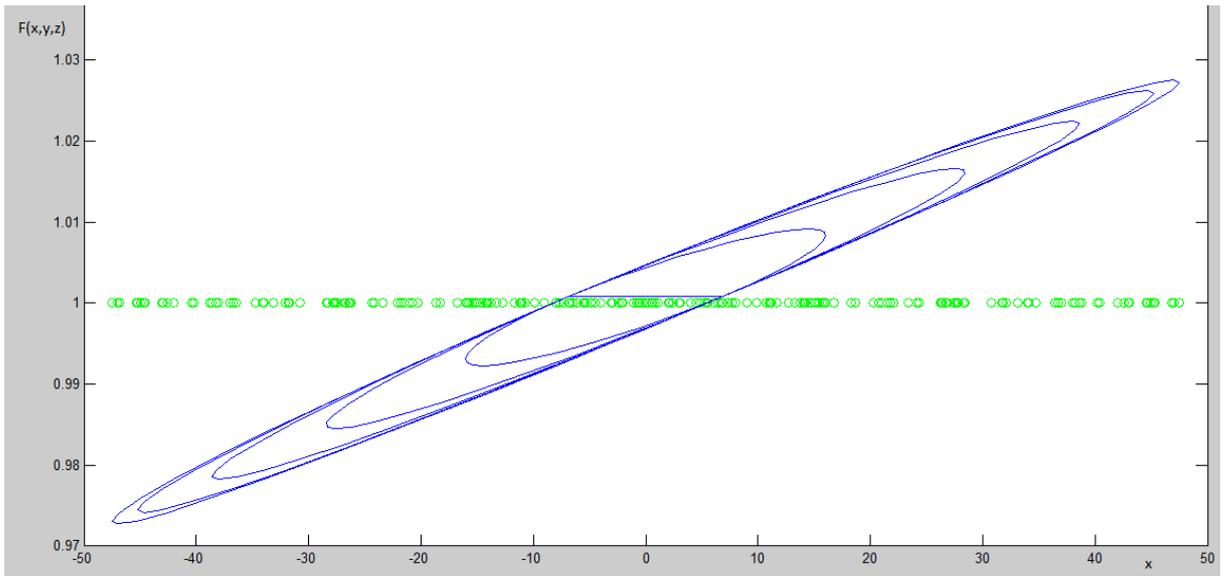
Uma vez obtido o vetor de parâmetros estimados β_{final} se fez reconstrução 3D descrita nas Figuras 27, 28, 29, 30 e 31.

- **Superelipsóide1**

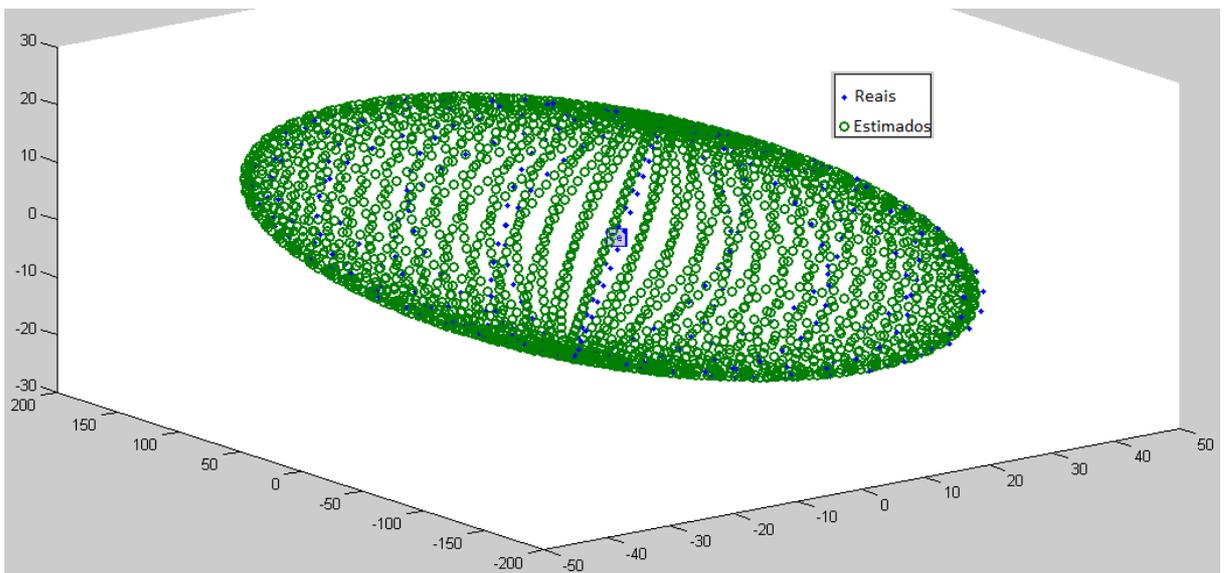
A superelipsóide 1 com 441 pontos conseguiu o resultado depois de 51 iterações em um tempo de 3 segundos. Na seguinte tabela se observam os resultados do vetor de parâmetros real com o vetor de parâmetros estimado. Além do erro padrão assintótico para cada parâmetro. Depois, nas figuras é observado o resultado desenhando em 3D os pontos das coordenadas com o vetor real e o vetor estimado.

Tabela 6. Resultados e erro padrão assintótico para a superelipsóide1

Parâmetros	Valor Real	Valor estimado	Erro padrão assintótico
a1	50	49,80	0,394
a2	170	169,95	1,469
a3	20	20,05	0,087
e1	1	1,00	0,017
e2	1	0,98	0,028
ϕ (rad)	0	0,00	0,003
θ (rad)	0,3490	0,35	0,004
φ (rad)	0	0,00	0,001
Xc	-	-1,29	0,192
Yc	-	0,00	0,722
Zc	-	0,52	0,087



(a)



(b)

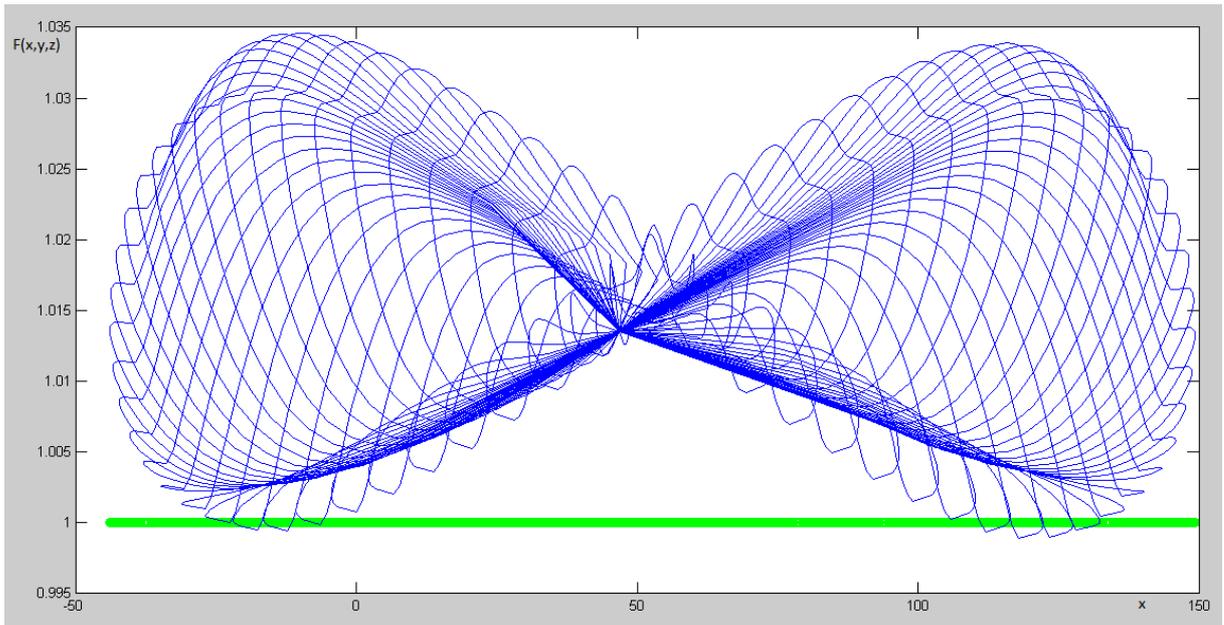
Figura 27. Resultados da superelipsóide 1 a). Gráfico de uma iteração do método iterativo Levenberg-Marquardt, comparando a aproximação da Função F estimada (Azul) com a real (verde). Gráfico de $F(x,y,z)$ Vs o valor da coordenada X b) Gráfico em 3D dos pontos x, y, z reais com a cor azul e os pontos estimados com a cor verde.

- **Superelipsóide2**

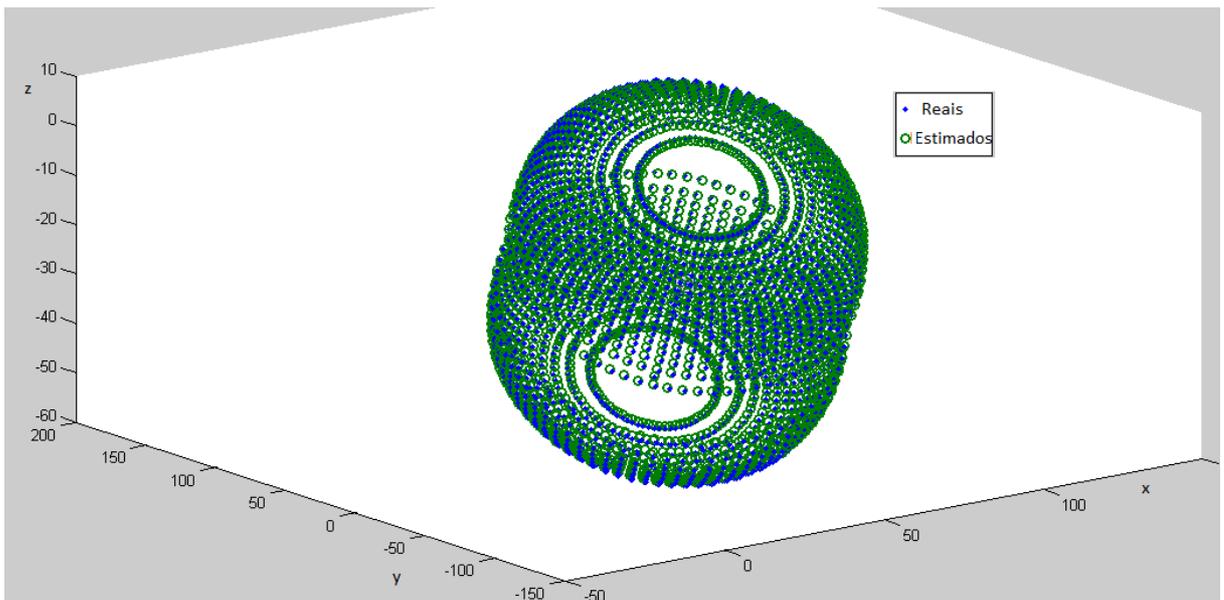
A superelipsóide 2 com 2738 pontos conseguiu o resultado depois de 197 iterações em um tempo de 18 segundos. Na seguinte tabela se observam os resultados do vetor de parâmetros real com o vetor de parâmetros estimado. Além do erro padrão assintótico para cada parâmetro. Depois, nas figuras é observado o resultado desenhando em 3D os pontos das coordenadas com o vetor real e o vetor estimado.

Tabela 7. Resultados e erro padrão assintótico para a superelipsóide2

Parâmetros	Valor Real	Valor Estimado	Erro padrão assintótico
	a1	50	49,999497
a2	170	169,99921	0,0004
a3	20	19,999893	2,47E-05
e1	0.4	0,4000297	4,47E-06
e2	1	1,0000014	7,27E-06
ϕ (rad)	0,5236	5,24E-01	8,33E-07
θ (rad)	0,3490	0,3490655	7,91E-07
φ (rad)	0	9,73E-06	3,54E-07
Xc	-	54,337888	0,0001
Yc	-	4,4236874	0,0002
Zc	-	25,900261	2,51E-05



(a)



(b)

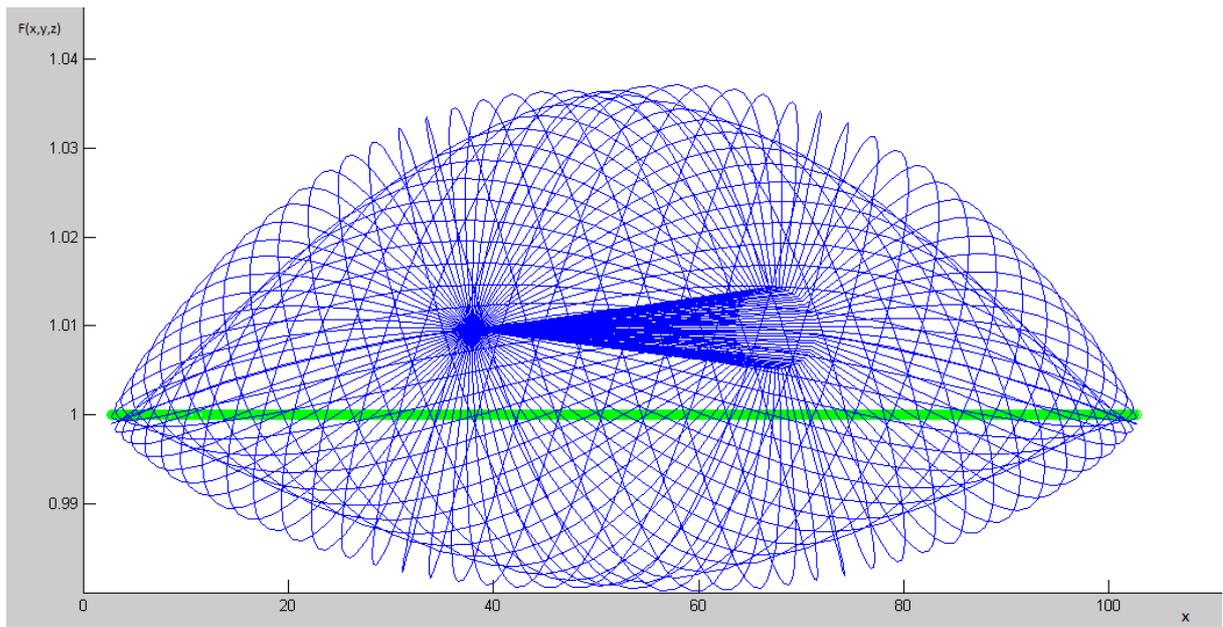
Figura 28. Resultados da superelipsóide 2 a). Gráfico de uma iteração do método iterativo Levenberg-Marquardt, comparando a aproximação da Função F estimada (Azul) com a real (verde). Gráfico de $F(x,y,z)$ Vs o valor da coordenada X b) Gráfico em 3D dos pontos x, y, z reais com a cor azul e os pontos estimados com a cor verde.

- **Superelipsóide 3**

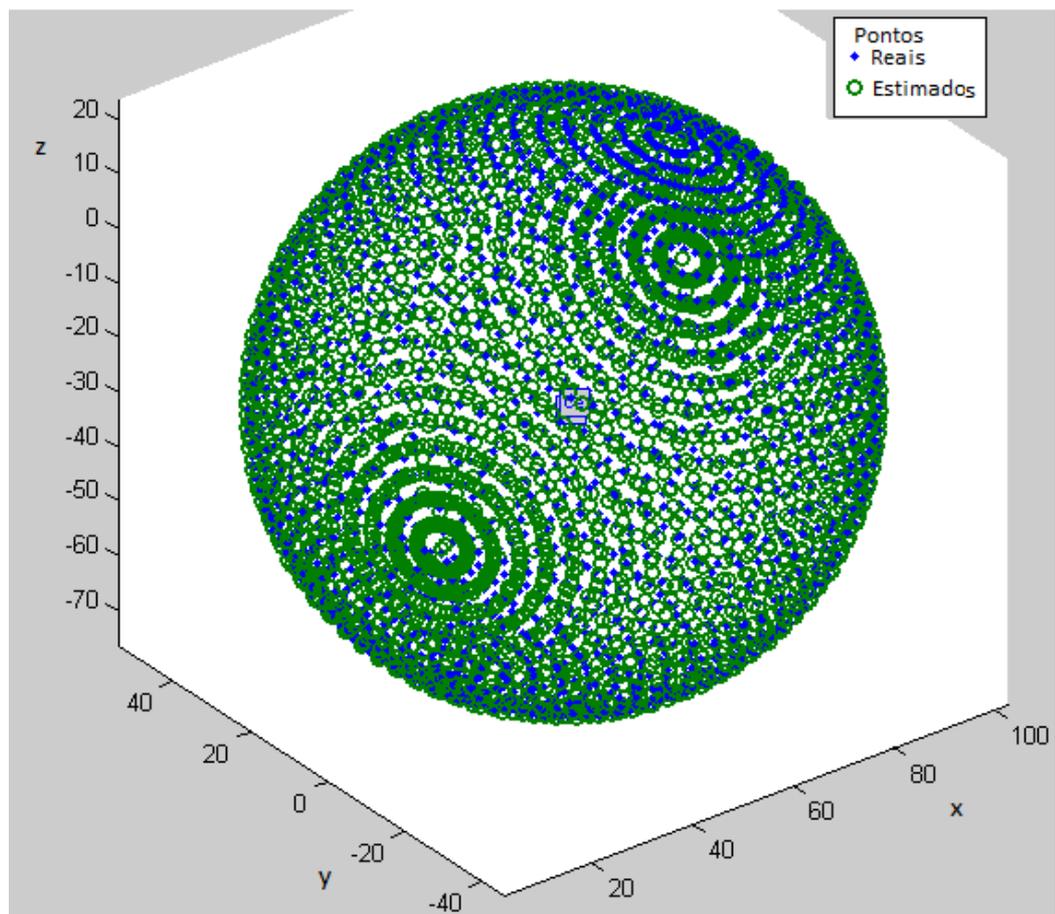
A superelipsóide 2 com 2738 pontos conseguiu o resultado depois de 51 iterações em um tempo de 3 segundos. Na seguinte tabela se observam os resultados do vetor de parâmetros real com o vetor de parâmetros estimado. Além do erro padrão assintótico para cada parâmetro. Depois, nas figuras é observado o resultado desenhando em 3D os pontos das coordenadas com o vetor real e o vetor estimado.

Tabela 8. . Resultados e erro padrão assintótico para a superelipsóide3

Parâmetros	Valor Real	Valor estimado	Erro padrão assintótico
a1	50	49,99	0,021
a2	50	50,02	0,047
a3	50	49,98	0,012
e1	1	0,999	0,001
e2	1	1,000	0,001
ϕ (rad)	0,524	0,537	0,478
θ (rad)	0,349	0,366	0,304
φ (rad)	0	0,015	0,127
Xc	-	54,442	0,174
Yc	-	4,394	0,865
Zc	-	-26,320	0,402



(a)



(b)

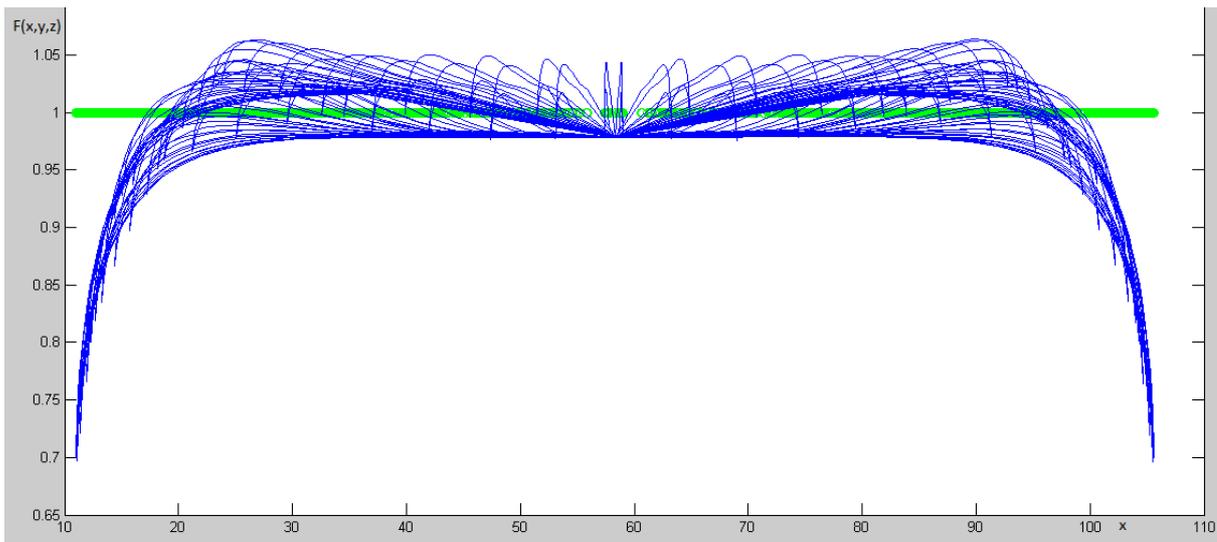
Figura 29. Resultados da superelipsóide 3. a). Gráfico de uma iteração do método iterativo Levenberg-Marquardt, comparando a aproximação da Função F estimada (Azul) com a real (verde). Gráfico de $F(x,y,z)$ Vs o valor da coordenada X b) Gráfico em 3D dos pontos x, y, z reais com a cor azul e os pontos estimados com a cor verde.

- **Superelipsóide 4**

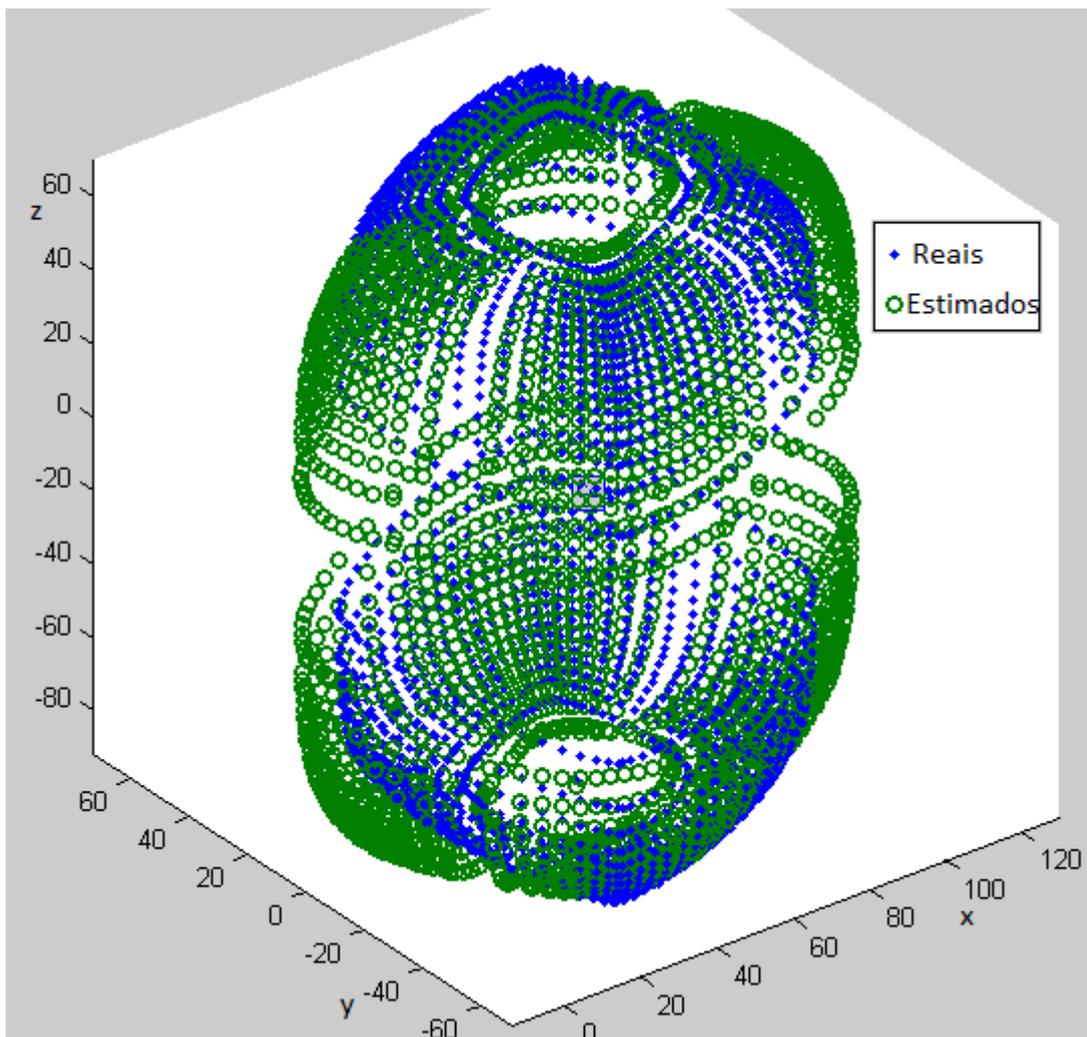
A superelipsóide 4 com 2738 pontos conseguiu o resultado depois de 51 iterações em um tempo de 8 segundos. Na seguinte tabela se observam os resultados do vetor de parâmetros real com o vetor de parâmetros estimado. Além do erro padrão assintótico para cada parâmetro. Depois, nas figuras é observado o resultado desenhando em 3D os pontos das coordenadas com o vetor real e o vetor estimado.

Tabela 9. Resultados e erro padrão assintótico para a superelipsóide 4

Parâmetros	Valor real	Valor estimado	Erro padrão assintótico
a1	50	49,98	0,079
a2	70	69,97	0,112
a3	80	80,04	0,071
e1	0.3	0,30	0,003
e2	1.5	1,50	0,005
ϕ (rad)	0,5236	0,52	0,002
θ (rad)	0	0,00	0,001
φ (rad)	0	0,00	0,001
Xc	-	59,61	0,042
Yc	-	1,05	0,049
Zc	-	-11,49	0,048



(a)



(b)

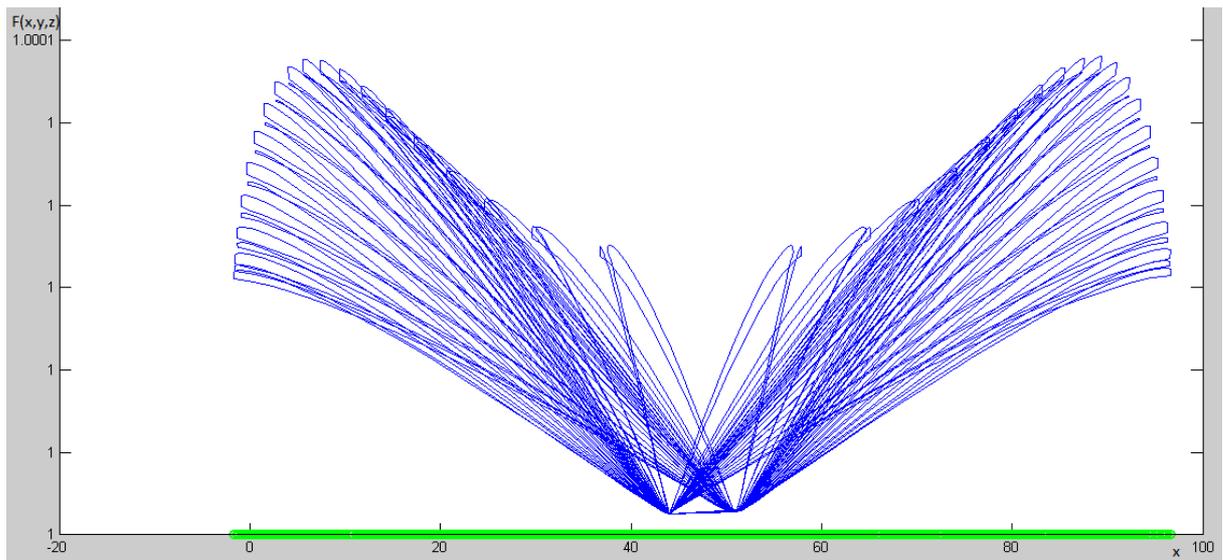
Figura 30. Resultados da superelipsóide 4 a). Gráfico de uma iteração do método iterativo Levenberg-Marquardt, comparando a aproximação da Função F estimada (Azul) com a real (verde). Gráfico de $F(x,y,z)$ Vs o valor da coordenada X b) Gráfico em 3D dos pontos x, y, z reais com a cor azul e os pontos estimados com a cor verde.

- Superelipsóide 5

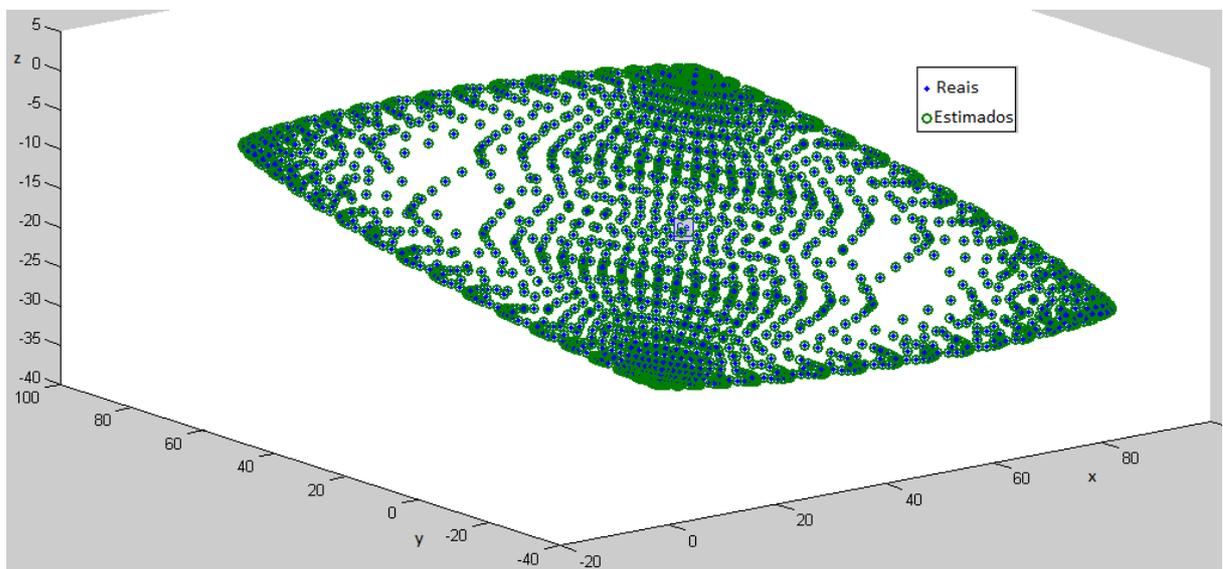
A superelipsóide 5 com 2738 pontos conseguiu o resultado depois de 63 iterações em um tempo de 9 segundos. Na seguinte tabela se observam os resultados do vetor de parâmetros real com o vetor de parâmetros estimado. Além do erro padrão assintótico para cada parâmetro. Depois, nas figuras é observado o resultado desenhando em 3D os pontos das coordenadas com o vetor real e o vetor estimado.

Tabela 10. Resultados e erro padrão assintótico para a superelipsóide5

Parâmetros	Valor real	Valor estimado	Erro padrão assintótico
a1	50	50,01	0,054
a2	70	69,97	0,076
a3	80	79,99	0,012
e1	1.7	1,70	0,002
e2	0.5	0,50	0,003
ϕ (rad)	0,52	0,00	0,001
θ (rad)	0	0,18	0,000
φ (rad)	0	0,00	0,000
Xc	-	49,14	0,023
Yc	-	31,00	0,032
Zc	-	-18,09	0,007



(a)



(b)

Figura 31. Resultados da superelipsóide 5 a). Gráfico de uma iteração do método iterativo Levenberg-Marquardt, comparando a aproximação da Função F estimada (Azul) com a real (verde). Gráfico de $F(x,y,z)$ Vs o valor da coordenada X b) Gráfico em 3D dos pontos x, y, z reais com a cor azul e os pontos estimados com a cor verde.

O tempo de execução para esta etapa esteve entre os 3 e 8 segundos. Os valores finais estimados para cada parâmetro são próximos ao valor real do objeto. O valor do erro padrão assintótico na maioria dos casos foi menor que 0,1. Gráficamente, para as cinco superelipsóides, os valores da função foram ajustados a cada iteração do algoritmo até chegar a seu valor esperado, próximo de 1. Nas Figuras (b), as coordenadas originais do objeto (cor azul) estão quase sobrepostas com as coordenadas finais (cor verde). Porém, obteve-se uma

máxima aproximação entre o vetor de parâmetros reais do objeto original e os valores estimados dos parâmetros característicos do objeto em estudo. Assim, os resultados obtidos com o método de reconstrução 3D implementado são validados. Desse modo, o método proposto garantiu a proximidade entre o objeto original e o objeto modelado e os resultados foram validados para se executar a segunda parte experimental do trabalho.

4.2 RECONSTRUÇÃO 3D COM OS OBJETOS REAIS

As nuvens de pontos dos quatro objetos são mostradas na (Figura 32). Nesta Figura observou-se, com mais detalhe, que as superfícies do objeto não são perfeitamente polidas porque apresentam pequenas ondulações.

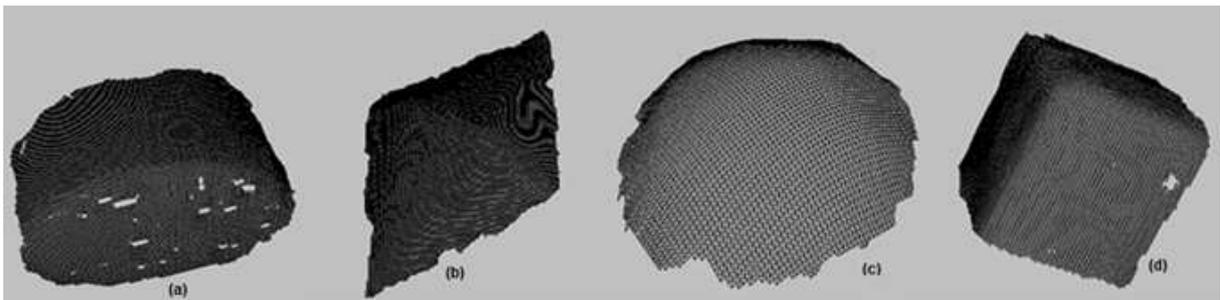


Figura 32. Nuvens de pontos dos objetos . (a) cilindro, (b) octaedro, (c) esfera e (d) cubo.

O método de reconstrução 3D foi executado até gerar o vetor de parâmetros β_{final} . Nas Tabelas 11, 12, 13 e 14 se descrevem o vetor de parâmetros final β_{final} , e o erro padrão assintótico . Nas Figuras 33, 34, 35 e 36 se observa (a) o gráfico das coordenadas (x,y,z) estimadas sobrepostas à nuvem de pontos adquiridas e (b) a reconstrução 3D do objeto com a função de superelipsóides.

4.2.1 Superelipsóide1: Cilindro

Numero de pontos: 10736

Numero de iterações para o resultado: 486

Tempo de execução: 3 min

Tabela 11. Resultados para a superelipsóide 1

Parâmetros	Valores estimados	Erro padrão assintótico
a1	0.0697	0.0009
a2	0.0737	0.0009
a3	0.0299	0.0001
ϵ_1	0.1000	0.0330
ϵ_2	1.2077	0.0306
$\varphi(\text{rad})$	3.1416	0.0070
$\theta(\text{rad})$	0.6350	0.0013
$\phi(\text{rad})$	1.5708	0.0292
Xc	0.0612	0.0008
Yc	-0.0758	0.0002
Zc	-0.7075	0.0006

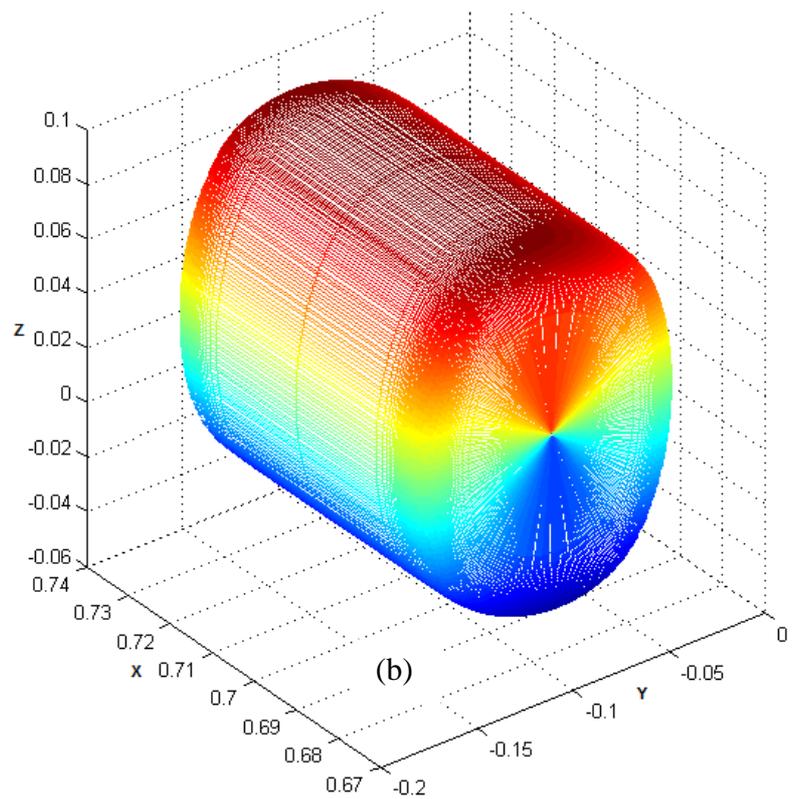
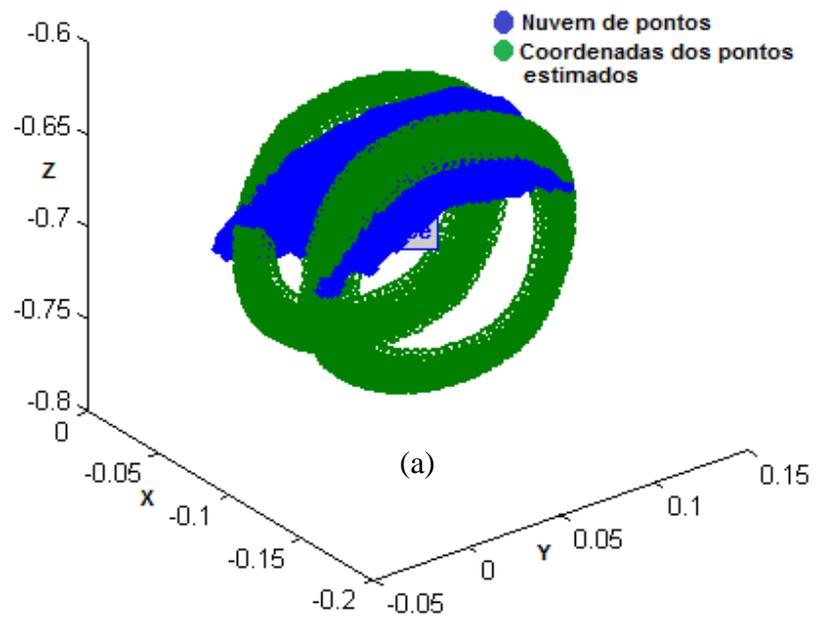


Figura 33. Resultados da superelipsóide em forma de cilindro (a) Coordenadas dos pontos ajustados (verdes) sobrepostos à nuvem de pontos (azuis) e (c), reconstrução da nuvem de pontos com o vetor de parâmetros ajustado β .

4.2.2 Superelipsóide2: Octaedro

Numero de pontos: 14302

Numero de iterações para o resultado: 114

Tempo de execução: 58 seg.

Tabela 12. Resultado para superelipsóide 2

Parâmetros	Valores estimados	Erro padrão assintótico
a1	0.0769	0.0007
a2	0.0517	0.0003
a3	0.0359	0.0002
ϵ_1	1.6932	0.0199
ϵ_2	0.6685	0.0625
φ(rad)	-0.5035	0.0057
θ (rad)	0.1069	0.0031
ϕ (rad)	0	0.0032
Xc	-0.0000	0.0002
Yc	-0.0000	0.0001
Zc	-0.0181	0.0001

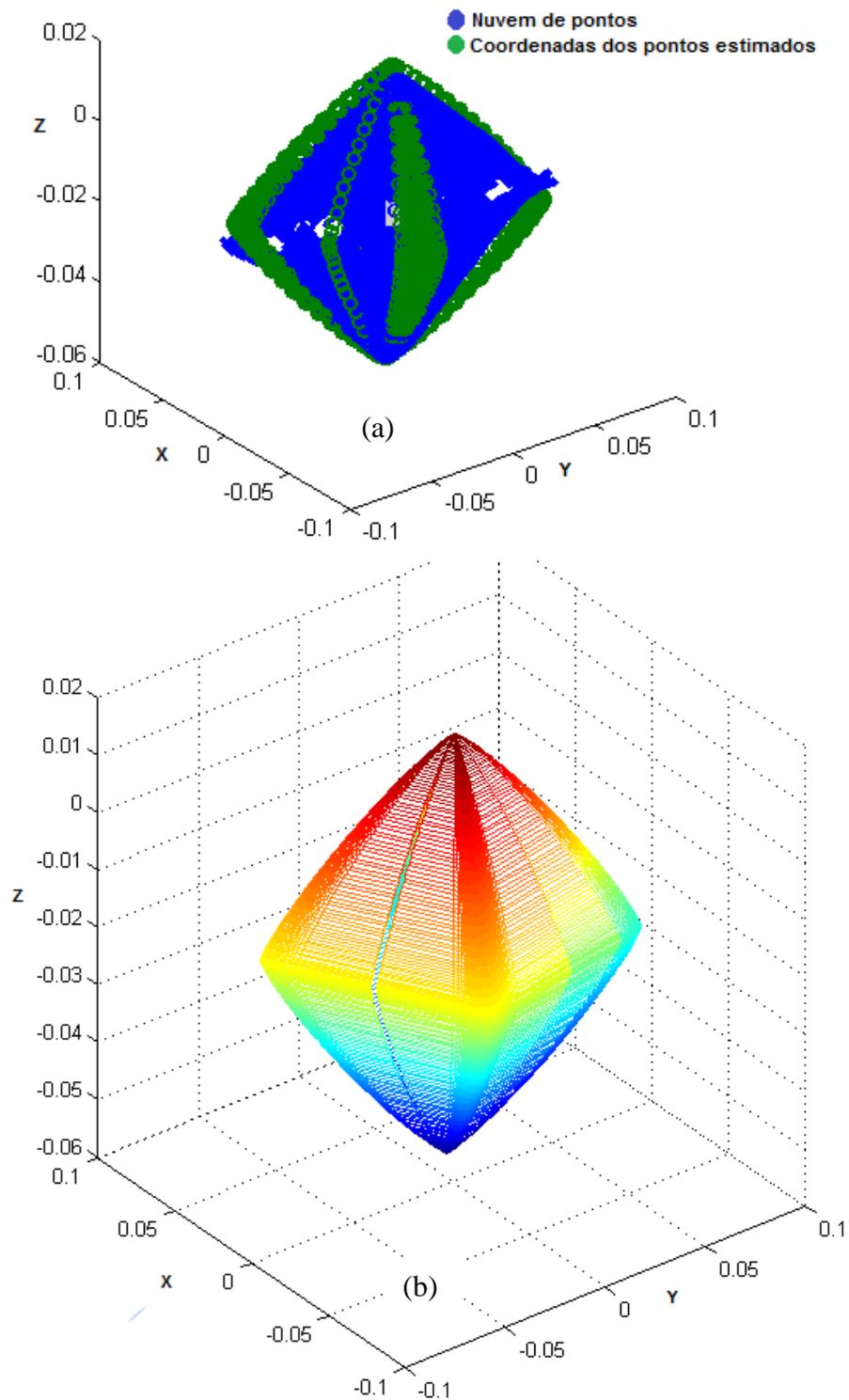


Figura 34. Resultados da superelipsóide em forma de octaedro (a) Coordenadas dos pontos ajustados (verdes) sobrepostos à nuvem de pontos (azuis) e (c), reconstrução da nuvem de pontos com o vetor de parâmetros ajustado β .

4.2.3 Superelipsóide3: Esfera

Numero de pontos: 6840

Numero de iterações para o resultado: 510

Tempo de execução: 62 seg.

Tabela 13. Resultado para superelipsóide 3

Parâmetros	Valores estimados	Erro padrão assintótico
a1	0.0370	0.0001
a2	0.0366	0.0000
a3	0.0417	0.0002
ϵ_1	1.1080	0.0099
ϵ_2	0.9230	0.0077
$\varphi(\text{rad})$	1.0472	0.0202
$\theta(\text{rad})$	0.1131	0.0099
$\phi(\text{rad})$	1.5708	0.0109
Xc	-0.0000	0.0001
Yc	-0.0000	0.0000
Zc	-0.0250	0.0000

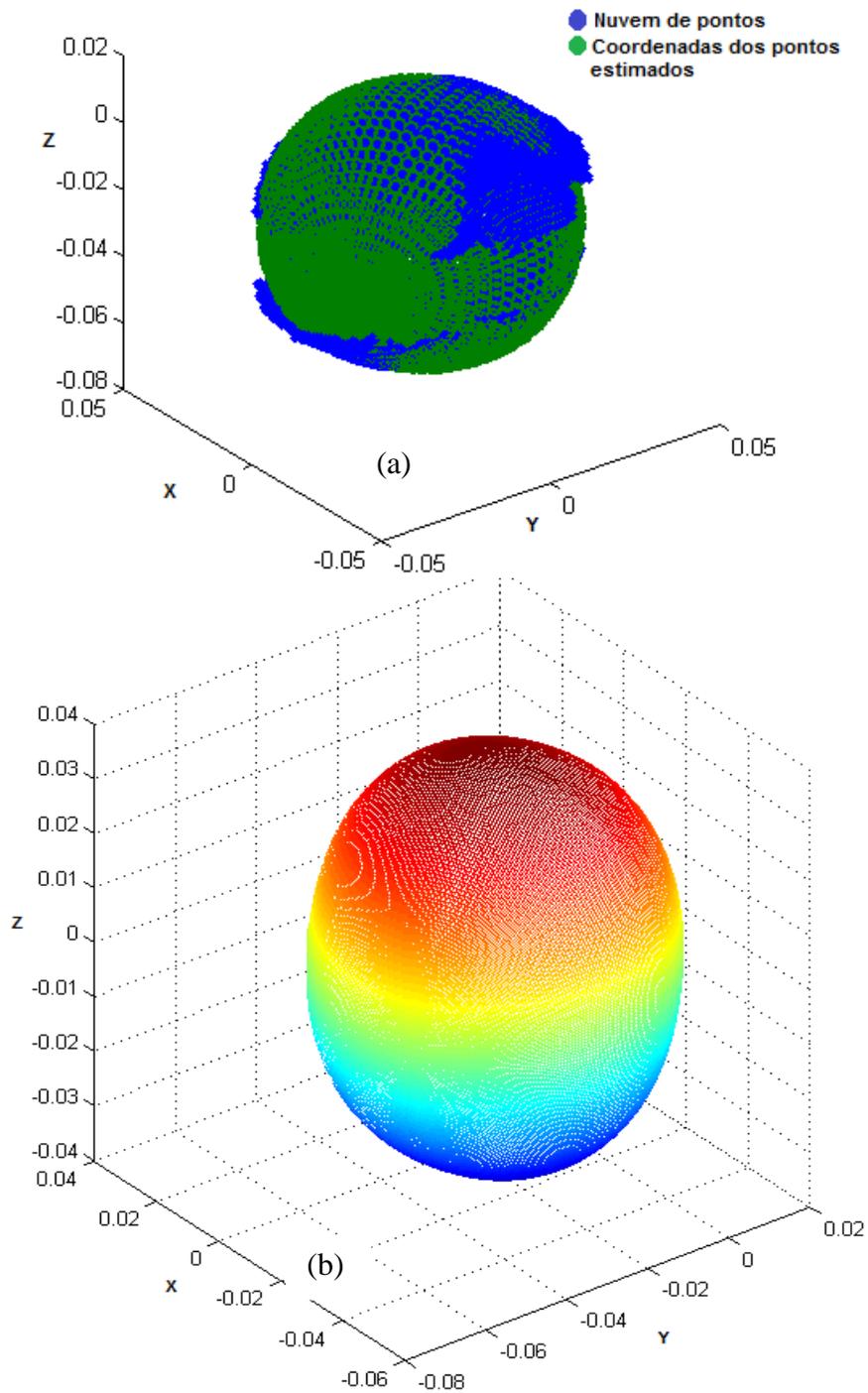


Figura 35. Resultados da superelipsóide em forma de esfera (a) Coordenadas dos pontos ajustados (verdes) sobrepostos à nuvem de pontos (azuis) e (c), reconstrução da nuvem de pontos com o vetor de parâmetros ajustado β .

4.2.4 Superelipsóide4: Cubo

Numero de pontos: 12518

Numero de iterações para o resultado: 126

Tempo de execução: 50 seg.

Tabela 14. Resultado para superelipsóide 4

Parâmetros	Valores estimados	Erro padrão assintótico
a1	0.0555	0.0004
a2	0.0569	0.0004
a3	0.0574	0.0003
ϵ_1	0.1142	0.0450
ϵ_2	1.9191	0.0276
$\varphi(\text{rad})$	1.0472	0.0060
$\theta(\text{rad})$	0.0166	0.0046
$\phi(\text{rad})$	1.5708	0.0063
Xc	0.0000	0.0002
Yc	0.0000	0.0002
Zc	-0.0242	0.0001

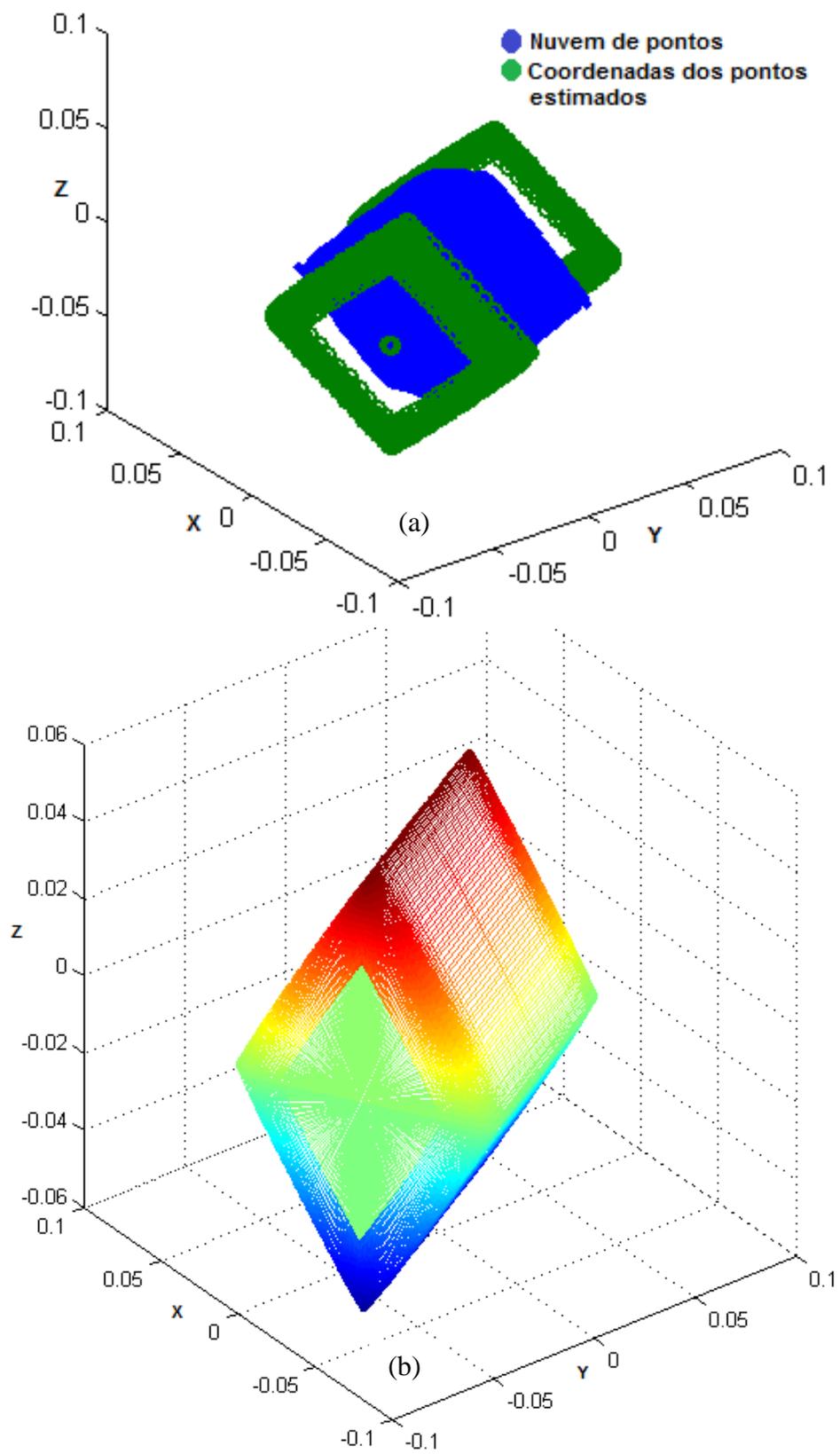


Figura 36. Resultados da elipsóide 1 em forma de cubo(a) Coordenadas dos pontos ajustados (verdes) sobrepostos à nuvem de pontos (azuis) e (c), reconstrução da nuvem de pontos com o vetor de parâmetros ajustado β .

4.2.5 Análise dos resultados

Nas subseções anteriores, foram descritos resultados tanto numéricos como gráficos. O valor de erro padrão assintótico permitiu avaliar a propagação das variabilidades em cada dado do vetor de parâmetros final. Os valores tiveram um porcentagem de erro padrão assintótico de 0,8%. Por isso, nas gráficas se observou uma aproximação entre o objeto modelado e a nuvem de pontos do objeto.

Cada Figura corresponde a um objeto reconstruído. As Figuras 33(a), 34(a), 35(a) e 36(a) trataram da representação dos pontos obtidos (verdes) sobrepostos às coordenadas da nuvem de pontos (azuis). Nestas Figuras, observou-se que os pontos verdes não correspondem em sua totalidade aos pontos azuis, isso causa dos erros aleatórios do sistema. A avaliação do vetor de parâmetros obtido na equação da superelipsóide (equação 15) gerou a representação tridimensional do objeto, exposto nas Figuras 33(b), 34(b), 35(b) e 36(b). O que significa, a avaliação do método implementado para reconstruir objetos simétricos que pertençam à família de superelipsóide.

5 CONCLUSÕES

Com base em modelos matemáticos de superquádricas, este trabalho implementou o método para reconstrução 3D de objetos simétricos a partir da nuvem de pontos utilizando o modelo matemático baseado em superquádricas para a elaboração de um objeto modelo muito próximo ao objeto real.

A metodologia foi validada tanto em relação a cada etapa do processo como na geração do modelo 3D, através de testes que utilizaram objetos sintéticos gerados por *software* e objetos reais com o sistema *Kinect*.

Os resultados dos testes com as imagens sintéticas foram observados graficamente com a aproximação dos dados estimados aos dados reais e se conseguiu avaliar o método implementado estimando os erros médios de cada parâmetro, que se mostraram inferiores a 1%

Graficamente, pode-se mostrar que o objeto 3D reconstruído é bem próximo das características geométricas reais do objeto-teste e, com os resultados obtidos dos valores do vetor de parâmetros, pôde-se comprovar que o método implementado neste trabalho atingiu o objetivo de gerar um objeto 3D a partir da nuvem de pontos utilizando um modelo geométrico de superquádricas.

Diante disso, pode-se concluir que o método implementado neste trabalho permite a modelagem 3D de objetos simétricos e para trabalhos futuros pode se testar seu funcionamento para visão robótica.

Por fim, os resultados desta pesquisa demonstraram não só a eficácia do método com nuvens de pontos simuladas, mas também sua adequação à nuvem de pontos de objetos reais contaminadas por ruído, erros de calibração do sistema de aquisição e de descrição do objeto.

5.1 Trabalhos Futuros

O complemento deste trabalho seria a elaboração de textura para o modelo 3D gerado e detecção de regiões que apresentam anomalias na superfície do objeto, como a inclusão de outros modelos geométricos 3D da família das superquádricas ao método apresentado e o desenvolvimento de modelos matemáticos de superquádricas de objetos deformáveis para integrar os objetos a um banco de imagens 3D da família das superquádricas. De fato, a parte inicial da metodologia apresentada, a geração de objetos sintéticos, pode contribuir para pesquisas e estudos de mais objetos da família de superquádricas e acrescentar possibilidades de novas geometrias de objetos. Esses objetos, por sua vez, podem ser utilizados para o projeto de um sistema que construa o mapa de geometria da cena e o reconhecimento de objetos que pertençam à família das superquádricas, o que pode ser relevante para um manipulador robótico que realize tarefas de manipulação.

6 REFERÊNCIAS BIBLIOGRÁFICAS

WATT, A. e POLICARPO F., The Computer Image, Addison-Wesley Pub Co (Net); 1998.

BARDINET, E., COHEN, L. D. e AYACHE, N. A Parametric Deformable Model To Fit Unstructured 3 D Data. Research report 2617 INRIA. 1995.

BELINI, V. L., Elaboração de textura para o modelo 3D gerado e detecção de regiões que apresentam anomalias na superfície do objeto, dentre outras aplicações. Tese de Doutorado. Escola de Engenharia de São Carlos da Universidade de São Paulo, 2010.

BARR, A. H., Global and Local Deformations of Solid Primitives. Computer Graphics, Vol. 18, Number 3, pp. 21- 30, 1984.

BARR, A. H.; Superquadrics and Angle-Preserving Transformations, IEEE Computer Graphics and Applications, pp. 11-22. 1981.

BENKO H. E WILSON A., DepthTouch: Using depth-sensing camera to enable freehand interactions on and above the interactive surface,” Microsoft, Tech. Rep., 2009.

BIEGELBAVER G. e VINCZE M., Efficient 3D Object Detection by Fitting Superquadrics to Range Image Data for Robot’s Object Manipulation. IEEE International Conference ON Robotics and Automation, pp 1086 – 1091. 2007.

CHEVALIER, L., JAILLET, F., E BASKURT, A. Segmentation and superquadric modeling of 3D objects. WSCG03, Plzen (CZ), vol. 11(1), February, 2003. ISSN 1213-6972.

GELB D., SUBRAMANIAN A., E TAN K. H., Augmented Reality for Immersive Remote Collaboration, HP Laboratories, Tech. Rep., 2010.

DOP, V., REGTIEN P.; Object Recognition From Range Images Using Superquadric Representations. IAPR Workshop on Machine Vision Applications, November, pp. 12-14, Tokyo, Japan, 1996.

EL-KHOURY S.; .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.

TRUCCO E. e VERRI A. , Introductory Techniques for 3-D Computer Vision, Prentice-Hall 1998, ISBN 0-13-261108-2.

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.

GAVIN H.; The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems. Dept. Civil & Environ. Engineering, Duke Univ. 13, 2011. Disponivel: <http://people.duke.edu/~hpgavin/ce281/lm.pdf>

GUINNESSWORLDRECORDS.com. "Kinect Confirmed As Fastest- Selling Consumer Electronics Device".Último acesso em 17 de setembro de 2011.

GUPTA, A., BOGONI, L AND 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.; 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.

JAKLIC A., LEONARDIS A., SOLINA F.; Segmentation and Recovery of Superquadrics. Kluwer Academic Publishers, Dordrecht (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.

KHOSHELHAM, K.; ELBERINK, S.O. Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *Sensors* **2012**, 12, 1437-1454.

KRIVIC, J.: Segmentation and 3D tracking of superquadric modeled objects. A dissertation in Computer and Information Science. Ljubljana, 2006.

LEONARDIS, A., JAKLIC A. AND SOLINA, F. Superquadrics for segmenting and modeling range data. *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 11, pp. 1269-1295. 1997.

LOURAKIS, M.:levmar: Levenberg-Marquardt nonlinear least squares algorithms in C/C++. <http://www.ics.forth.gr/~lourakis/levmar/>. Acessadoem 22/06/2010.

MATEUS S, e BRANCH J. Ajuste de Objetos 3D Mediante Parches NURBS Optimizados con Estrategias Evolutivas. *Revista avances en Sistemas e Informatica. Edicion Especial*. Pp, 141- 145. Medellin.2008

MICROSOFT, “Microsoft Kinect Web Site,” Jan 2011. (Online). Available: <http://www.xbox.com/en-US/kinect>Microsoft, 2010.

OGATA, K.; *Modern Control Engineering*, Prentice Hall. 2009

OPEN KINECT, Open Kinect Community Site, Jan 2011. (Online). Available: <http://openkinect.org/>

PENTLAND, A. P., Automatic Extraction of Deformable Part Models. *Computer Vision*, pp. 107-126, 1990.

PENTLAND, A. P., Recognition by Parts. *IEEE First Int. Conf. On Computer Vision*. pp. 612-620, London, 1987.

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

PONCE, J. AND BRADY, M., Toward a Surface Primal Sketch, in Three-Dimensional Machine Vision, T.Kanade (Ed.), Kluwer Academic Publishers, 1987. pp. 225-227.

RUSENKIEWICZ, S. AND LEVOY, M., A multiresolution point rendering system for large meshes. ACM , New York, NY, 343-352. DOI= <http://doi.acm.org/10.1145/344779.344940>, 2000.

RUSSELL, S. E NORVIG, P. Artificial Intelligence: A Modern Approach. New Jersey, Prentice Hall. 1995

SITEK, A., HUESMAN, R.H. e Gullberg, G.T.; Tomographic Reconstruction Using an Adaptive Tetrahedral Mesh Defined by a Point Cloud. IEEE Trans. Med. Imag. Pp, 1172-1179 .2006.

SMISEK J., JANCOSSEK M., PAJDLA T., Capitulo 1. 3D with Kinect.Consumer Depth Cameras for Computer Vision”.Advances em computer Vision and Pattern Recognition, London 2013.

SOLINA F., BAJCSY R., Recovery of parametric models from range images: The case for superquadrics with global deformations. IEEE Trans. Pattern Anal Machine Intell., pp 131-147. 1990.

STRAND, M. AND DILLMANN, R., Segmentation and Approximation of Objects in Pointclouds using Superquadrics.IEEE International Conference on Information and Automation, Zhulai/Macau, pp, 887-903. China.2009.

TEIXEIRA J. M, REIS B., MACEDOS E, e KELNER J. Uma avaliação estatística do problema de registro de imagens de profundidade usando o Kinect. Centro de Informática, Universidade Federal de Pernambuco. TR420, ICS-FORTH, July 2011.

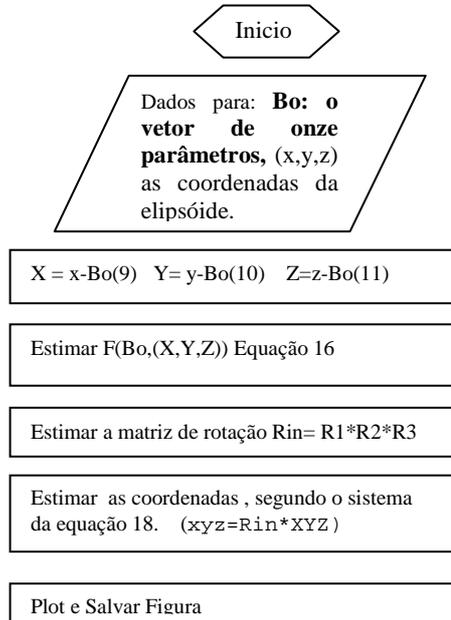
YOKOYA, N., LEVINE, M. D., Range Image Segmentation Based on Differential Geometry: A Hybrid Approach", IEEE Trans. on Pattern Analysis and Machine Intelligence, pp. 643-649. 1989.

WHAITE, P.; FERRIE, F., From Uncertainty to Visual Exploration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13 (10): pp. 1038-1049, October, 1991.

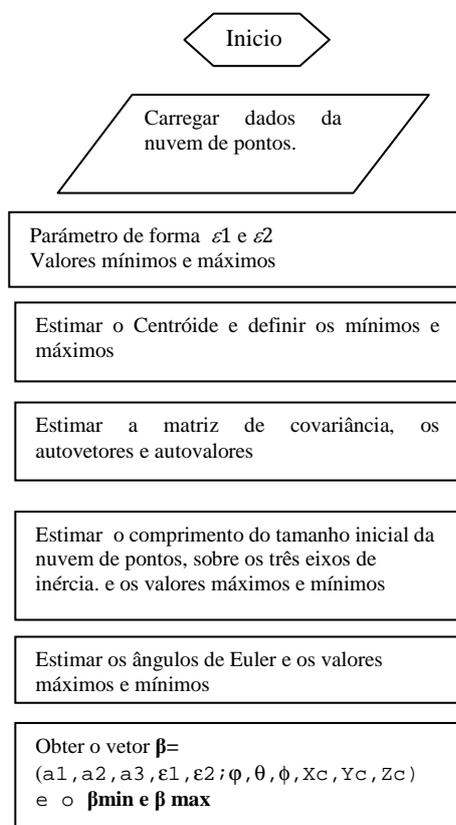
ZHANG, Y., *Experimental comparison of superquadric fitting objective functions*. Pattern Recognition Letters, pp 2185-2193, 2003

APÊNDICE 1: Programa de Geração De Pontos

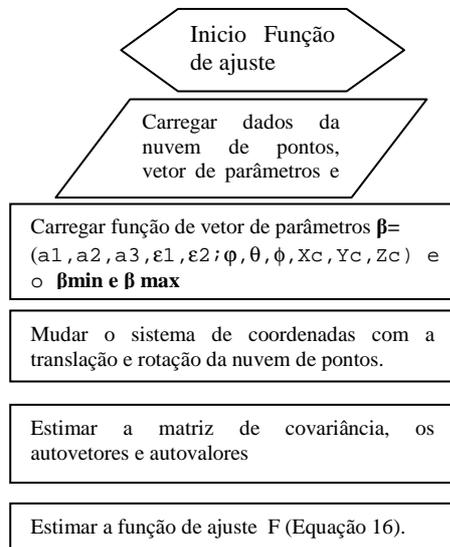
Programa de gerar a nuvem de pontos de uma elipsóide segundo procedimento do item 3.1.



**APÊNDICE 2. Programa de Estimativa Dos Parâmetros Iniciais Da
Nuvem De Pontos**



APÊNDICE 3. Programa de Estimativa Da Função Objetivo



APÊNDICE 4. Programa de Reconstruir O Objeto Em 3d

Início

Carregar arquivo de dados da nuvem de pontos
Vetor \mathbf{F}_o ($1 \times N$) = $\mathbf{1}$, sendo N o numero de pontos da nuvem
Matriz de ponderação $\mathbf{W}_{N \times N} = \mathbf{1}$

A saída da Função objetivo
(Apêndice 2) β , β_{\min} e β_{\max}

A saída da função de ajuste
 \mathbf{F} (Apêndice 3)

A saída do Algoritmo de Levenberg
Marquardt (Anexo 1): O vetor β
ajustado

Obter as coordenadas do objeto modelo com o
vetor β **ajustado** e fazer o gráfico .