

JOÃO GUILHERME DAREZZO MARTINS DE FRANÇA

DESENVOLVIMENTO DE UM SISTEMA DE  
AQUISIÇÃO DE INFORMAÇÕES VOLUMÉTRICAS  
USANDO MÉTODO DE TRIANGULAÇÃO A LASER E  
CAMPO DE VISÃO VARIÁVEL

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências Exatas e de Tecnologia da Universidade Federal de São Carlos como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Processamento de Imagens e Sinais.

Orientador: Prof. Dr. José Hiroki Saito

UFSCar – São Carlos  
Junho/2004

**Ficha catalográfica elaborada pelo DePT da  
Biblioteca Comunitária da UFSCar**

F814ds

França, João Guilherme Darezzo Martins de.  
Desenvolvimento de um sistema de aquisição de  
informações volumétricas usando método de triangulação a  
laser e campo de visão variável / João Guilherme Darezzo  
Martins de França. -- São Carlos : UFSCar, 2004.  
121 p.

Dissertação (Mestrado) -- Universidade Federal de São  
Carlos, 2004.

1. Computação gráfica. 2. Visão computacional. 3.  
Aquisição 3D. 4. Processamento de imagens. 5. I. Título.

CDD: 006.6 (20<sup>a</sup>)

# Agradecimentos

---

*Em primeiro lugar agradeço a meus pais, Azuaite e Cleide pelo amor, pela fé e pelo apoio que nunca me faltou. Com eles experimentei o sabor de desafiar as impossibilidades.*

*Agradeço, também, a meus amigos e colegas de laboratório, principalmente àqueles do grupo de Arquitetura e Processamento de Imagens e Sinais. Eles foram companheiros na reflexão e generosos na solução de dúvidas. Sei, perfeitamente, o quanto me ajudaram, em especial o Mário que me auxiliou nos sistemas de controle.*

*Agradeço, finalmente, a meu orientador Prof. Saito por ter acreditado na possibilidade desse trabalho ser desenvolvido. Com o Prof. Saito aprendi muito, especialmente a admirá-lo pelo conhecimento científico e pela referência intelectual que representa. Mas, acima de tudo, admiro-lhe o caráter, a perseverança e a simplicidade despojada.*

*João Guilherme*

# Resumo

---

Os sistemas de aquisição 3D mais eficientes utilizam o princípio da triangulação laser ou o princípio do tempo de vôo (TOF - *time of flight*). Nos sistemas baseados em triangulação, o alcance e a variação de profundidade são muito limitados, mas possuem grande precisão. Por outro lado, o oposto ocorre nos sistemas TOF, onde a precisão é baixa, e a variação de profundidade e o alcance são altos. Esse trabalho descreve o projeto de um sistema de aquisição 3D, com precisão e alcance superiores aos sistemas 3D tradicionais, baseados em triangulação laser e versatilidade similar aos sistemas TOF. Um protótipo simplificado desse sistema foi implementado.

# Abstract

---

The most efficient 3D acquisition systems use the principle of laser triangulation or the principle of time of flight – TOF. In the systems based on triangulation, the range and the depth variation are very limited, but it has a great precision. On the other hand the opposite occurs to the TOF systems where it has a low precision, and a great range and depth variation. This work describes a project of a 3D acquisition system, with greater precision and range than traditional laser triangulation 3D scanners, and it's as versatile as TOF systems. An initial prototype of this system had been implemented.

# Sumário

---

<b>INTRODUÇÃO</b> .....	<b>1</b>
<b>1.1. ORGANIZAÇÃO DO TRABALHO</b> .....	<b>1</b>
<b>1.2. OBJETIVOS</b> .....	<b>2</b>
<b>1.3. JUSTIFICATIVA</b> .....	<b>3</b>
<b>REVISÃO DA LITERATURA</b> .....	<b>4</b>
<b>2.1. SENSORES</b> .....	<b>4</b>
2.1.1. <i>Sensores Ativos</i> .....	4
2.1.2. <i>Sensores Passivos</i> .....	11
<b>2.2. RECONSTRUÇÃO</b> .....	<b>15</b>
2.2.1. <i>Superfícies Poligonais</i> .....	15
2.2.2. <i>Remoção de Triângulos nas Regiões de Sobreposição</i> .....	18
2.2.3. <i>Unindo os Patches de Redes de Superfície</i> .....	21
<b>2.3. PROCESSAMENTO DE IMAGENS</b> .....	<b>22</b>
2.3.1. <i>Álgebra de Imagem</i> .....	22
2.3.2. <i>Filtros Espaciais</i> .....	25
2.3.3. <i>Detecção de Bordas</i> .....	35
<b>2.4. APLICAÇÕES</b> .....	<b>50</b>
<b>2.5. SISTEMAS DE AQUISIÇÃO 3D</b> .....	<b>52</b>
2.5.1. <i>Luz Estruturada</i> .....	52
2.5.2. <i>Estéreo Omnidirecional</i> .....	56
2.5.3. <i>Tempo de Voo Laser</i> .....	61
<b>2.6. CONSIDERAÇÕES FINAIS</b> .....	<b>63</b>
<b>PROJETO E IMPLEMENTAÇÕES DE UM SISTEMA DE AQUISIÇÃO 3D</b> .....	<b>65</b>
<b>3.1. AQUISIÇÃO 3D POR TRIANGULAÇÃO LASER</b> .....	<b>65</b>
<b>3.2. SISTEMA DE AQUISIÇÃO DE INFORMAÇÕES VOLUMÉTRICA USANDO MÉTODO DE TRIANGULAÇÃO LASER E CAMPO DE VISÃO VARIÁVEL</b> .....	<b>67</b>
<b>3.3. MODELO VIRTUAL</b> .....	<b>71</b>
<b>3.4. IMPLEMENTAÇÃO DO SISTEMA DE AQUISIÇÃO 3D</b> .....	<b>73</b>
3.4.1. <i>Sensoriamento</i> .....	73
3.4.2. <i>Varredura</i> .....	75
3.4.3. <i>Estimador de distância</i> .....	78
3.4.4. <i>Software de Controle</i> .....	89
3.4.5. <i>Reconstrução</i> .....	103
3.4.6. <i>Interface Gráfica</i> .....	104
<b>RESULTADOS OBTIDOS</b> .....	<b>106</b>
<b>CONCLUSÕES</b> .....	<b>116</b>

# Lista de Figuras

---

FIGURA 2.1: TÍPICA CONFIGURAÇÃO DE SENSOR LASER. ....	6
FIGURA 2.2: TÍPICA GEOMETRIA DE LUZ ESTRUTURADA. ....	7
FIGURA 2.3: CONFIGURAÇÃO DE SENSOR MOIRÉ. ....	8
FIGURA 2.4: O EFEITO DO TAMANHO DA SONDA NAS MEDIDAS DE DISTÂNCIA. ....	9
FIGURA 2.5: CONFIGURAÇÃO ESTÉREO PARALELO. ....	11
FIGURA 2.6: CONFIGURAÇÃO DE PROFUNDIDADE-POR-FOCO. ....	14
FIGURA 2.7: REPRESENTAÇÃO EM MODELO DE ARAME DE UM CILINDRO. ....	15
FIGURA 2.8: REPRESENTAÇÃO DE TABELA DE DADOS GEOMÉTRICOS PARA DUAS SUPERFÍCIES DE POLÍGONOS ADJACENTES, FORMADOS POR SEIS MARGENS E CINCO VÉRTICES. ....	16
FIGURA 2.9: TABELA DE MARGENS PARA AS SUPERFÍCIES DA FIGURA 2.8 EXPANDIDA PARA INCLUIR PONTEIROS PARA A TABELA DE POLÍGONOS. ....	17
FIGURA 2.10: DOIS REGISTROS DE AQUISIÇÃO. ....	18
FIGURA 2.11: CAIXA DE DELIMITADORA. ....	19
FIGURA 2.12: TRIÂNGULOS EM INTERSECÇÃO. ....	19
FIGURA 2.13: CÍRCULO DE REMOÇÃO. ....	19
FIGURA 2.14: SOBREPOSIÇÃO DEPENDENTE DO PONTO DE VISTA. PELA VISTA 1 HÁ SOBREPOSIÇÃO, MAS PELA VISTA 2 NÃO. ....	20
FIGURA 2.15: NÃO "OLHANDO PARA FRENTE", MAS DEVE VERIFICAR SOBREPOSIÇÃO. (A) DUAS VISTAS, (B) RESULTADOS DA AMOSTRAGEM DE 2 VISTAS. ....	21
FIGURA 2.16: DUAS VISTAS DEPOIS DE REMOÇÃO DE TRIÂNGULOS SOBREPOSTOS. ....	21
FIGURA 2.17: A) PRIMEIRA IMAGEM ORIGINAL; B) SEGUNDA IMAGEM ORIGINAL; C) ADIÇÃO DAS IMAGENS (A) E (B). D) IMAGEM ORIGINAL; E) RUÍDO GAUSSIANO; F) ADIÇÃO DAS IMAGENS (D) E (E). ....	23
FIGURA 2.18: A) CENA ORIGINAL; B) MESMA CENA POSTERIORMENTE; C) SUBTRAÇÃO DA (B) E (A), NOTE QUE SOMENTE OBJETO QUE SE MEXEU NA CENA APARECE NO RESULTADO. ....	24
FIGURA 2.19: A) IMAGEM ORIGINAL; B) IMAGEM MULTIPLICADA POR 1.6; C) IMAGEM DIVIDIDA POR 1.6. ....	24
FIGURA 2.20: A) IMAGEM ORIGINAL; B) RETÂNGULO PARA A MÁSCARA AND; C) IMAGEM RESULTANTE (A) AND (B); D) RETÂNGULO PARA A MÁSCARA OR; C) IMAGEM RESULTANTE (A) OR (D); ....	26
FIGURA 2.21: A) IMAGEM ORIGINAL; B) OPERADOR NOT APLICADO NA IMAGEM. ....	26
FIGURA 2.22: A) IMAGEM ORIGINAL; B) APLICAÇÃO DE FILTRO MÉDIO 3X3. NOTE A APARÊNCIA MAIS SUAVE. ....	27
FIGURA 2.23: A) IMAGEM ORIGINAL COM A ADIÇÃO DE RUÍDO SAL-E-PIMENTA; B) IMAGEM RESTAURADA COM FILTRO MEDIANO DE 3X3. ....	28
FIGURA 2.24: A) IMAGEM ORIGINAL; B) IMAGEM (A) APÓS APLICAÇÃO DE FILTRO LAPLACIANO. ....	29
FIGURA 2.25: A) IMAGEM ORIGINAL; B) RESULTADO DO FILTRO DE DIFERENÇA HORIZONTAL; C) RESULTADO DO FILTRO DE DIFERENÇA VERTICAL; D) RESULTADO DO FILTRO DE DIFERENÇA DIAGONAL 1; E) RESULTADO DO FILTRO DE DIFERENÇA DIAGONAL 2. ....	30
FIGURA 2.26: RESULTADOS DE SUA VIZIAÇÃO USANDO FILTRO GAUSSIANO 3X3. (A) IMAGEM ORIGINAL CORROMPIDA POR RUÍDO GAUSSIANO. (B) IMAGEM SUA VIZIADA. ....	35
FIGURA 2.27: BORDA IDEAL VS. REAL. (A) BORDA IDEAL E (B) BORDA REAL. ....	36
FIGURA 2.28: $F_1, F_2, F_3$ E $F_4$ COMPREENDEM O SUB-ESPAÇO DE BORDAS; $F_5, F_6, F_7$ E $F_8$ COMPREENDEM O SUB-ESPAÇO DE LINHAS E $F_9$ COMPREENDE O SUB-ESPAÇO DE MÉDIA. ....	45
FIGURA 2.29: UMA REPRESENTAÇÃO BIDIMENSIONAL DO CONCEITO DE PROJEÇÃO DE FREI-CHEN. O ESPAÇO REAL DE FREI-CHEN É NOVE-DIMENSIONAL, ONDE CADA DIMENSÃO É DADA POR UMA DAS MÁSCARAS $F_k$ . ....	46
FIGURA 2.30: PARTIÇÃO DAS POSSÍVEIS ORIENTAÇÕES DO GRADIENTE EM SETORES PARA A SUPRESSÃO NÃO-MÁXIMA. EXISTEM QUATRO SETORES NUMERADOS DE 0 A 3, CORRESPONDENDO ÀS QUATRO POSSÍVEIS COMBINAÇÕES DE ELEMENTOS EM UMA VIZINHANÇA DE 3X3 QUE UMA LINHA PASSA ATRAVÉS À MEDIDA QUE ELA PASSA PELO CENTRO DA VIZINHANÇA. AS DIVISÕES DO CÍRCULO DE POSSÍVEIS ORIENTAÇÕES DE LINHAS ESTÃO EM GRAUS. ....	48
FIGURA 2.31: UMA SEQÜÊNCIA DE QUATRO QUADROS DE PADRÕES PROJETADOS. CADA PADRÃO TEM 110 LINHAS DE BORDA (111 LINHAS), COM A PARTICULARIDADE QUE CADA LINHA DE BORDA TEM UM ÚNICO CÓDIGO (CONSTITUÍDO DE UM HISTÓRICO ILUMINAÇÃO PRETO OU BRANCO DE CADA LADO DA BORDA POR UMA SEQÜÊNCIA DE QUATRO QUADROS). ....	55
FIGURA 2.32: UMA SEQÜÊNCIA DE QUADROS DE VÍDEO DE UM ELEFANTE ILUMINADO COM O CÓDIGO DE BORDA DE LINHA. O PROJETO FICA ALTERNANDO ENTRE ESSES QUATRO PADRÕES A 60Hz E O OBJETO ILUMINADO É FOTOGRAFADO COM UMA CÂMERA NTSC SINCRONIZADA AO PROJETO. ....	55

FIGURA 2.33: RASTREAMENTO DE LINHAS DE BORDA. O SISTEMA IDENTIFICA OS LIMITES NOS QUADROS DE VÍDEO, RASTREIA A LINHA DE BORDA PELO TEMPO, E PROPAGA O HISTÓRICO DE ILUMINAÇÃO DE CADA BORDA POR QUATRO QUADROS PARA DETERMINAR O CÓDIGO DE BORDA. NOTE-SE QUE O ESTÁGIO DE RASTREAMENTO TEM QUE INFERIR, NA PRESENÇA DE BORDAS “FANTASMA” (MOSTRADOS EM LINHAS TRACEJADAS) EM ALGUMAS DAS REGIÕES P&B COM LARGURA DUPLA. OS PADRÕES PROJETADOS SÃO DESENHADOS PARA MINIMIZAR A PRESENÇA DESSAS BORDAS INVISÍVEIS E TEM A PROPRIEDADE DE QUE NUNCA EXISTEM DOIS FANTASMAS ADJACENTES NO ESPAÇO E TEMPO. ....	56
FIGURA 2.34: DIAGRAMA ESQUEMÁTICO DE UMA CÂMERA ESTEREOSCÓPICA PANORÂMICA COMPACTA. DUAS CÂMERAS OMNIDIRECIONAIS SÃO UTILIZADAS PARA CAPTURAR OS DADOS PANORÂMICOS ESTEREOSCÓPICOS. O IMAGEAMENTO OMNIDIRECIONAL É FEITO POR PROJEÇÃO ORTOGRÁFICA DE UMA SUPERFÍCIE PARABÓLICA REFLEXIVA. A ÓTICA TELECÊNTRICA É USADA PARA FAZER A PROJEÇÃO ORTOGRÁFICA ENQUANTO MANTÉM A CÂMERA PERTO DO ESPELHO. ....	57
FIGURA 2.35: O CAMPO DE VISÃO. ESTENDENDO A PARÁBOLA ALÉM DO PONTO FOCAL, OS ESPELHOS PARABÓLICOS SÃO CAPAZES DE CAPTURAR UM CAMPO DE VISÃO MAIOR QUE UM HEMISFÉRIO. UM CONE DE OCLUSÃO É CRIADO ONDE A CÂMERA “SE VÊ”. ....	58
FIGURA 2.36: MAPEANDO UM PONTO DA IMAGEM PARA UM PANORAMA. UM PONTO NA IMAGEM P É MAPEADO PARA O PONTO P’ NO PANORAMA. ....	59
FIGURA 2.37: GEOMETRIA EPIPOLAR. (A) AS LINHAS EPIPOLARES PARA UM PAR DE IMAGENS OMNIDIRECIONAIS COAXIAIS SÃO LINHAS RADIAIS. (B) QUANDO AS IMAGENS SÃO PROJETADAS SOBRE UM PANORAMA AS LINHAS EPIPOLARES SE TORNAM PARALELAS. ....	60
FIGURA 2.38: TRIANGULAÇÃO E COMPUTAÇÃO DE PROFUNDIDADE. DADO UM PAR DE PONTOS CORRESPONDENTES P E P’ DAS IMAGENS, O PONTO DE CENA P É ENCONTRADO POR TRIANGULAÇÃO. ....	61
FIGURA 2.39: SCANNER LASER PLANAR ..... 62	62
FIGURA 2.40: MAPEAMENTO AÉREO COM O LASER PLANAR. .... 63	63
FIGURA 3.1: GEOMETRIA DO SISTEMA DE AQUISIÇÃO POR TRIANGULAÇÃO LASER. .... 66	66
FIGURA 3.2: QUADRO DE AQUISIÇÃO DE VÍDEO. .... 67	67
FIGURA 3.3: EM (A) O FOV $\lambda$ E O CAMPO DE PROFUNDIDADE $D$ SÃO MAIORES QUE EM (B), MAS A PRECISÃO É MAIOR EM (B). .... 68	68
FIGURA 3.4: SEQÜÊNCIA DE AQUISIÇÃO INICIAL (A) (B) (C). AJUSTE DO FOV E ÂNGULO DO LASER (D). SEGUNDA AQUISIÇÃO (D) (E) (F). .... 70	70
FIGURA 3.5: (A) SISTEMA AJUSTADO PARA A PRIMEIRA AQUISIÇÃO. (B) SEGUNDA AQUISIÇÃO, AJUSTE DO LASER SEM O AJUSTE DO FOV ( $D' < D$ ). .... 71	71
FIGURA 3.6: (A) IMAGEM EM RGB, (B) CANAL VERMELHO DE (A), (C) CANAL VERDE DE (A), (D) CANAL AZUL DE (A), (E) INTENSIDADE DE (A) E (F) LIMIAZIZAÇÃO DE (B). .... 72	72
FIGURA 3.7: OBJETO 3D GERADO PELA SEQÜÊNCIA DE IMAGENS SEGMENTADAS. (A) ESCALA DE DISTÂNCIA EM (MM). (B) SOMENTE OS PONTOS VIZINHOS COM $X$ DIFERENTE SÃO APRESENTADOS. 73	73
FIGURA 3.8: ESTRUTURA BÁSICA DO SISTEMA DE AQUISIÇÃO 3D. .... 74	74
FIGURA 3.9: MÓDULO DE SENSORIAMENTO. .... 74	74
FIGURA 3.10: MÓDULO DE VARREDURA. .... 75	75
FIGURA 3.11: MÓDULO DE ESTIMAÇÃO DE DISTÂNCIA. .... 79	79
FIGURA 3.12: (A) PADRÃO UTILIZADO NA CALIBRAÇÃO DA CÂMERA, (B) AS CRUZES EM VERMELHO MOSTRADAS INDICAM A ESTIMATIVA DE ONDE OS CANTOS DE CADA QUADRADO ESTÃO POSICIONADOS. (C) CORREÇÃO AUTOMÁTICA DOS CANTOS DOS QUADRADOS, ANALISANDO AS JANELAS DE 5X5 PIXELS, MOSTRADAS EM AZUL. .... 80	80
FIGURA 3.13: COMPONENTE RADIAL DO MODELO DE DISTORÇÃO. .... 81	81
FIGURA 3.14: COMPONENTE TANGENCIAL DO MODELO DE DISTORÇÃO. .... 81	81
FIGURA 3.15: MODELO DE DISTORÇÃO COMPLETO. .... 82	82
FIGURA 3.16: (A) IMAGEM ORIGINAL, (B) IMAGEM COM CORREÇÃO DE DISTORÇÃO. .... 82	82
FIGURA 3.17: (A) CANAL VERMELHO, (B) CANAL VERDE E (C) CANAL AZUL. .... 83	83
FIGURA 3.18: (A) IMAGEM DE REFERÊNCIA, (B) IMAGEM DA SEQÜÊNCIA DE AQUISIÇÃO, (C) SUBTRAÇÃO DE (B) E (A), (D) IMAGEM (C) MULTIPLICADA POR 20, (E) ELIMINAÇÃO DE PIXELS QUE NÃO SÃO 8-CONECTADOS DE (C), (F) IMAGEM (E) MULTIPLICADA POR 20, (G) FILTRAGEM MEDIANA DE (E), (H) IMAGEM (G) MULTIPLICADA POR 20. .... 84	84
FIGURA 3.19: DETECÇÃO DE BORDAS. (A) MAGNITUDE DE PREWITT, (B) LIMIAZIZAÇÃO DE (A) $T=200$ , (C) MAGNITUDE DE PREWITT SEM A SEGUNDA BORDA DE COLUNA, (D) LIMIAZIZAÇÃO DE (C) $T=110$ E (E) CANNY. .... 86	86
FIGURA 3.20: ESQUEMÁTICO DO ALGORITMO DE SEGMENTAÇÃO DE IMAGENS. .... 87	87
FIGURA 3.21: ESQUEMÁTICO DO ALGORITMO DE INICIALIZAÇÃO DO SISTEMA DE VARREDURA. .... 91	91

FIGURA 3.22: LIMITES DE VARREDURA. ....	92
FIGURA 3.23: (A) PLOTAGEM DA CORRENTE EM FUNÇÃO DO TEMPO. O PRIMEIRO PICO INDICA A MUDANÇA DE VELOCIDADE DE 0 PARA 4000 RPM E O SEGUNDO PICO INDICA A MUDANÇA DE VELOCIDADE DE 4000 PARA 0 RPM. (B) PLOTAGEM DE VELOCIDADE EM FUNÇÃO DO TEMPO COM ACELERAÇÃO DE 30000 REVOLUÇÕES/S <sup>2</sup> . ....	92
FIGURA 3.24: ESQUEMÁTICO DO ALGORITMO DE CALIBRAÇÃO DO TERMO PROPORCIONAL E INTEGRAL. ....	94
FIGURA 3.25: ESQUEMÁTICO DO ALGORITMO DE CALIBRAÇÃO ENTRE A CÂMERA E O LASER. ....	97
FIGURA 3.26: ESQUEMÁTICO DA PRIMEIRA PARTE DO ALGORITMO DE AQUISIÇÃO. ....	101
FIGURA 3.27: ESQUEMÁTICO DO ALGORITMO DE AQUISIÇÃO COM MODO DE POSICIONAMENTO IGUALMENTE ESPAÇADO. ....	102
FIGURA 3.28: ESQUEMÁTICO DO ALGORITMO DE AQUISIÇÃO, MODO DE POSICIONAMENTO. ....	102
FIGURA 3.29: ESQUEMÁTICO DO ALGORITMO DE AQUISIÇÃO, MODO DE VELOCIDADE. ....	103
FIGURA 3.30: MÓDULO DE RECONSTRUÇÃO. ....	104
FIGURA 3.31: INTERFACE GRÁFICA DO SISTEMA DE AQUISIÇÃO. ....	105
FIGURA 4.1: NUVENS DE PONTOS OBTIDAS PELA PRIMEIRA IMPLEMENTAÇÃO: (A) CAVEIRA (15419 PONTOS), (B) ELEFANTE (20870 PONTOS), (C) ANJO (24885 PONTOS) E (D) BONECO DE PALHA (22900 PONTOS). ....	108
FIGURA 4.2: NUVENS DE PONTOS OBTIDAS PELA PRIMEIRA IMPLEMENTAÇÃO COM OS OBJETOS NA HORIZONTAL: (A) ASTROBOY (18579 PONTOS), (B) ANJO DE PORCELANA (27515 PONTOS), (C) ANJO FRENTE (43797 PONTOS) E (D) ANJO COSTAS (43333 PONTOS). ....	109
FIGURA 4.3: NUVENS DE PONTOS OBTIDAS PELA AQUISIÇÃO FEITA EM DIFERENTES ÂNGULOS PELA SEGUNDA IMPLEMENTAÇÃO. (A) (80522 PONTOS), (B) (87093 PONTOS), (C) (92396 PONTOS) E (D) (69320 PONTOS). ....	110
FIGURA 4.4: (A) OBJETO COM A APLICAÇÃO DE TEXTURA, (B) OBJETO COM MAPEAMENTO DE COR EM FUNÇÃO DA DISTANCIA Z. (C) CARRO (57083 PONTOS) EFEITO DE INCRUSTAMENTO CAUSADO PELA TEXTURA DO OBJETO ORIGINAL NA AQUISIÇÃO. ....	112
FIGURA 4.5: NUVENS DE PONTOS OBTIDAS PELA AQUISIÇÃO DA TERCEIRA IMPLEMENTAÇÃO. (A) CARRO (49083 PONTOS) E (B) (136352 PONTOS), AQUISIÇÕES FEITAS NO MODO DE POSICIONAMENTO. (C) CARRO (98223 PONTOS) AQUISIÇÃO FEITA NO MODO DE VELOCIDADE E (D) SUPERFÍCIE TRIANGULAR DE (B). ....	113
FIGURA 4.6: SISTEMA DE VARREDURA DO SEGUNDO (A) E TERCEIRO PROTÓTIPO (B). ....	115

# Lista de Tabelas

---

TABELA 3.1: INFORMAÇÕES TÉCNICAS DO MOTOR. ....	76
TABELA 3.2: INFORMAÇÕES TÉCNICAS DO REDUTOR MECÂNICO. ....	76
TABELA 3.3: INFORMAÇÕES TÉCNICAS DO <i>ENCODER</i> . ....	76
TABELA 3.4: INFORMAÇÕES TÉCNICAS DO CONTROLADOR DE MOVIMENTO. ....	77
TABELA 3.5: PARÂMETROS DE CALIBRAÇÃO DA CÂMERA. ....	82
TABELA 3.6: TEMPO DAS OPERAÇÕES DO MÓDULO DE ESTIMAÇÃO DE DISTÂNCIA. ....	88

# Lista de Algoritmos

---

ALGORITMO 1: DETECÇÃO DE BORDA CANNY.....	50
ALGORITMO 2 SEGMENTAÇÃO DE IMAGENS.....	86
ALGORITMO 3 ESTIMAÇÃO DE DISTÂNCIA.....	88
ALGORITMO 4 INICIALIZAÇÃO DO SISTEMA DE VARREDURA.....	89
ALGORITMO 5 CALIBRAÇÃO DO TERMO PROPORCIONAL E INTEGRAL.....	93
ALGORITMO 6 CALIBRAÇÃO LASER CÂMERA.....	95
ALGORITMO 7 AQUISIÇÃO.....	99
ALGORITMO 8 AQUISIÇÃO, MODO DE POSICIONAMENTO IGUALMENTE ESPAÇADO.....	99
ALGORITMO 9 AQUISIÇÃO, MODO DE POSICIONAMENTO.....	100
ALGORITMO 10 AQUISIÇÃO, MODO DE VELOCIDADE.....	100

# Capítulo 1

## Introdução

A computação gráfica nos permite gerar imagens realísticas do mundo em três dimensões, mas antes que essas imagens possam ser geradas, modelos gráficos do mundo devem estar disponíveis. Tradicionalmente esses modelos são obtidos por modelagem 3D, um processo muito demorado e limitado em realismo de detalhes. Esse trabalho descreve métodos automáticos de aquisição de objetos reais usando sensores de distância.

O sistema de aquisição volumétrica é constituído, basicamente, de três partes: a) sensoriamento – onde a geometria de superfície da cena de interesse é capturada, b) processamento de imagens – que tem a função de segmentar a projeção de luz utilizada no sensoriamento para que, a partir dela, parâmetros possam ser extraídos e c) reconstrução – que usa como entrada os dados brutos do sensoriamento e os transforma em objetos poligonais tridimensionais. Geralmente os sensores dos sistemas de aquisição 3D produzem como saída uma nuvem de pontos ou imagem de profundidade. A nuvem de pontos é um conjunto de coordenadas de vértices e a imagem de profundidade é uma matriz 2D onde os valores de intensidade de cada *pixel* representam a distância entre o objeto alvo e o sensor. Para que essa informação possa ser visualizada, modelos de superfície devem ser criados a partir de dados. Como o scanner 3D só pode fazer a aquisição de dados onde o sensor “vê” o alvo, reconstruir objetos inteiros ou cenas, geralmente, envolve a integração de várias aquisições feitas em diferentes pontos de vista. O problema de interesse é então, construir uma única representação de superfície para essas múltiplas aquisições. Este trabalho trata da implementação de sistemas de aquisição volumétricos baseados em triangulação laser, gerando como resultados, uma série de nuvens de pontos que demonstram como cada fator pode influenciar nos resultados.

### 1.1. Organização do trabalho

A presente dissertação foi organizada em cinco capítulos de forma a conduzir o leitor pelo caminho das experiências vivenciadas. Assim sendo, além desta introdução

que inicia o trabalho apresentando o tema desenvolvido, tem-se uma necessária revisão bibliográfica que permitiu que se implementasse um sistema de aquisição 3D, os resultados dos experimentos e finalmente as conclusões finais com sugestões e propostas para futuros trabalhos de pesquisa.

- Capítulo 2 – *Revisão da Literatura*: Nesse capítulo é apresentado um levantamento bibliográfico sobre visão computacional onde serão discutidas várias técnicas de sensoriamento, uma breve abordagem sobre reconstrução, o processamento de imagens, abordando álgebra de imagens, técnicas de pré-processamento e detecção de bordas e algumas aplicações de sensoriamento 3D; e trabalhos relacionados de sistemas de aquisição 3D, usando diferentes tecnologias.
- Capítulo 3 – *Projeto e Implementações de um Sistema de Aquisição 3D*: Nesse capítulo, são descritas a metodologia e as implementações realizadas ao longo desse trabalho.
- Capítulo 4 – *Resultados Obtidos*: Nesse capítulo são apresentados e discutidos os resultados obtidos nas implementações desenvolvidas.
- Capítulo 5 – *Conclusões*: Na conclusão do trabalho são relatados os principais tópicos observados durante a pesquisa e são apresentadas algumas sugestões de trabalhos futuros.

## **1.2. Objetivos**

O foco deste trabalho é a apresentação de um sistema de aquisição volumétrico com avanços em relação aos sistemas tradicionais de aquisição 3D baseados em triangulação laser. Para tanto fez-se necessário mostrar as etapas de desenvolvimento e os problemas enfrentados na implementação de um sistema de aquisição simplificado e de baixo custo, capaz de incentivar novas pesquisas nessa área pois, os custos praticados em sistemas comerciais semelhantes são injustificáveis.

### **1.3. Justificativa**

Os sistemas de aquisição 3D mais eficientes utilizam o princípio da triangulação laser ou o princípio do tempo de voo (TOF - *time of flight*). Nos sistemas baseados em triangulação, o alcance e a variação de profundidade são muito limitados, mas possuem grande precisão. Por outro lado, o oposto ocorre nos sistemas TOF, onde a precisão é baixa, e a variação de profundidade e o alcance são altos. Esse trabalho descreve o projeto de um sistema de aquisição 3D, baseado em triangulação laser com versatilidade similar aos sistemas TOF e com precisão e alcance superiores aos sistemas 3D tradicionais.

## Capítulo 2

# Revisão da Literatura

A reflexão teórica sobre o tema de visão computacional já produziu razoável bibliografia. O levantamento, aqui realizado, permitiu discutir várias técnicas de sensoriamento e uma breve abordagem sobre reconstrução. Quanto ao processamento de imagem, discutem-se álgebra de imagens, técnicas de pré-processamento e detecção de bordas, algumas aplicações de sensoriamento 3D e trabalhos relacionados a sistemas de aquisição 3D, usando diferentes tecnologias.

### 2.1. Sensores

Alguns sensores são dispositivos de hardware que fornecem dados explícitos de alcance usando iluminação ativa. Outros “sensores” são algoritmos de software que, passivamente, medem reflexões da cena ambiente e, subseqüentemente, derivam a distância. Se, por um lado, são necessários poucos cálculos para se estimar a distância usando sensores ativos, por outro lado, os sensores passivos requerem complexos algoritmos para o cálculo de distância [Sinha e Jain 1994].

#### 2.1.1. Sensores Ativos

O sensor ativo é geralmente definido como aquele que provê e controla sua própria iluminação. Irradia energia sobre as superfícies a serem medidas e computa a distância pela energia refletida. Tal medida pode ser feita por tempo de vôo do iluminador ou por triangulação de um padrão projetado. Uma variedade de fenômenos físicos pode ser usada como fonte radiante.

Os sensores discutidos, a seguir, usam o tempo de vôo de um feixe de laser, triangulação de um padrão de luz, interferência *Moiré* de padrões de luz branca e sonar ou feixes de radar.

## Tempo de Vôo

Sensores laser usam um feixe de luz focado e empregam o tempo de vôo (TOF – *Time of Flight*) ou diferença de fase para encontrar a distância. Os lasers de TOF medem o intervalo de tempo entre a emissão de um pulso laser de menos de um nanosegundo e a recepção do eco ou feixe refletido. O relacionamento entre o tempo e a distância é:

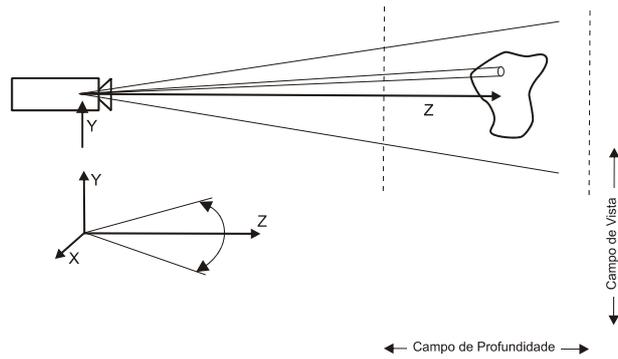
$$r = \frac{ct}{2} \quad (2.1)$$

onde  $r$  é a distância,  $c$  é a velocidade da luz, e  $t$  é o intervalo de tempo entre emissão e recepção da onda.

Sensores laser de diferença de fase usam um feixe laser com onda contínua e amplitude modulada, e medem a diferença de fase entre a onda emitida e a onda sensoriada. O relacionamento entre a diferença de fase  $\Delta\Phi$  e a distância  $r$  é dada por:

$$r = \left( \frac{c}{4\pi f_{am}} \right) \Delta\phi r \quad (2.2)$$

onde  $f_{am}$  é a frequência modulada. Desde que a diferença de fase pode ser medida até  $\lambda_{am}$ , a resolução de alcance de um sistema está restrita ao intervalo de ambigüidade  $\lambda_{am}/2$ . Um sensor laser consiste de uma fonte de laser montada em uma estrutura mecânica que permita que o feixe seja traçado na direção  $\theta$  e  $\Phi$  de um sistema de coordenada esférica virtual situado no sensor (Figura 2.1). As coordenadas obtidas devem ser convertidas por esse sistema de coordenada esférica para coordenadas cartesianas para derivar os dados de distância. Geralmente, faz-se a média das leituras de distância para remover o efeito do ruído resultante do efeito quantum termodinâmico ou a trepidação na estrutura eletromecânica do laser.



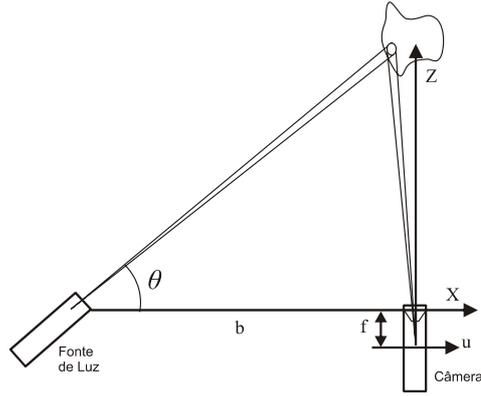
**Figura 2.1:** Típica configuração de sensor laser.

Três aspectos importantes devem ser considerados no uso de sensores laser. O primeiro é a potência do sensor. Lasers de alta potência causam um sinal refletido mais forte e, por isso, melhor alcance. No entanto, esses lasers são caros e mais prejudiciais à vista desprotegida, o que pode limitar seu uso em ambientes de vivência humana. O segundo aspecto é o volume de vista. O volume de vista de um sensor é determinado pela área de projeção e o campo de profundidade. A área de projeção é controlada pelo campo de vista dos espelhos de varredura. O terceiro aspecto é a resolução de distância. Tipicamente, um sensor de distância produz 8-12 bits de informação e uma resolução de 25-500  $\mu\text{m}$  [Sinha e Jain 1994].

O problema com sensores laser (e sensores ativos em geral) é que eles dependem da reflexão da superfície do objeto. A energia refletida é determinada pelo ângulo de incidência do feixe e as propriedades reflexivas do material do objeto. Leituras imprecisas acontecem quando a superfície do objeto é tangente à direção do feixe e o ângulo de incidência é  $0^\circ$ . As medições também são imprecisas para superfícies especulares, pelo modo como as superfícies brilhantes espalham a luz. Objetos com material transparente como vidro, obviamente, não podem ser medidos com lasers ópticos.

### **Luz Estruturada**

Sistemas de luz estruturada são sensores ativos baseados em triangulação. Uma luz na forma de ponto, malha, ou grade é projetada sobre o objeto, e essa forma é capturada por uma câmera. A distância até o objeto é calculada por trigonometria, dado um conhecimento a priori das posições da fonte iluminante e do detector. Uma configuração é mostrada na Figura 2.2.



**Figura 2.2:** Típica geometria de luz estruturada.

A câmera possui comprimento focal  $f$  e tem suas lentes localizadas na origem. Um projetor é configurado de forma que emita o padrão estruturado em um ângulo  $\theta$  em relação à linha base. A distância entre o projetor e a câmera é a linha de base  $b$ . Um ponto  $(X, Y, Z)$  no espaço real é projetado na câmera no pixel  $(u, v)$ , de forma que:

$$u = \frac{Xf}{Z} \quad (2.3)$$

$$v = \frac{Yf}{Z} \quad (2.4)$$

As coordenadas reais dos pontos na luz estruturada são dados por:

$$X = \frac{bu}{f \cot \theta - u} \quad (2.5)$$

$$Y = \frac{bv}{f \cot \theta - u} \quad (2.6)$$

$$Z = \frac{bf}{f \cot \theta - u} \quad (2.7)$$

Os valores  $f$ ,  $b$ ,  $\theta$  são conhecidos pela configuração do equipamento e a localização dos *pixels*  $u, v$  são encontradas por detecção de pico percorrendo a linha de varredura da imagem. A vantagem dos sensores de luz estruturada é o custo. No entanto, eles

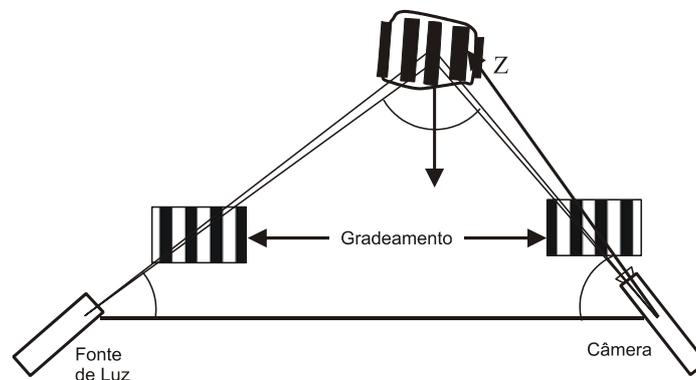
não são muito precisos, desde que a resolução do sistema é limitada pela resolução do *pixel* de medida  $u, v$ . Uma outra desvantagem dos sensores de luz estruturada é o problema das partes perdidas. A fonte de luz e a câmera não são coaxiais, e algumas áreas do padrão de projeção podem ser escondidas por outras superfícies que fazem parte do próprio objeto. Uma melhor resolução pode ser alcançada pelo sensor aumentando-se a linha de base  $b$ . Entretanto, o problema de oclusão aumenta. Uma possível solução para esse problema é uma configuração com múltiplas câmeras e integrar a informação de suas fontes.

### Interferometria *Moiré*

Sensores *Moiré* são pares emissor-detector de luz onde uma grade é colocada na frente do projetor e outra na câmera. Uma configuração é mostrada na Figura 2.3. Um padrão de interferência é criado quando a luz é processada pelas duas grades que têm formas regularmente espaçadas. Se  $p_0$  é o período das franjas na superfície do objeto, e  $\theta_l, \theta_v$  os ângulos das fontes de luz e do observador, então a mudança relativa na distância é dada por:

$$\Delta z \approx \frac{p_0}{\tan(\theta_l) + \tan(\theta_v)} \quad (2.8)$$

O sensor de projeção *Moiré* é usado para medir a profundidade relativa de uma superfície suave.



**Figura 2.3:** Configuração de sensor *Moiré*.

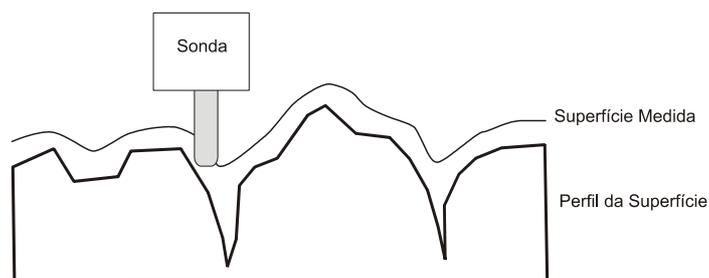
## Sensoriamento Tátil

Um dos modos mais precisos de se medir coordenadas tridimensionais de um ponto é realmente tocá-lo, usando uma sonda. Existem duas sondas táteis muito dissimilares: os microscópios de sonda de varredura (S x M – *scanning probe microscopes*) e as máquinas de medição de coordenadas (CMM – *coordinate measuring machines*).

### SxM

Microscópios de sonda de varredura, que incluem técnicas de microscopia como a STM (*Scanning Tunneling Microscopy*) e SEM (*Scanning Electron Microscopy*) (a razão pela qual o “x” está no título), são ferramentas valiosas para medir estruturas em nível atômico. Eles permitem a análise de estrutura de superfície menor que 0.1nm e têm grande potencial de aplicação em ciência de materiais e fabricação de semicondutores. São empregados como dispositivos de avaliação de distância para detectar falhas nos wafers de semicondutores. Há também um grande escopo em biologia molecular.

O SxM opera com base no princípio do fenômeno quantum-mecânico. Uma sonda muito pequena ou *stylus* é colocada em posição de quase contato com a superfície a ser mediada. Em distâncias tão próximas, interações quantum mecânicas ocorrem entre a sonda e a superfície. A distância pode ser medida calculando esse fenômeno. Técnicas de análise de imagem podem ser utilizadas para extrair informações das imagens produzidas de alcance denso quando a sonda percorre a superfície. Uma desvantagem para sensores SxM é que as leituras são dependentes do tamanho da sonda utilizada, como ilustra a Figura 2.4.



**Figura 2.4:** O efeito do tamanho da sonda nas medidas de distância.

Nos sistemas STMs, o fenômeno de interação é um tunelamento de corrente que segue entre a sonda e a superfície, quando são colocados quase juntos. A corrente de tunelamento varia inversamente com a distância entre a sonda e a superfície.

$$i_t = \frac{e^2 k_0 V A_{\text{eff}} e^{-2k_0 Z}}{4\pi h Z}, \quad (2.9)$$

onde  $e$  é a carga do elétron,  $V$  é a inclinação (bias) de voltagem entre a sonda e a superfície,  $A_{\text{eff}}$  é a área efetiva pela qual a corrente de tunelamento segue, e  $Z$  é a distância entre a sonda e a superfície;  $k_0$  é dado por:

$$k_0 = \sqrt{\frac{2m_e \phi}{h^2}}, \quad (2.10)$$

onde  $m_e$  é a massa do elétron e  $\phi$  é a função de trabalho da amostra. A corrente de tunelamento é realimentada para o mecanismo de posicionamento manter a sonda a uma distância constante da superfície.

## CMM

Uma máquina de medição de coordenadas consiste num braço mecânico preciso com uma sonda tátil ou sensor laser preso em seu pulso. O braço é um ligamento mecânico de altíssima tolerância provendo leituras precisas mesmo em grandes volumes. Os CMMs são vastamente usados na indústria para medição e verificação de partes manufaturadas. A grande vantagem do CMM é sua precisão. Um típico CMM tem uma precisão na ordem de 10 $\mu$ m. Uma desvantagem é que o dispositivo é movido manualmente para os pontos na superfície ou programado para seguir um caminho particular. Em ambos os casos o processo é trabalhoso e demorado. Uma outra desvantagem do CMM é o custo do maquinário de alta precisão envolvido.

## Sonar

Sonares são sensores de distância que usam ondas ultrasônicas como “iluminação” ativa. A distância do objeto pode ser calculada pela diferença de fase da onda retornada com a onda emitida. Sonares são empregados como sensores de distância na

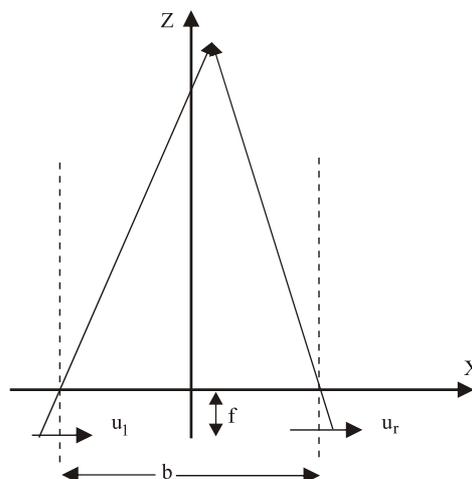
navegação de robôs para evitar obstáculos próximos. Portanto eles não podem ser considerados realmente como sensores “geométricos”, porque sua saída é muito ruidosa.

### 2.1.2. Sensores Passivos

Sensores passivos, como o nome diz, meramente absorvem a radiação ambiente e deriva a distância, primeiramente, extraindo uma representação intermediária. A maioria das técnicas de sensoriamento passivo caem no título do método SfX (Shape-from- $X$ ), onde  $X$  pode ser estéreo, movimento, foco, ou similar. Enquanto muita energia foi gasta no desenvolvimento de sistemas de medida de distância passiva, a maior parte desses sistemas permanecem no mundo acadêmico.

#### Estéreo

Forma-por-estereoscopia é um velho e bem estudado problema, em visão computacional. Os princípios por trás da visão estereoscópica são similares àqueles da triangulação. A distância é derivada da disparidade entre as imagens formadas em câmeras que estão observando a mesma cena, mas posicionadas separadamente. A Figura 2.5 mostra a configuração idealizada para um sistema estéreo binocular onde as câmeras são colocadas em paralelo.



**Figura 2.5:** Configuração estéreo paralelo.

A geometria da câmera é normalmente configurada de forma que uma possa combinar parâmetros por uma linha epipolar (i.e., a mesma linha na imagem da direita e esquerda). Isso faz a busca por características correspondentes uma procura linear pela linha de varredura do par de imagens. Desde que as câmeras sejam colocadas separadas com uma linha de base de separação, as imagens formadas nas retinas ou câmeras são levemente diferentes em seus valores de  $u$ , mas um parâmetro pode ser encontrado na mesma linha de varredura  $v$  em cada imagem. A disparidade  $d$  entre as duas imagens é definida como a distância entre projeções  $(u_l, v)$  e  $(u_r, v)$  do mesmo ponto da cena nas duas câmeras. A disparidade, junto com o conhecimento da linha de base, pode ser usada para calcular a distância até o objeto da cena com o simples relacionamento:

$$d(u_l, u_r) = \frac{bf}{Z} \quad (2.11)$$

$$X = \frac{b(u_l + u_r)}{2d} \quad (2.12)$$

$$Y = \frac{bv}{d} \quad (2.13)$$

$$Z = \frac{bf}{d}, \quad (2.14)$$

onde  $(u_l, v)$  e  $(u_r, v)$  é a projeção de perspectiva do ponto  $(X, Y, Z)$  em um par de câmeras de comprimento focal  $f$  separado por uma linha de base  $b$ .

A disparidade pode ser calculada fazendo a correlação da área local entre regiões nas imagens da esquerda e direita ou combinação de parâmetros distintos nas imagens. O método de correlação de área produz um denso mapa de disparidade, mas é menos preciso que a combinação de parâmetros. Por outro lado, a combinação de parâmetros nas imagens da esquerda e direita resulta em um mapa de profundidade esparsa, que deve ser interpolado para ter a informação completa. Os problemas com o sensor passivo de distância estéreo são: 1) estabelecer a correspondência entre as características das imagens da esquerda e direita é complexo devido às características de oclusão quando vistos de diferentes posições; 2) a profundidade só pode ser

computada em pontos discretos onde a correspondência tenha sido estabelecida, e para produzir um mapa de profundidade denso, os valores de profundidade (ou disparidade) encontrados devem ser interpolados. Dessa forma, a precisão da distância depende tanto da correspondência de características quanto da interpolação.

## Movimento

Forma-por-movimento possui uma grande área de pesquisa e inclui dois subtópicos básicos: estrutura de fluxo óptico instantâneo e estrutura de seqüência de imagens. Fluxo óptico é o campo de velocidade instantânea produzida na retina de um sistema pelo “ego-motion” do observador. Pesquisadores vêm tentando extrair fluxo óptico de uma seqüência de imagens usando regularização, e então extraem parâmetros de estrutura do fluxo óptico. A geometria do vetor da estrutura por fluxo óptico é elegante, porém complexa. A debilidade dessas abordagens está na suposição que o campo do fluxo óptico é conhecido. A estimação robusta do fluxo óptico tem iludido os pesquisadores por muitos anos.

As equações básicas envolvendo a distância  $Z$  e o fluxo óptico  $(\dot{u}, \dot{v})$  são:

$$\dot{u} = \frac{V_x - xV_z}{Z} - \omega_x xy + \omega_y(1 + x^2) - \omega_z y \quad (2.15)$$

$$\dot{v} = \frac{V_y - yV_z}{Z} - \omega_x(1 + y^2) + \omega_y xy - \omega_z x \quad (2.16)$$

onde um ponto  $(X, Y, Z)$  movendo no espaço com velocidade  $(V_x, V_y, V_z)$  é imageado em  $x, y$  com um fluxo óptico de  $(\dot{u}, \dot{v})$ .

Pesquisadores também vêm tentando extrair estrutura de uma seqüência de imagens temporais. Esse problema é similar ao de forma por estereoscopia. Pares de imagens de tempo adjacente de uma forma seqüencial (temporal) de pares estéreo. O conhecimento da velocidade do sistema de câmera é equivalente ao conhecimento da linha base de um sistema estéreo. No caso mais simples de um sistema de forma-por-movimento, os eixos da câmera são alinhados na direção do movimento. A disparidade (temporal) entre parâmetros em quadros (frames) consecutivos de uma

seqüência de imagens é encontrada por técnicas correspondentes. A distância pode ser encontrada por trigonometria.

## Foco

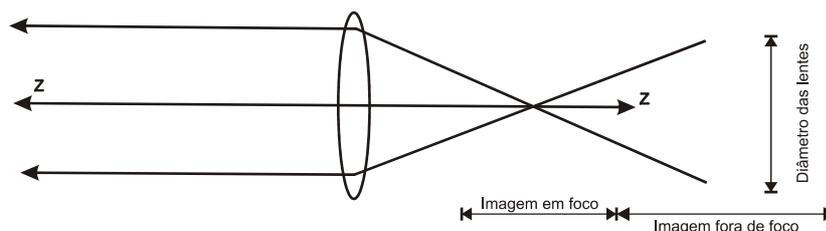
Forma-por-foco é a recuperação da informação de distância de uma imagem de acordo com os objetos de interesse, que podem estar dentro ou fora do foco. Objetos que permanecem dentro de um campo de profundidade finito de uma lente convexa estão em foco no plano de imageamento. O princípio básico das lentes delgadas é um simples relacionamento entre objetos em foco e sua distância. Lentes delgadas de comprimento focal  $f$  irão focar um ponto fonte localizado a uma distância  $Z$  sobre um plano focal localizado a uma distância  $z$ , ou:

$$\frac{1}{Z} + \frac{1}{z} = \frac{1}{f} \quad (2.17)$$

como ilustrado na Figura 2.6. Variando  $z$  e tendo um método para encontrar a imagem em foco, o  $z$  apropriado pode ser lido fora e a distância  $Z$  pode ser computada. Dado que uma lente real vai sofrer borramento ocasionado pela difração a equação deve ser modificada para corrigir esse problema.

$$Z_{\pm} = \frac{zf}{z - f \pm \sigma f / D}, \quad (2.18)$$

onde  $D$  é o diâmetro da lente. A distancia sinalizada por  $Z_+ - Z_-$  é a profundidade do campo, a região onde todos os pontos estão em foco.



**Figura 2.6:** Configuração de profundidade-por-foco.

## 2.2. Reconstrução

Geralmente os sensores dos sistemas de aquisição 3D produzem como saída uma nuvem de pontos ou imagem de profundidade. A nuvem de pontos é um conjunto de coordenadas de vértices e a imagem de profundidade é uma matriz 2D onde os valores de intensidade de cada pixel representam a distância entre o objeto alvo e o sensor. Para que essa informação possa ser visualizada, modelos de superfície devem ser criados a partir de dados. Como o scanner 3D só pode fazer a aquisição de dados onde o sensor “vê” o alvo, reconstruir objetos inteiros ou cenas, geralmente, envolve a integração de várias aquisições feitas em diferentes pontos de vista. O problema de interesse é então, construir uma única representação de superfície para essas múltiplas aquisições.

### 2.2.1. Superfícies Poligonais

A representação de borda mais comum usada para um objeto gráfico tridimensional é um conjunto de polígonos de superfície que fecham o interior do objeto. Isso simplifica e acelera a renderização de superfície e exibição do objeto, já que todas as superfícies são descritas com equações lineares. Por essa razão, descrições de polígonos são, freqüentemente, referenciadas como "objetos gráficos padrão". Em alguns casos, uma representação poligonal é a única disponível [Herm e Baker 1997].

Uma representação poligonal para um poliedro define precisamente as características de superfície do objeto. Mas para outros objetos, as superfícies são *pavimentações* para produzir aproximação de malhas de polígonos *polygon-mesh*. Na Figura 2.7, a superfície de um cilindro é representada como uma malha de polígonos. Essas representações são comuns em desenho e aplicações de modelagem de sólidos, já que o desenho em modelo de arame (*wireframe*) pode ser mostrado rapidamente para dar uma indicação geral da estrutura da superfície.

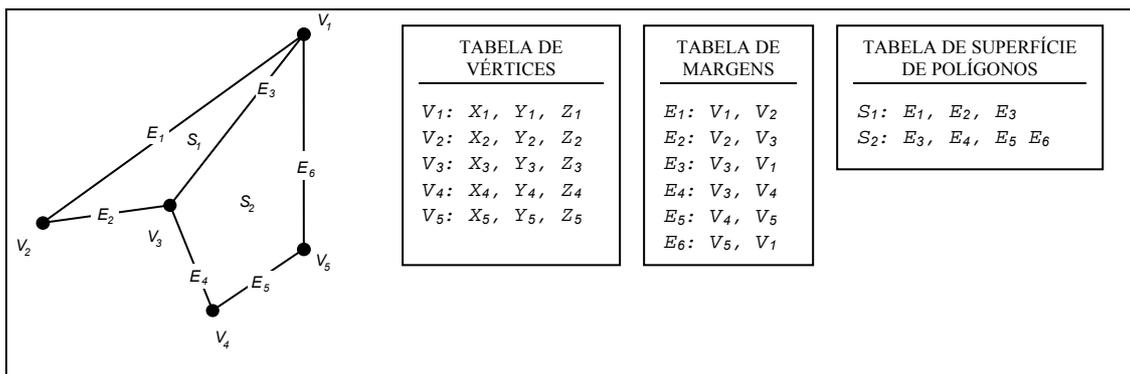


**Figura 2.7:** Representação em modelo de arame de um cilindro.

## Tabela de Polígonos

Uma superfície poligonal é especificada como um conjunto de coordenadas de vértices e parâmetros de atributos associados. Quando a informação para cada polígono é introduzida, os dados são colocados em tabelas que serão usadas, posteriormente, em processamento, exibição e manipulação dos objetos na cena. As tabelas de dados de polígonos podem ser organizadas em dois grupos: tabelas geométricas e tabelas de atributos. As tabelas de dados geométricas contêm coordenadas de vértices e parâmetros para identificar a orientação espacial das superfícies dos polígonos. As informações de atributos para um objeto incluem parâmetros especificando o grau de transparência do objeto, reflexão e características de textura.

Uma organização conveniente para armazenar dados geométricos é criar listas de árvores: uma tabela de vértices, uma tabela de margens e uma tabela de polígonos. Valores de coordenadas para cada vértice do objeto são armazenados na tabela de vértices. A tabela de margens contém ponteiros para a tabela de vértices, que são usados para identificar os vértices de cada margem do polígono. E a tabela de polígonos contém ponteiros para a tabela de margens, que identificam as margens de cada polígono. Esse esquema é ilustrado na Figura 2.8 para dois polígonos adjacentes em uma superfície de objeto.



**Figura 2.8:** Representação de tabela de dados geométricos para duas superfícies de polígonos adjacentes, formados por seis margens e cinco vértices.

Os dados são listados em três tabelas, como ilustrado na Figura 2.8, para fornecer uma referência conveniente para os componentes individuais (vértices, margens e polígonos) de cada objeto. O objeto também pode ser mostrado de maneira eficiente usando dados da tabela de margem para desenhar as linhas componentes. Um modo alternativo de disposição é usar somente duas tabelas: uma tabela de vértices e uma tabela de polígonos. Mas esse esquema é menos conveniente, e algumas margens podem ser desenhadas duas vezes. Uma outra possibilidade é usar somente a tabela de polígonos, mas isso duplicaria as informações de coordenadas, já que valores explícitos de coordenadas são listados para cada vértice em cada polígono. A informação de margem também teria que ser reconstruída pela listagem de vértices na tabela de polígonos.

É possível adicionar informações extra, nas tabelas de dados da Figura 2.8, para fazer a extração de informação mais rápida. Por exemplo, poderia-se expandir a tabela de margens para incluir ponteiros para a tabela de polígonos, desse modo, margens comuns entre polígonos poderiam ser identificadas de forma mais rápida (Figura 2.9). Similarmente, a tabela de vértices poderia ser expandida para que os vértices pudessem ser referenciados, em cruz, com as correspondentes margens.

$E_1$	$V_1, V_2, S_1$
$E_2$	$V_2, V_3, S_1$
$E_3$	$V_3, V_1, S_1, S_2$
$E_4$	$V_3, V_4, S_2$
$E_5$	$V_4, V_5, S_2$
$E_6$	$V_5, V_1, S_2$

**Figura 2.9:** Tabela de margens para as superfícies da Figura 2.8 expandida para incluir ponteiros para a tabela de polígonos.

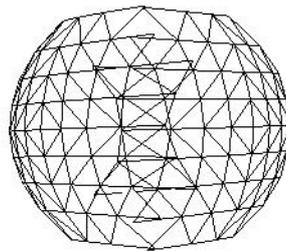
A informação geométrica adicional que normalmente é armazenada em tabela de dados inclui a inclinação para cada margem e a extensão das coordenadas para cada polígono. Como os vértices são a entrada, é possível calcular a inclinação das margens e varrer os valores das coordenadas para identificar os valores mínimo e máximo de  $x$ ,  $y$  e  $z$  para polígonos individuais.

Visto que as tabelas de dados geométricos podem conter extensiva listagem de vértices e margens para objetos complexos, é importante que os dados sejam checados quanto a consistência e integridade. Quando as definições de vértice, margem e polígono são especificadas, é possível, particularmente em aplicações interativas, que

certos erros de entrada possam ocorrer e causar distorção na exibição do objeto. Quanto mais informação houver na tabela de dados, mais fácil é fazer a checagem de erros. Assim, a checagem de erros é mais fácil quando três tabelas de dados (vértice, margem e polígono) são usadas, uma vez que esse esquema fornece mais informação.

### 2.2.2. Remoção de Triângulos nas Regiões de Sobreposição

Se dois objetos são gerados por duas aquisições em pontos de vista distintos, é necessário determinar as regiões de sobreposição. A Figura 2.10 apresenta dois objetos de duas aquisições virtuais de uma esfera. Nela os triângulos redundantes no centro mostram a região de sobreposição.

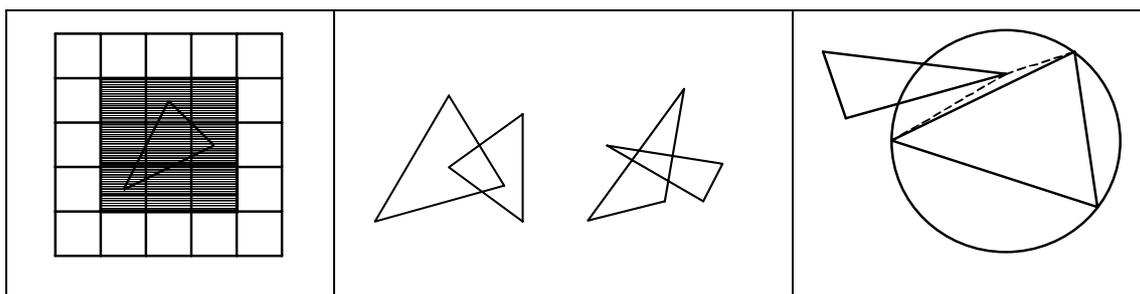


**Figura 2.10:** Dois registros de aquisição.

A detecção de região de sobreposição é baseada na retroprojeção. Conhecendo-se o modelo de calibração, os pontos 3D podem ser projetados de volta para um quadro de referência 2D. Dada uma nova malha triangular (*mesh*), cada triângulo da rede antiga é projetado em um novo quadro de referência 2D [Sun et al. 2000] (ex.: o plano de imagem da nova aquisição). Se a projeção está fora desse quadro de referência, ela não está no *view port* da nova aquisição e o triângulo em questão não será alterado. Se a projeção estiver dentro do novo quadro de referência, é feita uma checagem de sobreposição. Primeiro, uma caixa delimitadora da projeção do triângulo é calculada como ilustra a Figura 2.11. Então, é checado se o triângulo está de frente para a nova posição de aquisição ou não. Se o produto do ponto da normal do triângulo com um dos três raios medidos (ex.: os raios do ponto de vista até cada vértice do triângulo) for positivo, diz-se que ele está “de frente”. Para os triângulos de frente, serão checados todos os triângulos do novo objeto de aquisição que estão na caixa delimitadora. A Figura 2.12, mostra as condições de intersecção de triângulo 2D. A

checagem de cada par permite identificar as margens em intersecção, embora, essa operação tenha um custo computacional alto.

Na maioria dos casos de intersecção, um ponto de um triângulo estará dentro do outro triângulo. Essa observação é de custo computacional baixo. Por essa razão, esse caso é verificado primeiro e, se esse caso não for satisfeito, a verificação de intersecção de margem é verificada. Note-se que existem algoritmos eficientes para verificar intersecção de linha 2D [O'Rourke 1998].



**Figura 2.11:** Caixa de delimitadora.

**Figura 2.12:** Triângulos em intersecção

**Figura 2.13:** Círculo de remoção

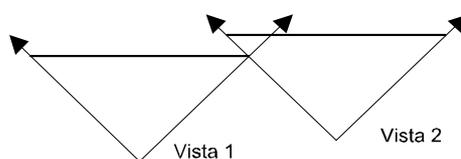
Quando se checa se um ponto está dentro de um triângulo, emprega-se um círculo circunscrito de “remoção” (Figura 2.13). Se um ponto estiver posicionado levemente fora do triângulo, o triângulo que seria criado estaria mal formado, como mostra a Figura 2.13, em linhas tracejadas. É por esse motivo que se verifica se aquele ponto está dentro do círculo de remoção. Nesse caso, os dois triângulos serão classificados como sobrepostos.

Quando todas as caixas delimitadoras estiverem checadas e existirem regiões sobrepostas, deve-se remover ou o triângulo na malha de superfície antiga ou todos os triângulos sobrepostos na nova malha de superfície. Para manter as melhores medições, é computada uma confiança para cada triângulo. A confiança é definida como o produto do ponto para a normal do triângulo e o raio medido, ambos normalizados, e terá o valor entre  $[-1, 1]$ . A confiança média é calculada para todos os triângulos sobrepostos na caixa delimitadora. Se essa média for maior que a do triângulo da malha triangular antiga, o triângulo é removido na malha triangular antiga. Caso contrário, todos os triângulos sobrepostos na caixa delimitadora serão removidos. Note-se que sobreposição em 2D não implica sobreposição em 3D. Em 3D, dois remendos (*patches*) de superfície sobrepostos em sua projeção 2D podem vir de diferentes áreas do objeto se, por exemplo, o objeto tiver algumas auto-ocluções. Um limiar é escolhido para determinar quando dois remendos sobrepostos em 2D são

da mesma área do objeto. Se a distância entre os dois triângulos for menor que o limiar, assume-se que eles são representações do mesmo remendo de superfície. O limiar é escolhido de acordo com a precisão do sistema de aquisição e a distância medida.

Como sempre, há algum erro de registro e ruído nos dados de aquisição, pois, raramente, os remendos de superfície são alinhados perfeitamente. Enquanto por um ponto de vista os triângulos podem estar sobrepostos, por outro eles podem não estar. Isso é mostrado na Figura 2.14. Como a detecção de sobreposição depende do ponto de vista, não se deve checar apenas na *viewport* atual, mas também checar na *viewport(s)* antiga.

Isso explica o motivo pelo qual os triângulos, na malha triangular antiga, podem vir de muitos pontos de vista diferentes, previamente integrados, retroprojetando cada triângulo da malha de superfície nova em cada *viewport* anterior. Essa operação tem alto custo computacional. Ao invés disso, só os triângulos na caixa delimitadora são projetados no quadro de referência 2D do triângulo da malha triangular velha. Novamente a remoção também é baseada na medida de confiança média.



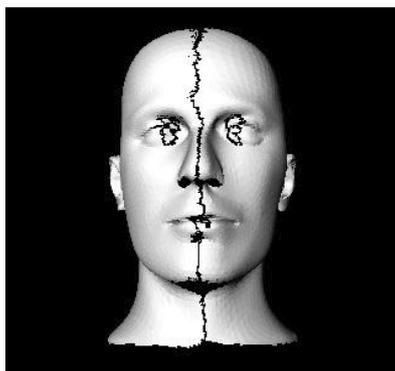
**Figura 2.14:** Sobreposição dependente do ponto de vista. Pela vista 1 há sobreposição, mas pela vista 2 não.

De maneira geral, os triângulos na malha triangular velha que não estão de frente, não precisam fazer verificação de sobreposição. No entanto, existe um caso específico que deve ser considerado, como mostra a Figura 2.15. Quando o passo de descontinuidade for menor que o limiar, dois pontos juntos com uma descontinuidade são conectados. Mas, quando a superfície real é medida, pode ser necessário remover tal remendo de superfície. Por exemplo, na Figura 2.15(a) dois pontos de vista diferentes são indicados. A linha tracejada mostrada em (b), criada a partir da vista 2, mas não está correta, como na vista 1. Assim, o triângulo indicado pela linha tracejada deveria ser removido.



**Figura 2.15:** Não "olhando para frente", mas deve verificar sobreposição. (a) Duas vistas, (b) Resultados da amostragem de 2 vistas.

A Figura 2.16, mostra uma imagem de duas vistas do modelo de uma cabeça depois da remoção de sobreposições. Pode ser visto que toda região de sobreposição foi removida e uma lacuna foi deixada no centro da face e nos olhos. As medições mais confidentes são mantidas. Nessa implementação, para calcular a confiança, também é contada a distância entre o objeto e o sistema de aquisição. As medições mais próximas tendem a ser mais precisas mesmo quando o feixe não toca a superfície perpendicularmente. É adicionado um fator no calculo da confiança para compensar a diferença de distância entre um par de vistas.



**Figura 2.16:** Duas vistas depois de remoção de triângulos sobrepostos

### 2.2.3. Unindo os Patches de Redes de Superfície

Para unir as lacunas entre os remendos de superfície, deve-se nomear um triângulo candidato para ser combinado com outros pontos para construir novos triângulos. Esses triângulos candidatos são chamados "triângulos ativos" e eles devem estar na fronteiras da malha de superfície. Note-se que nem todos os triângulos da fronteira são triângulos ativos, porque alguns podem não ter relação com a outra malha de superfície. Se qualquer um dos vizinhos do triângulo for removido, ele será marcado como triângulo ativo, se ele estiver na malha de superfície nova ou velha. Quando um malha de superfície é criada por um único ponto de vista, a informação dos triângulos

vizinhos é armazenada. Por exemplo, para cada ponto do triângulo, armazena-se um ponteiro para o triângulo vizinho oposto. Se o ponteiro for nulo, o triângulo está na fronteira da malha de superfície. O ponteiro será atualizado quando um triângulo vizinho for removido ou um novo triângulo for criado para atravessar a lacuna.

Um triângulo ativo pode ter uma, duas, ou mesmo três “margens ativas” que devem encontrar um ponto para construir um novo triângulo. Para uma margem ativa, primeiro encontram-se alguns pontos vizinhos como candidatos. Se a margem ativa e os pontos mais próximos estiverem ambos na malha de superfície nova ou velha, os pontos mais próximos não estão necessariamente na margem ativa. Então, verifica-se a validade de cada candidato e encontra-se qual é a melhor. Em seguida, examina-se a intersecção do triângulo novo com os triângulos existentes em uma região. Para todos os pontos candidatos válidos, aquele que estiver de frente para a margem ativa com o maior ângulo é o melhor. Esse ponto é usado para criar um novo triângulo. Se o triângulo tiver uma margem comum com qualquer triângulo existente, os dois triângulos irão atualizar a informação de seus vizinhos. Depois de fazer a união com todas as lacunas, obtém-se uma representação global da malha de superfície. Para a procura do ponto candidato, emprega-se uma Kd-tree [Mount].

## **2.3. Processamento de Imagens**

O sensoriamento utilizando o método de luz estruturada baseado em triangulação utiliza imagens para captar a projeção de um padrão de luz, que deve ser segmentado da imagem. O primeiro passo dessa segmentação utiliza operações algébricas de imagens, mas os resultados são ruidosos e necessitam de uma etapa de pré-processamento, para suavizar a imagem filtrando os ruídos e para realçar o a projeção de luz na imagem, são utilizadas técnicas de detecção de bordas.

### **2.3.1. Álgebra de Imagem**

Existem duas categorias primárias de operações algébricas aplicadas a imagens: aritmética e lógica. Adição, subtração, divisão e multiplicação compreendem as operações aritméticas, enquanto que os operadores lógicos são: AND, OR e NOT

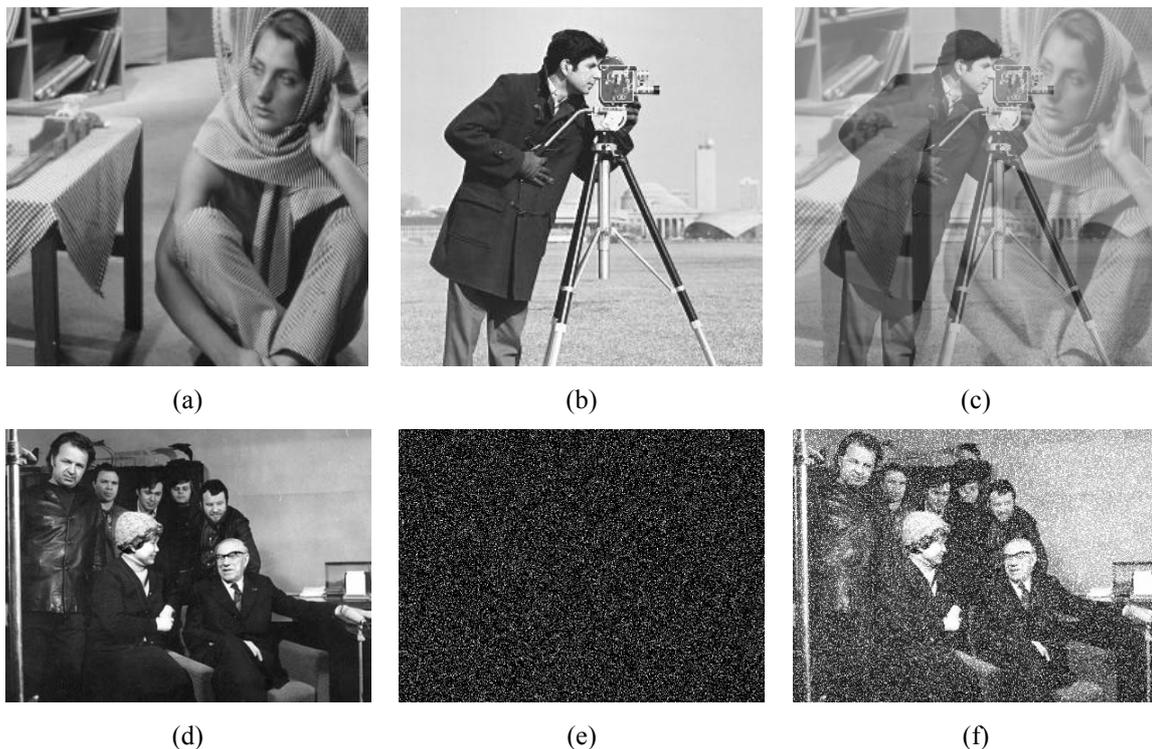
[Umbaugh 1998]. Essas operações são feitas em duas imagens, exceto para a operação lógica NOT que requer somente uma imagem e é feito pixel-a-pixel.

Para aplicar operações aritméticas em duas imagens, simplesmente se opera nos valores correspondentes de pixel. Por exemplo, para adicionar imagens  $I_1$  e  $I_2$  e gerar  $I_3$ :

$$I_1(r, c) + I_2(r, c) = I_3(r, c) \quad (2.19)$$

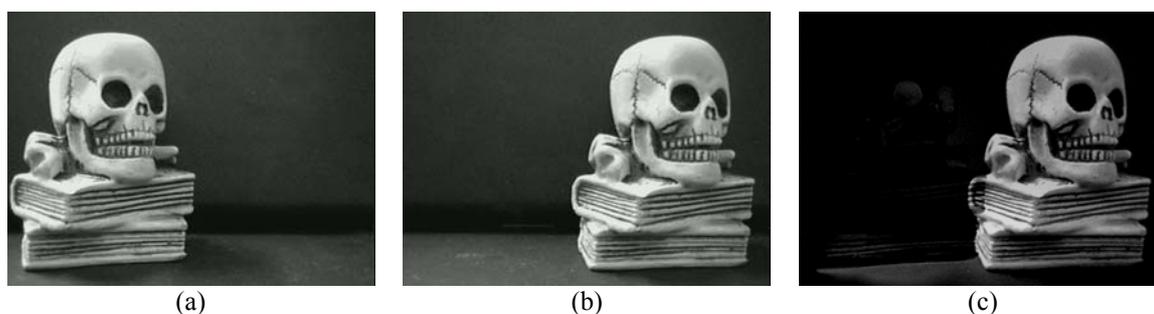
$$I_1 = \begin{bmatrix} 3 & 4 & 7 \\ 3 & 4 & 5 \\ 2 & 4 & 6 \end{bmatrix} \quad I_2 = \begin{bmatrix} 6 & 6 & 6 \\ 4 & 2 & 6 \\ 3 & 5 & 5 \end{bmatrix} \quad I_3 = \begin{bmatrix} 3+6 & 4+6 & 7+6 \\ 3+4 & 4+2 & 5+6 \\ 2+3 & 4+5 & 6+5 \end{bmatrix} = \begin{bmatrix} 9 & 10 & 13 \\ 7 & 6 & 11 \\ 5 & 9 & 11 \end{bmatrix}$$

A adição é utilizada para combinar informação de duas imagens. As aplicações da adição incluem o desenvolvimento de algoritmos de restauração de imagens para modelar ruído adaptativo, e efeito de transição de vídeo (Figura 2.17).



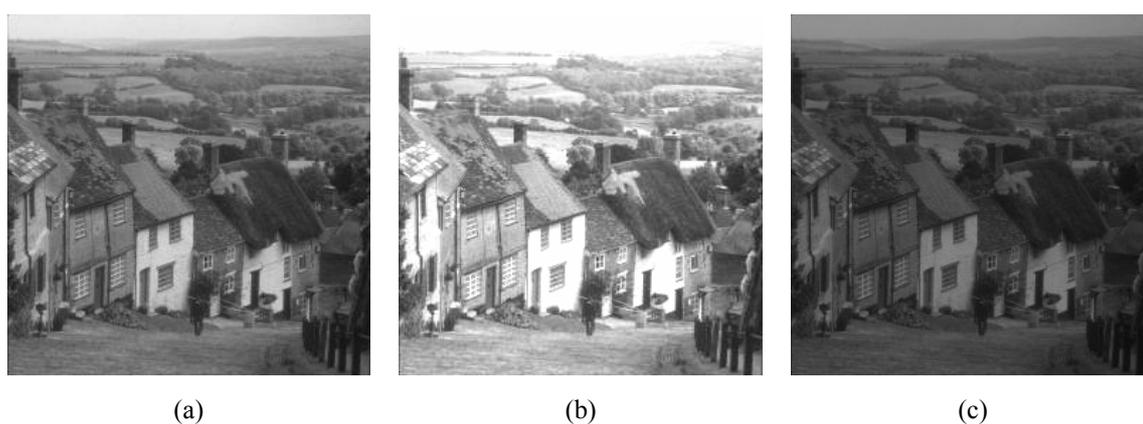
**Figura 2.17:** a) primeira imagem original; b) segunda imagem original; c) adição das imagens (a) e (b). d) imagem original; e) ruído gaussiano; f) adição das imagens (d) e (e).

A subtração de duas imagens é frequentemente utilizada para detecção de movimento. Considere o caso onde nada mudou em uma cena; o resultado da subtração de duas imagens seqüenciais é preenchido com zeros, uma imagem preta. Se algo se move na cena, a subtração produz resultados diferentes de zero no local de movimento. A Figura 2.18 mostra o uso da subtração em detecção de movimento.



**Figura 2.18:** a) cena original; b) mesma cena posteriormente; c) subtração da (b) e (a), note que somente objeto que se mexeu na cena aparece no resultado.

Multiplicação e divisão são utilizadas para ajuste de brilho em uma imagem. Uma imagem tipicamente consiste de valores constantes maiores que um. A multiplicação dos valores dos pixels por um número maior que um aumenta o brilho da imagem e a divisão por um valor maior que um escurece a imagem. O ajuste de brilho é frequentemente utilizado como etapa de pré-processamento em realce de imagem como mostra a Figura 2.19.



**Figura 2.19:** a) imagem original; b) imagem multiplicada por 1.6; c) imagem dividida por 1.6.

As operações lógicas AND, OR e NOT formam um conjunto completo, isto é, qualquer outra operação lógica (XOR, NOR, NAND) pode ser criada combinando esses elementos básicos.

Executando um AND lógico em duas imagens, com dois pixels correspondentes de valor  $111_{10} = 01101111_2$  em uma imagem e  $88_{10} = 01011000_2$  na outra, tem-se:

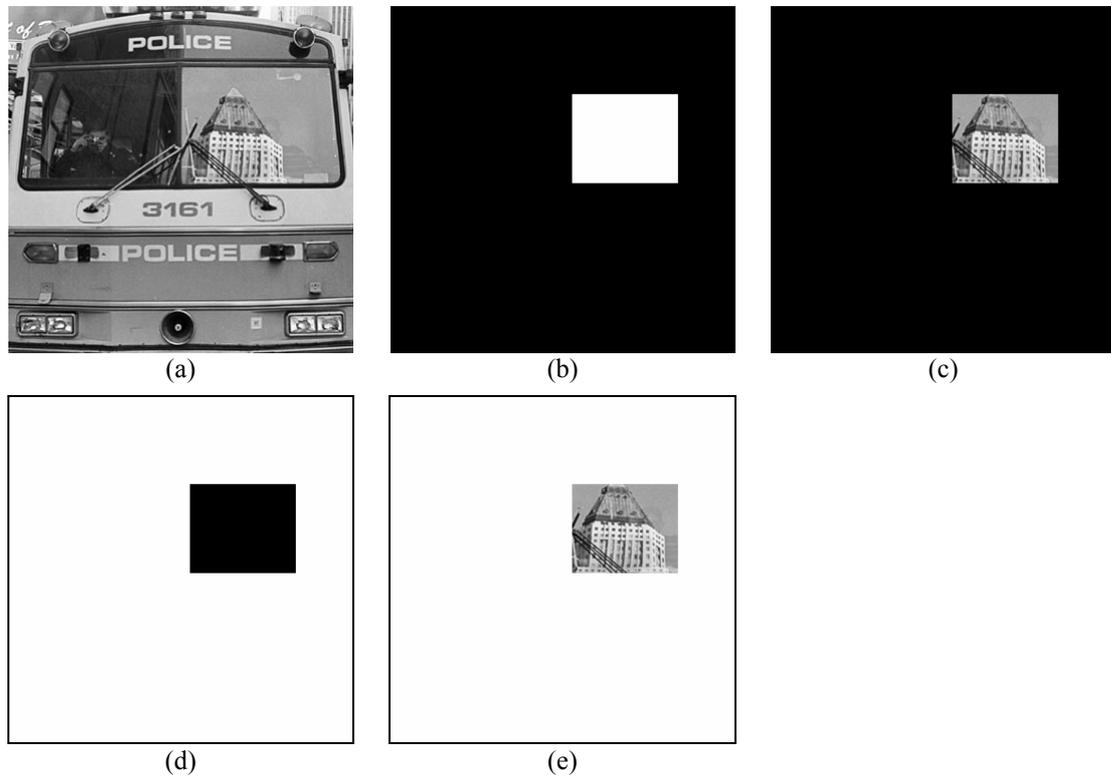
$$\begin{array}{r} 01101111_2 \\ \text{AND } 01011000_2 \\ \hline 01001000_2 \end{array}$$

Os operadores lógicos AND e OR são utilizados para combinar informações em duas imagens. Essas ferramentas são utilizadas em análise de imagem para operação de mascaramento. Se um método mais sofisticado não estiver disponível, AND e OR podem ser utilizados como um método simples de extração de uma região de interesse na imagem. Por exemplo, um retângulo branco sendo aplicado como AND em uma imagem permite que somente a região coincidente ao retângulo apareça na imagem de saída, com fundo preto; e um retângulo preto sendo aplicado como OR em uma imagem permite que somente a parte correspondente na imagem ao retângulo preto apareça na imagem de saída, mas com o fundo da imagem em branco. Esse processo é chamado de mascaramento de imagem e a Figura 2.20 ilustra os resultados dessas operações.

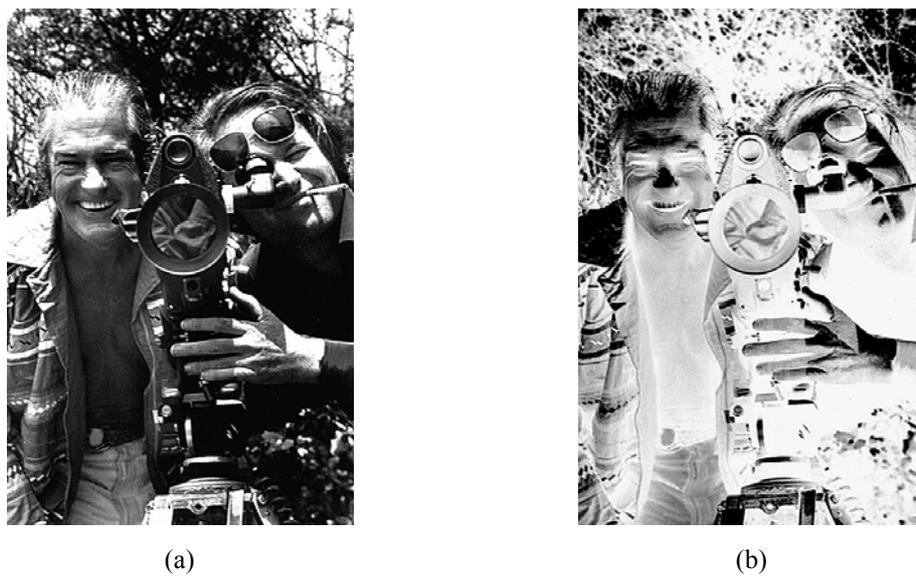
A operação de NOT cria um negativo da imagem original, invertendo cada bit dentro da cada valor de cada pixel, como mostra a Figura 2.21.

### 2.3.2. Filtros Espaciais

A filtragem espacial normalmente é utilizada para remoção de ruído ou realce de imagem. Esses operadores são chamados de filtros espaciais para distingui-los dos filtros no domínio da frequência. Os três tipos de filtros discutidos aqui incluem: 1) filtros médios, 2) filtros medianos e 3) filtros de realce. Os dois primeiros são usados tipicamente para ocultar ou remover ruído, mas eles também podem ser usados para aplicações especiais. Por exemplo, um filtro médio suaviza uma imagem, como mostra a Figura 2.22. Os filtros de realce destacam as bordas e detalhes dentro da imagem.



**Figura 2.20:** a) imagem original; b) retângulo para a máscara AND; c) imagem resultante (a) AND (b); d) retângulo para a máscara OR; e) imagem resultante (a) OR (d);



**Figura 2.21:** a) imagem original; b) operador NOT aplicado na imagem.



(a)



(b)

**Figura 2.22:** a) imagem original; b) aplicação de filtro médio 3x3. Note a aparência mais suave.

Muitos filtros espaciais são implementados com máscaras de convolução. Uma operação de máscara de convolução fornece um resultado que é uma soma de pesos dos valores de um pixel e seus vizinhos, isso é chamado de filtro linear. Um aspecto interessante das máscaras de convolução é que o efeito geral pode ser previsto com base em seu padrão geral. Por exemplo, se os coeficientes de uma máscara somam um, o brilho médio da imagem é mantido. Se os coeficientes da soma forem zero, o brilho médio é perdido e a imagem retornada é escura. Além disso, se os coeficientes alternam positivos e negativos, a máscara é um filtro que retorna somente a informação de borda; se os coeficientes são todos positivos, ele é um filtro que retorna uma imagem embaçada.

Os filtros de média operam em grupos locais de pixels chamados de vizinhos e substituem o pixel central pela média dos pixels em sua vizinhança. Essa substituição é feita com uma máscara de convolução como a máscara 3x3 a seguir:

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

Note que os coeficientes dessa máscara somam um, então o brilho da imagem será mantido, e os coeficientes são todos positivos, então ele tenderá a embaçar a imagem.

O filtro mediano é um filtro não-linear. Um filtro não-linear tem um resultado que não pode ser encontrado por uma soma de peso dos pixels vizinhos, como é feito com as máscaras de convolução. No entanto, o filtro mediano opera na vizinhança local.

Depois que o tamanho da vizinhança local é definido, o pixel central é substituído pelo valor mediano, ou centro, presente junto com os vizinhos.

Dada a seguinte vizinhança 3x3:

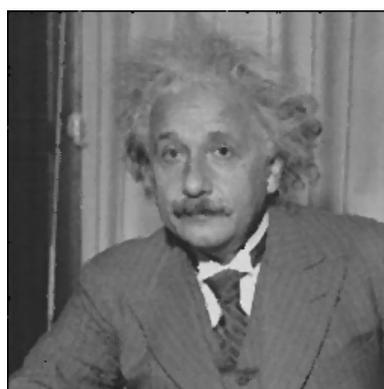
$$\begin{bmatrix} 5 & 5 & 6 \\ 3 & 4 & 5 \\ 3 & 4 & 7 \end{bmatrix}$$

Primeiro ordena-se os valores por tamanho (3,3,4,4,5,5,5,6,7); então se seleciona o valor central, neste caso o 5. Esse 5 é então colocado na posição central.

Um filtro mediano pode utilizar uma vizinhança de qualquer tamanho, mas 3x3, 5x5, 7x7 são típicos. A imagem de saída deve ser gerada em uma imagem separada (um *buffer*), de maneira que os resultados não sejam corrompidos pelo processo executado. A Figura 2.23 ilustra o uso de filtro mediano em remoção de ruído.



(a)



(b)

**Figura 2.23:** a) imagem original com a adição de ruído sal-e-pimenta; b) imagem restaurada com filtro mediano de 3x3.

Os filtros de realce, considerados aqui, incluem o tipo-laplaciano e filtros de diferenças. Esses filtros tendem a aumentar o brilho, ou realçar, os detalhes na imagem. As duas máscaras de convolução para filtros laplacianos são:

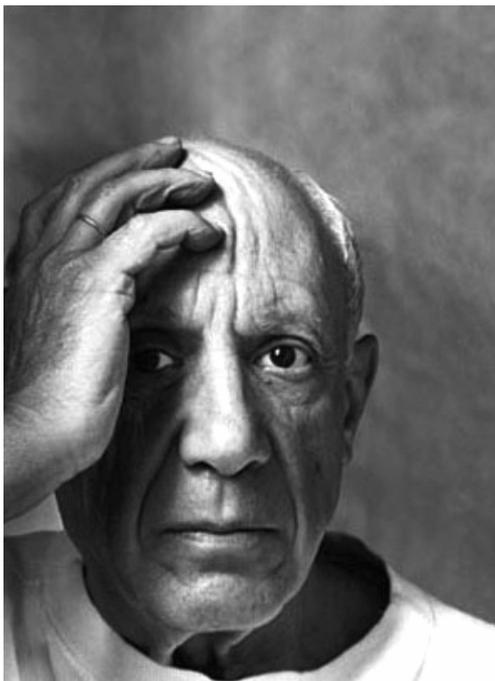
$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Os filtros laplacianos realçam os detalhes em todas as direções igualmente. A Figura 2.24 mostra o resultado da aplicação desse filtro.

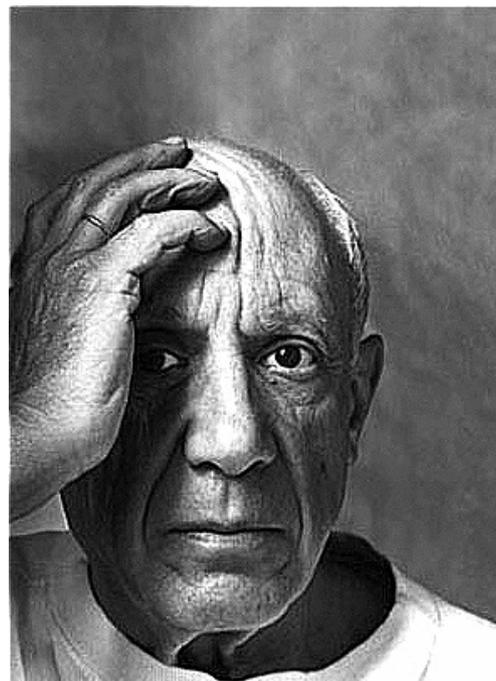
Os filtros de diferenças realçam os detalhes na direção específica da máscara selecionada.

Existem quatro máscaras de convolução de filtro de diferença, correspondendo a linhas na vertical, horizontal e nas duas direções diagonais:

Vertical	Horizontal	Diagonal 1	Diagonal 2
$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$



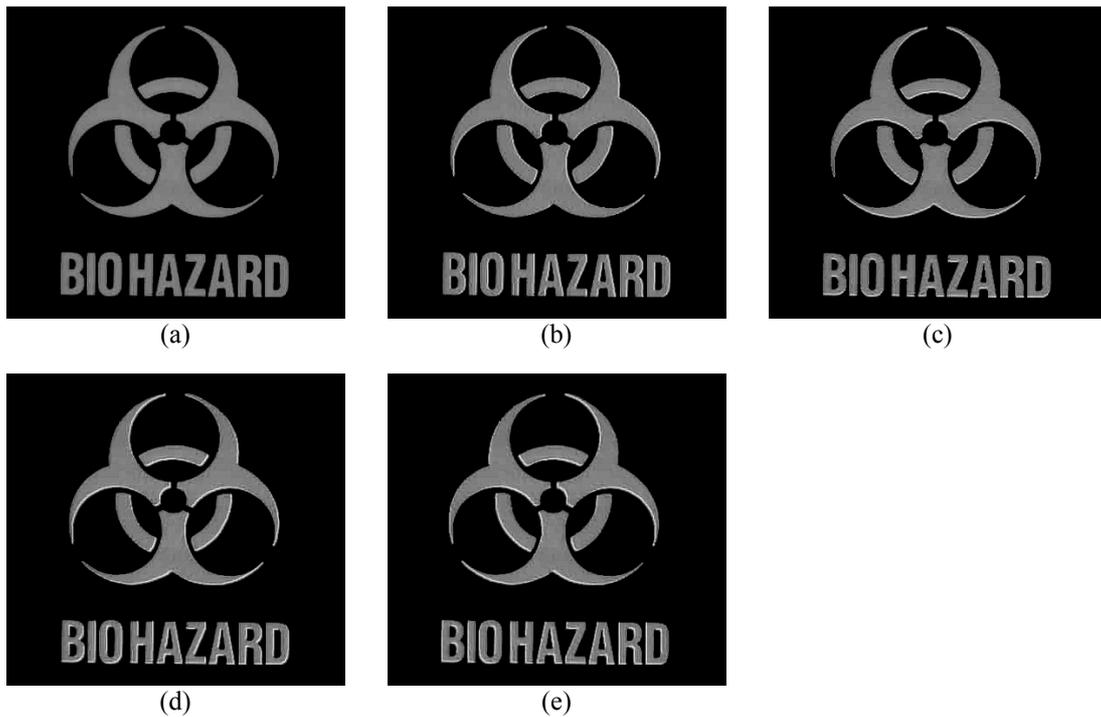
(a)



(b)

**Figura 2.24:** a) imagem original; b) imagem (a) após aplicação de filtro laplaciano.

A Figura 2.25 mostra os resultados da aplicação dessas máscaras.



**Figura 2.25:** a) imagem original; b) resultado do filtro de diferença horizontal; c) resultado do filtro de diferença vertical; d) resultado do filtro de diferença diagonal 1; e) resultado do filtro de diferença diagonal 2.

### Suavização Gaussiana

Filtros Gaussianos são de uma classe de filtros lineares de suavização com os pesos escolhidos de acordo com a forma da função Gaussiana. O filtro de suavização Gaussiana é muito eficiente para remoção de ruído feito por distribuição normal<sup>1</sup> [Jain et al. 1995]. A função Gaussiana de média zero em uma dimensão é:

$$g(x) = e^{-\frac{x^2}{2\sigma^2}}, \quad (2.20)$$

onde o parâmetro de espalhamento  $\sigma$  Gaussiano determina a largura da Gaussiana. Para processamento de imagem, a função discreta da Gaussiana de média zero,

$$g(i, j) = e^{-\frac{(i^2+j^2)}{2\sigma^2}} \quad (2.21)$$

<sup>1</sup> O fato dos pesos dos filtros serem escolhidos por uma distribuição Gaussiana e que o ruído também é distribuído dessa maneira é mera coincidência.

é usada como filtro de suavização. As funções Gaussianas têm quatro propriedades que as fazem particularmente úteis em processamento de visão. Essas propriedades indicam que os filtros de suavização Gaussianos são filtros passa-baixas efetivos para as perspectivas do domínio espacial e da frequência. Tais filtros são efetivamente implementados, e podem ser utilizados em aplicações práticas de visão. As cinco propriedades são listadas abaixo. A explicação dessas propriedades pode ser encontrada em [Jain e Schunck 1995].

1. Em duas dimensões, as funções Gaussianas possuem simetria rotacional. Isso significa que a quantidade de suavização é igual em todas as direções. Geralmente, as bordas em uma imagem não são orientadas em alguma direção particular conhecida; conseqüentemente, não há necessidade de se suavizar em mais de uma direção que em outra. A propriedade de simetria rotacional implica que o filtro de suavização Gaussiano não influenciará detecções de bordas subseqüentes em nenhuma direção particular.
2. A função da Gaussiana tem uma única ponta. Isso significa que um filtro Gaussiano suaviza, substituindo cada pixel da imagem com uma média balanceada de pixels vizinhos, de tal forma que o peso dado ao vizinho decresce monotonicamente com a distância em relação ao pixel central. Essa propriedade é importante, já que uma borda é característica local em uma imagem e uma operação de suavização que dê mais significância a pixels mais afastados distorce as características da borda.
3. A transformada de Fourier de uma Gaussiana tem um único pico no espectro da frequência. Essa propriedade é diretamente concluída com o fato que a transformada de Fourier de uma Gaussiana é por si só uma Gaussiana. Imagens são frequentemente corrompidas por sinais de alta frequência indesejáveis (ruídos e textura fina). As características desejáveis de uma imagem, como as bordas, têm componentes tanto em baixa como em alta frequência. O único pico na transformada de Fourier da Gaussiana significa que a imagem suavizada não será corrompida por contribuições de sinais de alta frequência indesejados, enquanto a maioria dos sinais desejados serão mantidos.
4. A largura, e por conseqüência o grau de suavização, de um filtro Gaussiano é parametrizado por  $\sigma$ , e o relacionamento entre  $\sigma$  e o grau de suavização é

muito simples. Um  $\sigma$  maior implica um filtro Gaussiano mais amplo e maior suavização. Pode-se ajustar o grau de suavização para alcançar um compromisso entre borramento excessivo das características da imagem desejada (muita suavização) e variações indesejáveis excessivas na imagem suavizada devido a ruído e textura fina (pouca suavização).

5. Filtros Gaussianos grandes podem ser implementados eficientemente porque as funções Gaussianas são separáveis. A convolução Gaussiana bidimensional pode ser feita convoluindo a imagem com uma Gaussiana unidimensional e então convoluindo o resultado com o mesmo filtro unidimensional orientado ortogonalmente ao Gaussiano utilizado no primeiro estágio. Assim, a quantidade de esforço computacional exigido para um filtro Gaussiano 2-D cresce linearmente na largura da máscara do filtro ao invés de crescer quadraticamente.

### Desenhando Filtros Gaussianos

Uma excelente aproximação para o filtro Gaussiano é fornecida pelos coeficientes de expansão binomiais:

$$(1+x)^n = \binom{n}{0} + \binom{n}{1}x + \binom{n}{2}x^2 + \dots + \binom{n}{n}x^n \quad (2.22)$$

Em outras palavras, utilizando a linha  $n$  do triângulo de Pascal como unidimensional, aproximação de  $n$ -pontos para um filtro Gaussiano. Por exemplo, a aproximação 5-pontos é:

1	4	6	4	1
---	---	---	---	---

correspondendo à quinta linha do triângulo de Pascal. Essa máscara é utilizada para suavizar a imagem na direção horizontal. Um filtro Gaussiano bidimensional pode ser implementado como convoluções sucessivas de duas Gaussianas unidimensionais, uma na direção horizontal e a outra na direção vertical. Isso também pode ser implementado usando somente uma única máscara unidimensional transpondo a imagem entre as convoluções e depois da convolução final.

Essa técnica funciona bem para filtros de até aproximadamente  $n=10$ . Para filtros maiores, os coeficientes em expansão binomial são grandes demais para a maioria dos computadores; no entanto, filtros arbitrariamente grandes podem ser implementados pela aplicação repetitiva de um filtro Gaussiano menor. O  $\sigma$  da aproximação binomial para um filtro Gaussiano pode ser computado utilizando mínimo-quadrático para adequar a uma função Gaussiana aos coeficientes binomiais.

Uma outra abordagem em projeto de filtro Gaussiano é calcular a máscara de pesos diretamente pela distribuição Gaussiana discreta [Kohl 1992]:

$$g[i, j] = ce^{-\frac{(i^2+j^2)}{2\sigma^2}} \quad (2.23)$$

Onde  $c$  é uma constante de normalização. Reescrevendo isso como:

$$\frac{g[i, j]}{c} = e^{-\frac{(i^2+j^2)}{2\sigma^2}} \quad (2.24)$$

e escolhendo um valor para  $\sigma^2$ , pode-se avaliar uma janela  $n \times n$  para obter um *kernel*, ou máscara, para que o valor [0,0] seja igual a 1. Por exemplo, escolhendo  $\sigma^2 = 2$  e  $n=7$ , a equação (2.24) leva a essa matriz:

[i,j]	-3	-2	-1	0	-1	-2	-3
-3	.011	.039	.082	.105	.082	.039	.011
-2	.039	.135	.287	.368	.287	.135	.039
-1	.082	.287	.606	.779	.606	.287	.082
0	.105	.368	.779	1.000	.779	.368	.105
1	.082	.287	.606	.779	.606	.287	.082
2	.039	.135	.287	.368	.287	.135	.039
3	.011	.039	.082	.105	.082	.039	.011

no entanto, deseja-se que o filtro de pesos tenha valores inteiros para cálculos simplificados. Então, toma-se o valor de um dos cantos da matriz, e escolhe-se  $k$  de maneira que esse valor se torne 1. Para esse exemplo, tem-se:

$$\frac{g[3,3]}{k} = e^{-\frac{(3^2+3^2)}{2(2)^2}} = 0.011 \Rightarrow k = \frac{g[3,3]}{0.011} = \frac{1.0}{0.011} = 91$$

multiplicando o resto dos pesos por  $k$ , obtém-se:

[i,j]	-3	-2	-1	0	-1	-2	-3
-3	1	4	7	10	7	4	1
-2	4	12	26	33	26	12	4
-1	7	26	55	71	55	26	7
0	10	33	71	91	71	33	10
1	7	26	55	71	55	26	7
2	4	12	26	33	26	12	4
3	1	4	7	10	7	4	1

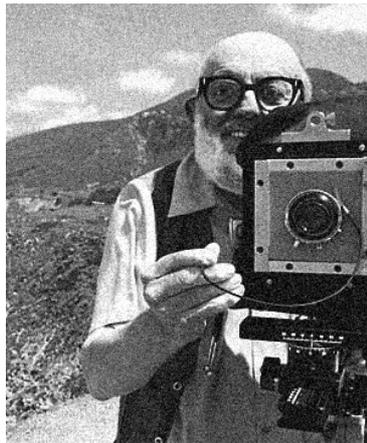
Essa é a máscara de convolução resultante para o filtro Gaussiano. No entanto, a soma dos pesos da máscara não soma 1. Assim, quando a convolução for feita, os valores dos pixels de saída devem ser normalizados pela soma dos pesos da máscara para assegurar que as regiões de intensidade uniforme não sejam afetadas. Então:

$$\sum_{i=-3}^3 \sum_{j=-3}^3 g[i, j] = 1115$$

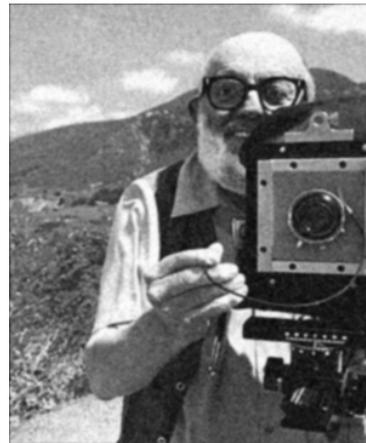
Assim,

$$h[i, j] = \frac{1}{1115} (f[i, j] * g[i, j])$$

onde os pesos de  $g[i, j]$  são todos valores inteiros. Os resultados da suavização Gaussiana utilizando a máscara acima são apresentados na Figura 2.26.



(a)



(b)

**Figura 2.26:** Resultados de suavização usando filtro Gaussiano 3x3. (a) Imagem original corrompida por ruído Gaussiano. (b) Imagem suavizada.

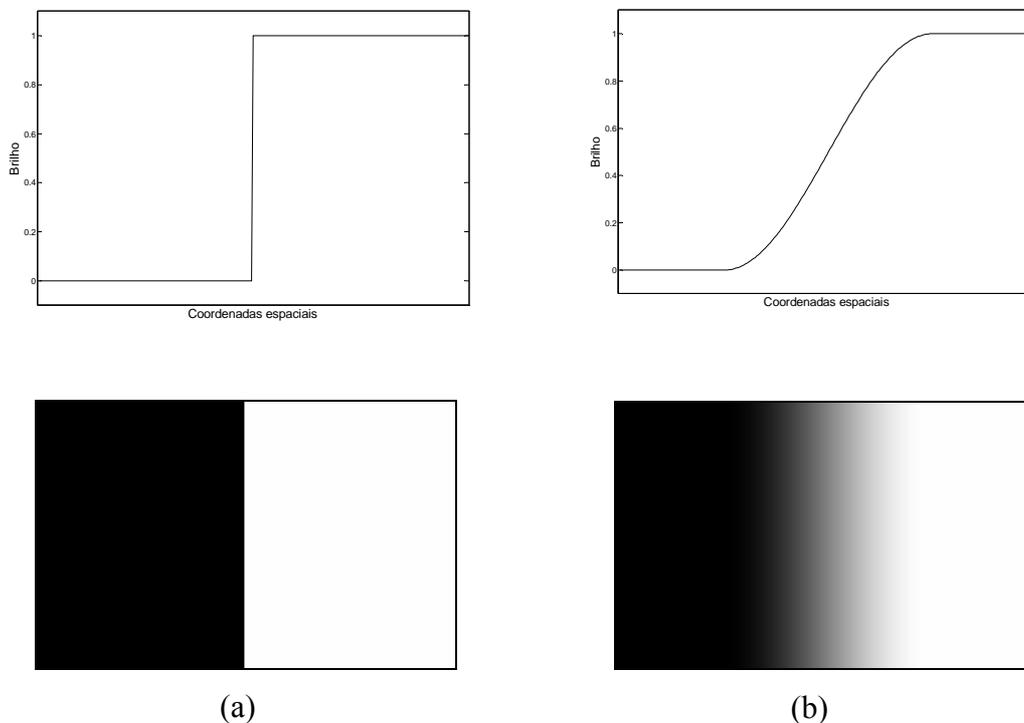
### 2.3.3. Detecção de Bordas

Os operadores de detecção de borda apresentados aqui representam os vários tipos de operadores em uso hoje. Muitos são implementados usando máscara de convolução e a maior parte é baseada em aproximação discreta de operadores diferenciais. Os operadores diferenciais medem a taxa de mudança em uma função, nesse caso, a função de brilho da imagem. Uma grande mudança no brilho da imagem em uma curta distância espacial indica a presença de uma borda. Alguns operadores de detecção de borda retornam informação de orientação (informação sobre a direção da borda), enquanto outras só retornam informação da existência de uma borda em cada ponto.

Em muitos desses operadores, o ruído na imagem pode causar problemas. Por isso é melhor fazer um pré-processamento da imagem para eliminar, ou pelo menos minimizar, os efeitos do ruído. Para lidar com os efeitos do ruído, é preciso analisar as vantagens e desvantagens entre sensibilidade e precisão de um detector de bordas. Por exemplo, se os parâmetros são configurados de maneira que o detector de bordas seja bastante sensível, ele tenderá a encontrar muitos pontos de borda que são atribuídos ao ruído. Se ele for configurado com menos sensibilidade, pontos válidos podem ser perdidos. Os parâmetros que podem ser configurados incluem o tamanho da máscara de detecção de borda e o valor do limiar de tom-de-cinza. Uma máscara grande é menos sensível a ruído; um baixo limiar de tom-de-cinza tende a reduzir os efeitos de ruído.

Os operadores de detecção de borda são baseados na idéia que a informação de borda em uma imagem é encontrada observando o relacionamento entre o pixel e seus vizinhos. Se o valor de tom-de-cinza de um pixel é similar ao de seus vizinhos, provavelmente não há borda nesse ponto. Em outras palavras, uma borda é definida como uma descontinuidade em valores em tom-de-cinza. Idealmente, uma borda separa dois objetos distintos. Na prática, bordas aparentes são causadas por mudanças na cor ou textura, ou condições especiais de iluminação presentes durante o processo de aquisição da imagem.

A Figura 2.27 ilustra as diferenças entre uma borda ideal e uma borda real. O eixo vertical representa brilho, e o eixo horizontal representa as coordenadas espaciais.



**Figura 2.27:** Borda ideal vs. real. (a) Borda ideal e (b) Borda real.

A mudança abrupta no brilho caracteriza uma borda ideal. Na correspondente imagem, a borda aparece muito distinta. A Figura 2.27 (b) mostra a representação de uma borda real, que causa mudanças graduais. Essa mudança gradual, é uma forma reduzida de borramento, causada pelo dispositivo de imageamento, as lentes, ou a iluminação, e isso é típico das imagens do mundo real (em oposição às geradas por computador). Na imagem da Figura 2.27 (b), onde a borda foi exagerada para propósitos de ilustração, nota-se que, por uma perspectiva visual, essa imagem

contém a mesma informação que a imagem ideal: preto de um lado e branco do outro, com uma linha no centro.

## Gradiente

Detecção de bordas é essencialmente a operação de detecção de mudanças locais significantes na imagem. Em uma dimensão, uma borda de degrau é associada a um pico local na primeira derivativa. O gradiente é uma medida de mudança em uma função e uma imagem pode ser considerada um array de amostras de alguma função contínua de intensidade de imagem. Por analogia, mudanças significantes nos valores de cinza de uma imagem podem ser detectadas usando aproximação discreta para o gradiente. O gradiente é equivalente bidimensional da primeira derivativa e é definido como o vetor [Gonzalez 1993]

$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}. \quad (2.25)$$

Existem duas propriedades importantes associadas ao gradiente: (1) o vetor  $G[f(x, y)]$  aponta na direção da taxa máxima de aumento da função  $f(x, y)$ , e (2) a magnitude do gradiente, dada por:

$$G[f(x, y)] = \sqrt{G_x^2 + G_y^2}, \quad (2.26)$$

igual a taxa máxima de aumento de  $f(x, y)$  por unidade de distância na direção de  $G$ . No entanto, é prática comum aproximar a magnitude do gradiente por valores absolutos:

$$G[f(x, y)] \approx |G_x| + |G_y|, \quad (2.27)$$

ou

$$G[f(x, y)] \approx \max(|G_x|, |G_y|). \quad (2.28)$$

Por análise vetorial, a direção do gradiente é definida como:

$$\alpha(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \quad (2.29)$$

onde o ângulo  $\alpha$  é medido em relação ao eixo  $x$ .

Note-se que a magnitude do gradiente é realmente independente da direção da borda. Tais operadores são chamados de isotrópicos.

### Aproximação Numérica

Para imagens digitais, as derivativas na equação (2.24) são aproximadas pelas diferenças. A aproximação do gradiente mais simples é:

$$G_x \cong f[i, j+1] - f[i, j] \quad (2.30)$$

$$G_y \cong f[i, j] - f[i+1, j]. \quad (2.31)$$

Lembrando que  $j$  corresponde à direção  $x$  e  $i$  à direção negativa de  $y$ . Isso pode ser implementado com uma simples máscara de convolução como na equação (2.32):

$$G_x = \begin{bmatrix} -1 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad (2.32)$$

Calculando-se uma aproximação para o gradiente, é crítico que as parciais derivativas  $x$  e  $y$  sejam calculadas exatamente na mesma posição no espaço. No entanto, utilizando-se a aproximação acima,  $G_x$  é realmente a aproximação para o gradiente no ponto interpolado  $[i, j + \frac{1}{2}]$  e  $G_y$  em  $[i + \frac{1}{2}, j]$ . Por essa razão, as primeiras diferenças  $2 \times 2$ , ao invés de máscaras  $2 \times 1$  e  $1 \times 2$ , são frequentemente usadas para as derivadas parciais  $x$  e  $y$ :

$$G_x = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \quad (2.33)$$

Assim, as posições em que os gradientes nas direções  $x$  e  $y$  são calculados, são as mesmas. Esse ponto fica entre todos os quatro pixels na vizinhança  $2 \times 2$  no ponto interpolado  $[i + \frac{1}{2}, j + \frac{1}{2}]$ . Esse fato pode causar alguma confusão. Então, uma abordagem alternativa é usar uma vizinhança  $3 \times 3$  e calcular o gradiente sobre o pixel central.

### Operador de Roberts

O operador de Roberts marca somente pontos de borda; ele não retorna nenhuma informação sobre orientação de borda. Ele é o mais simples dos operadores de detecção de bordas e funciona melhor com imagens binárias (imagens em tons de cinza podem ser transformadas em binárias pela operação de limiarização). Existem duas formas do operador de Roberts. A primeira consiste na raiz quadrada da soma das diferenças dos vizinhos diagonais quadrados, como segue:

$$\sqrt{[I(r, c) - I(r - 1, c - 1)]^2 + [I(r, c - 1) - I(r - 1, c)]^2} \quad (2.34)$$

A segunda forma do operador de Roberts é a soma da magnitude das diferenças dos vizinhos diagonais, como segue:

$$|[I(r, c) - I(r - 1, c - 1)]| + |[I(r, c - 1) - I(r - 1, c)]| \quad (2.35)$$

A segunda forma da equação é frequentemente usada na prática devido à sua eficiência computacional. Normalmente é mais rápido para um computador encontrar valor absoluto do que encontrar a raiz quadrada.

## Operador de Sobel

As máscaras de detecção de bordas de Sobel procuram por bordas na direção horizontal e vertical e combinam essa informação em uma simples métrica. Essas máscaras são:

Máscara de Linha	Máscara de Coluna
$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

Cada uma dessas máscaras é convoluída na imagem. A cada posição de pixel tem-se dois valores:  $s_1$ , correspondendo ao resultado da máscara de linha e  $s_2$ , da máscara de coluna. Esses valores são utilizados para calcular duas métricas: a magnitude da borda e a direção da borda, definidas como:

Magnitude de Borda

$$\sqrt{s_1^2 + s_2^2} \quad (2.36)$$

Direção de Borda

$$\tan^{-1} \left[ \frac{s_1}{s_2} \right] \quad (2.37)$$

A direção de borda é perpendicular à borda porque a direção especificada é a direção do gradiente, junto com a mudança do nível de cinza.

## Operador de Prewitt

O operador de Prewitt é similar ao operador de Sobel, mas com coeficientes diferentes nas máscaras, definidos como:

Máscara de Linha	Máscara de Coluna
$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$

Cada uma dessas máscaras é convoluída na imagem. A cada posição de pixel, dois valores são encontrados:  $p_1$ , correspondendo à máscara de linha e  $p_2$  à máscara de coluna. Esses resultados são utilizados para determinar duas métricas: a magnitude de borda e a direção de borda, que são definidas como:

Magnitude de Borda

$$\sqrt{p_1^2 + p_2^2} \quad (2.38)$$

Direção de Borda

$$\tan^{-1} \left[ \frac{p_1}{p_2} \right] \quad (2.39)$$

Como no detector de borda de Sobel, a direção está 90° em relação à direção aparente da borda.

## Máscaras de Bússola de Kirsh

As máscaras de detecção de bordas de Kirsch são chamadas de máscaras de bússola porque são definidas por utilizar uma única máscara e a girar nas oito

principais orientações de uma bússola: Norte, Noroeste, Oeste, Sudoeste, Sul, Sudeste e Nordeste. As máscaras são definidas como:

$$\begin{array}{cccc}
 k_0 & k_1 & k_2 & k_3 \\
 \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} & \begin{bmatrix} -3 & -5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} & \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} & \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \\
 \\
 k_4 & K_5 & k_6 & k_7 \\
 \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} & \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} & \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} & \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}
 \end{array}$$

A magnitude de borda é definida como o valor máximo encontrado pela convolução de cada máscara com a imagem. A direção de borda é definida pela máscara que produz a maior magnitude; por exemplo,  $k_0$  corresponde a uma borda vertical, enquanto que  $k_5$  corresponde a uma borda diagonal na direção Noroeste/Sudeste. Note-se que as máscaras  $k_4$ ,  $k_5$ ,  $k_6$  e  $k_7$  são realmente, o espelhamento das máscaras  $k_0$ ,  $k_1$ ,  $k_2$  e  $k_3$  sobre o eixo central.

### Máscaras de Bússola de Robinson

As máscaras de bússola de Robinson são usadas de maneira similar às máscaras de Kirsch, mas implementadas de forma simplificada porque elas contam somente com os coeficientes 0, 1 e 2, e são simétricas sobre seus eixos de direção, o eixo com zeros. É necessário calcular somente os resultados de quatro máscaras; o resultado das outras quatro pode ser obtido negando o resultado das quatro primeiras. As máscaras são:

$$\begin{array}{cccc}
 r_0 & r_1 & r_2 & r_3 \\
 \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} & \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix} \\
 \\
 r_4 & r_5 & r_6 & r_7 \\
 \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} & \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} & \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} & \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}
 \end{array}$$

A magnitude de borda é definida como o valor máximo encontrado pela convolução de cada máscara com a imagem. A borda de direção é definida pela máscara que produz a magnitude máxima. Note-se que as máscaras  $r_0$  e  $r_6$  são as mesmas que as máscaras de Sobel e que qualquer máscara de detecção de borda pode ser estendida, girando-as de forma que essas máscaras de bússola permitam extrair informação explícita de suas bordas, em qualquer direção.

## Operadores Laplacianos

Os operadores Laplacianos são similares aos utilizados em pré-processamento na seção (2.3.2). As três máscaras laplacianas a seguir, representam diferentes aproximações dos operadores laplacianos. Diferentemente das máscaras de bússola, as máscaras laplacianas têm simetria de rotação, implicando que as bordas em todas as direções contribuam com o resultado. Elas são aplicadas selecionando-se uma das máscaras e convoluindo-a com a imagem. O sinal do resultado (positivo ou negativo) de dois pixels adjacentes fornece informação direcional, e informa qual lado da borda tem maior brilho.

### Máscaras Laplacianas

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Essas máscaras diferem do tipo laplaciano descrito anteriormente na seção (2.3.2) pelo fato de os coeficientes centrais serem decrescidos de 1. Com essas máscaras, procura-se encontrar as bordas. O interesse não é a imagem propriamente dita. Uma maneira fácil de se visualizar a diferença é considerar o efeito que cada máscara teria quando aplicada a uma área de valor constante. As máscaras de convolução anteriores retornariam o valor zero. Se os coeficientes internos fossem incrementados de 1, cada um teria o valor original em tom de cinza. Dessa forma se o interesse é a detecção de bordas, a soma dos coeficientes deve ser zero. Se o interesse for manter a maior parte a informação original na imagem, os coeficientes devem somar um valor maior que zero. Quanto maior a soma, menor a diferença entre a imagem original e processada.

## Máscaras de Frei-Chen

As máscaras de Frei-Chen são únicas pelo fato de formarem um completo conjunto de vetores de base. Isso significa que é possível representar qualquer sub-imagem como uma soma de pesos das nove máscaras de Frei-Chen Figura 2.28. Esses pesos são encontrados projetando-se uma sub-imagem sobre essas máscaras. Esse processo de projeção é similar ao processo de convolução porque ambos sobrepõem a máscara na imagem, multiplicam os termos coincidentes e somam os resultados (um produto interno de vetor) [Umbaugh 1998]. Isso é melhor ilustrado no seguinte exemplo:

Suponda-se a seguinte sub-imagem:

$$I_s = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

Para projetar essa sub-imagem nas máscaras de Frei-Chen, inicia-se encontrando a projeção em  $f_l$ . Sobrepondo a sub-imagem na máscara e considerando a primeira linha. O 1 no canto superior-esquerdo da sub-imagem coincide com o 1 do canto superior-esquerdo da máscara  $f_l$ , o 0 está sobre o  $\sqrt{2}$ , e o 1 no canto superior-direito da sub-imagem coincide com o 1 da máscara. Todos devem ser somados e então multiplicados por  $1/2\sqrt{2}$  fator de normalização da máscara. A projeção  $I_s$  em  $f_l$  é:

$$\frac{1}{2\sqrt{2}} [1(1) + 0(\sqrt{2}) + 1(1) + 1(0) + 0(0) + 1(-1) + 0(-\sqrt{2}) + 1(-1)] = 0$$

Se esse processo for seguido, para sub-imagem  $I_s$  em cada máscara de Frei-Chen, tem-se:

$$f_1 \rightarrow 0, f_2 \rightarrow 0, f_3 \rightarrow 0, f_4 \rightarrow 0, f_5 \rightarrow -1, f_6 \rightarrow 0, f_7 \rightarrow 0, f_8 \rightarrow 0 - 1, f_9 \rightarrow 2$$

Agora pode ser visto que, para um conjunto completo de vetores de base, é possível representar uma sub-imagem por uma soma de pesos. Os vetores de base nesse caso são as máscaras de Frei-Chen, e os pesos são os valores de projeção. Utilizando os

valores e os multiplicando por cada máscara; então somando os valores correspondentes. Para esse exemplo, os únicos termos diferentes de zero correspondem aos das máscaras  $f_5, f_8$  e  $f_9$ , então:

$$(-1)\left(\frac{1}{2}\right)\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} + (-1)\left(\frac{1}{6}\right)\begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} + (2)\left(\frac{1}{3}\right)\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} = I_s$$

$$\begin{array}{ccc} \frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} & \frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} & \frac{1}{2\sqrt{2}}\begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix} \\ f_1 & f_2 & f_3 \\ \\ \frac{1}{2\sqrt{2}}\begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix} & \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} & \frac{1}{2}\begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix} \\ f_4 & f_5 & f_6 \\ \\ \frac{1}{6}\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} & \frac{1}{6}\begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} & \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \\ f_7 & f_8 & f_9 \end{array}$$

**Figura 2.28:**  $f_1, f_2, f_3$  e  $f_4$  compreendem o sub-espço de bordas;  $f_5, f_6, f_7$  e  $f_8$  compreendem o sub-espço de linhas e  $f_9$  compreende o sub-espço de média.

Até o momento foi apresentada a maneira como as máscaras de Frei-Chem podem ser usadas para representar uma sub-imagem como uma soma de pesos, mas como elas podem ser utilizadas para detecção de bordas? As máscaras de Frei-Chem podem ser agrupadas em um conjunto de quatro máscaras para um sub-espço de borda, quatro máscaras para o sub-espço de linhas e uma máscara para um sub-espço de média. Para utilizá-las para detecção de bordas, seleciona-se um sub-espço particular de

interesse e encontra-se a projeção relativa da imagem no sub-espço particular. Isso é dado pelas equações (2.4), (2.41) e (2.42):

$$\cos \theta = \sqrt{\frac{M}{S}} \quad (2.40)$$

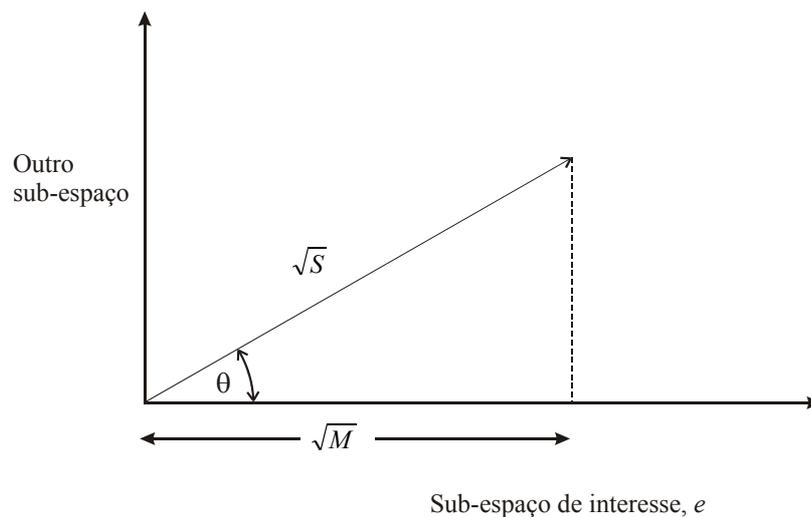
onde

$$M = \sum_{k \in \{e\}} (I_s, f_k)^2 \quad (2.41)$$

e

$$S = \sum_{k=1}^9 (I_s, f_k)^2 \quad (2.42)$$

o conjunto  $\{e\}$  consiste das máscaras de interesse. A notação  $(I_s, f_s)$  refere-se ao processo de sobreposição de máscaras na sub-imagem, multiplicando os termos coincidentes e somando os resultados (um produto interno de vetor). Uma ilustração disso é apresentada na Figura 2.29. A vantagem desse método é poder selecionar máscaras particulares de interesse para borda ou linha e considerar somente a projeção dessas máscaras.



**Figura 2.29:** Uma representação bidimensional do conceito de projeção de Frei-Chen. O espaço real de Frei-Chen é nove-dimensional, onde cada dimensão é dada por uma das máscaras  $f_k$ .

## Detector de Borda Canny

O detector de borda Canny é o primeiro derivativo da Gaussiana e é o operador que mais se aproxima do produto da taxa de sinal-para-ruído e localização. O algoritmo de detecção de borda de Canny é definido pela seguinte notação. Seja  $I[i, j]$  a denotação da imagem, o resultado da convolução da imagem com um filtro de suavização Gaussiano é uma matriz de dados suavizados,

$$S[i, j] = G[i, j; \sigma] * I[i, j] \quad (2.43)$$

onde  $\sigma$  é o espalhamento da Gaussiana e controla o grau de suavização [Jain et al. 1997].

O gradiente da matriz suavizada  $S[i, j]$  pode ser calculado utilizando aproximações de primeira diferença 2x2 para produzir duas matrizes  $P[i, j]$  e  $Q[i, j]$  para as derivadas parciais de  $x$  e  $y$ :

$$P[i, j] \approx (S[i, j+1] - S[i, j] + S[i+1, j+1] - S[i+1, j]) / 2 \quad (2.44)$$

$$Q[i, j] \approx (S[i, j] - S[i+1, j] + S[i, j+1] - S[i+1, j+1]) / 2. \quad (2.45)$$

A média das diferenças finitas é feita com um quadrado 2x2 de maneira que as derivadas parciais de  $x$  e  $y$  são calculadas no mesmo ponto da imagem. A magnitude e orientação do gradiente podem ser calculadas por fórmulas de conversão padrão de retangular para polar:

$$M[i, j] = \sqrt{P[i, j]^2 + Q[i, j]^2} \quad (2.46)$$

$$\theta[i, j] = \arctan\left(\frac{Q[i, j]}{P[i, j]}\right), \quad (2.47)$$

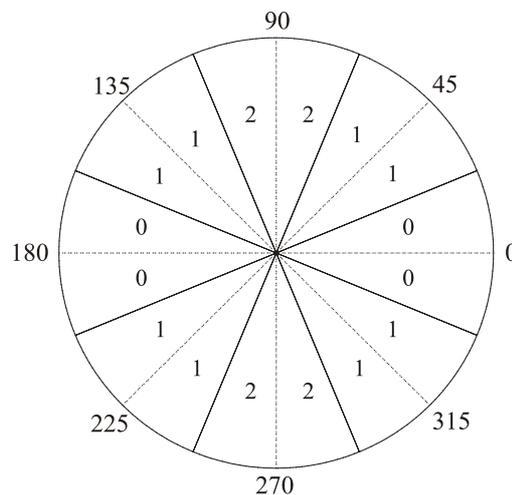
onde a função de arco-tangente toma dois argumentos e gera uma borda sobre o círculo inteiro de possíveis direções. Essas funções devem ser calculadas eficientemente, preferencialmente sem a utilização de aritmética de ponto flutuante. É possível calcular a magnitude e orientação do gradiente das derivadas parciais por uma tabela lookup. O arco-tangente pode ser calculado, na sua maior parte, usando

aritmética de ponto fixo com poucos cálculos essenciais de ponto flutuante feitos em software inteiro e aritmética de ponto fixo [Cody e Waite 1980]. Sedgewick [Sedgewick 1988] fornece um algoritmo para uma aproximação de inteiro para o ângulo do gradiente que pode ser suficientemente eficiente para muitas aplicações.

### Supressão não-máxima

A imagem de magnitude  $M[i, j]$ , tem valores grandes onde o gradiente da imagem é grande, mas isso não é suficiente para identificar as bordas, já que o problema de encontrar localizações na imagem quando há rápidas mudanças foi meramente transformado em um problema de encontrar as localizações na magnitude  $M[i, j]$  que são máximas locais. Para identificar bordas, as grandes elevações na magnitude devem ser estreitadas de maneira que somente as magnitudes em pontos de maior mudança local permaneçam. Esse processo é chamado de supressão não-máxima que nesse caso resulta em bordas mais estreitas.

A supressão não-máxima estreita elevações na magnitude do gradiente em  $M[i, j]$ , suprimindo todos os valores ao longo da linha do gradiente. O algoritmo inicia-se reduzindo o ângulo do gradiente  $\theta[i, j]$  em um dos quatro setores mostrados na Figura 2.30.



**Figura 2.30:** Partição das possíveis orientações do gradiente em setores para a supressão não-máxima. Existem quatro setores numerados de 0 a 3, correspondendo às quatro possíveis combinações de elementos em uma vizinhança de 3x3 que uma linha passa através à medida que ela passa pelo centro da vizinhança. As divisões do círculo de possíveis orientações de linhas estão em graus.

$$\zeta[i, j] = \text{Sector}(\theta[i, j]). \quad (2.48)$$

O algoritmo verifica a vizinhança 3x3 da magnitude  $M[i, j]$ . Nesse momento, o elemento central  $M[i, j]$  da vizinhança é comparado com seus dois vizinhos ao longo da linha do gradiente dado pelo valor do setor  $\zeta[i, j]$  no centro da vizinhança. Se o valor da magnitude  $M[i, j]$  central não for maior que os dois vizinhos de magnitude pela linha do gradiente, então  $M[i, j]$  será zero. Esse processo estreita as grandes elevações da magnitude do gradiente  $M[i, j]$  em elevações de apenas um pixel de espessura. Os valores para a altura da elevação são retidos na magnitude de supressão não-máxima.

Seja

$$N[i, j] = nms(M[i, j], \zeta[i, j]). \quad (2.49)$$

a denotação do processo de supressão não-máxima. Os valores diferentes de zero em  $N[i, j]$  correspondem à quantidade de contraste em uma mudança de degrau em uma imagem de intensidade. Quanto à suavização feita na primeira etapa na detecção de borda, a imagem de magnitude de supressão não-máxima  $N[i, j]$  conterá muitos fragmentos de bordas falsas causados por ruídos e texturas finas. O contraste das bordas falsas é reduzido.

### Limiarização

O procedimento típico utilizado para reduzir o número de falsos fragmentos de borda no gradiente de magnitude de supressão não-máxima é aplicar uma limiarização em  $N[i, j]$ . Todos os valores abaixo do limiar são alterados para zero. O resultado da aplicação de uma limiarização na magnitude de supressão não-máxima é uma matriz de bordas detectadas na imagem  $I[i, j]$ . Ainda haverá falsos contornos se o limiar  $\tau$  for muito baixo (falso positivo), e as porções do contorno podem se perdidas (falso negativo) devido à suavização do contraste de borda por sombras ou porque o limiar  $\tau$  está alto. Selecionar o limiar apropriado não é fácil e envolve um pouco de tentativa e erro. Uma limiarização mais efetiva envolve o uso de dois limiares.

O algoritmo de limiarização dupla utiliza a imagem de supressão não-máxima,  $N[i, j]$ , e aplica dois limiares  $\tau_1$  e  $\tau_2$ , com  $\tau_2 \approx 2\tau_1$ , para produzir duas imagens de borda  $T_1[i, j]$  e  $T_2[i, j]$ . A imagem  $T_2$  é formada por um limiar mais alto, e contém menos bordas falsas; mas  $T_2$  pode ter lacunas nos contornos (muitos falsos negativos). O algoritmo de limiarização dupla conecta as bordas de  $T_2$  em contornos. Quando alcança o final de um contorno, o algoritmo procura em  $T_1$  pela localização dos 8-vizinhos para bordas que podem ser conectadas ao contorno. O algoritmo continua juntando bordas de  $T_1$  até que a lacuna tenha sido atravessada por uma borda em  $T_2$ . O algoritmo executa a conexão de bordas como uma limiarização por produto e resolve alguns dos problemas de escolha de limiar. O algoritmo de detecção de borda de Canny é mostrado em, Algoritmo 1.

---

**Algoritmo 1:** Detecção de Borda Canny.

---

- 1 Suaviza a imagem com um filtro Gaussiano.
  - 2 Calcula a magnitude do gradiente e a orientação utilizando aproximações para a diferença finita para as derivações parciais.
  - 3 Aplica a supressão não-máxima na magnitude do gradiente.
  - 4 Utiliza o algoritmo de limiarização dupla para detectar e conectar bordas.
- 

## 2.4. Aplicações

Seguem abaixo exemplos de aplicações que podem se beneficiar de um sistema de aquisição 3D, algumas já existentes e outras que ainda podem ser implementadas.

- **Animação 3D:** o processo de modelagem é muito demorado dependendo da complexidade dos objetos. Com a aquisição 3D esse tempo pode ser drasticamente reduzido com um nível de detalhe superior.
- **Cinema:** aquisição 3D de atores e utilização de sistema de captura de movimentos para inserir os personagens em ambientes hostis, sem colocar a vida humana em risco.
- **Jogos:** o usuário participa de jogos com a sua representação digital gerada por um sistema de aquisição 3D.

- **Simulação:** uma possível aplicação é a simulação da postura e mobilidade animais extintos, uma vez que seus ossos tenham sido digitalizados em 3D.
- **Mapeamento topológico em lugares remotos:** Miller [Miller et al. 1999] descreve um sistema de aquisição 3D utilizado para fazer o levantamento de um denso mapa topológico da cratera Haughton nas Ilhas Devon no ártico canadense, para o estudo de ambientes análogos a Marte, para um grupo de cientistas da NASA. Esse sistema utiliza um sensor de distância laser baseado no método de tempo-de-vôo e um helicóptero autônomo que é utilizado como sistema de varredura.
- **Mapeamento de Cavernas:** atualmente o trabalho de mapeamento de cavernas é extremamente trabalhoso e demorado. Uma vez que a caverna é mapeada em 3D, esse mapa pode ser utilizado como treinamento prévio de instrutores e para se analisar as alterações sofridas ao longo do tempo.
- **Inspeção:** falhas de estrutura em represas, pontes, torres e prédios [EDG 2004)] podem ser detectadas através de aquisição 3D.
- **Criminalística:** uma possível aplicação para um sistema de aquisição 3D, é a digitalização de uma cena de crime para estudo detalhado sem depender de grande período de isolamento do local.
- **Aplicações Espaciais:** o grupo Neptec [Neptec 2004] desenvolveu um sistema de aquisição 3D baseado em triangulação laser para fazer o imageamento e rastreamento de elementos da Estação Espacial Internacional [Sansón 2002].
- **Perícias em acidentes aéreos:** digitalizando os destroços de uma aeronave, pode-se remontá-la digitalmente e fazer análises e simulações para determinação das causas do acidente.

- **Prototipagem rápida e reengenharia:** quando se deseja alterar uma peça que não está funcionando adequadamente ela é digitalizada, alterada digitalmente e remontada utilizando uma máquina de prototipagem rápida.
- **Construção de Museus virtuais:** algumas peças de museu não podem ser deslocadas devido à seu peso ou fragilidade (esqueletos de animais pré-históricos, esculturas, etc). Essas peças uma vez digitalizadas poderiam ser exibidas na internet e exploradas pelo usuário em um computador de mesa, em cavernas virtuais de maneira imersiva ou reconstruídas fisicamente em máquinas específicas e exibidas naturalmente em museus reais. No projeto *“The Digital Michelangelo Project: 3D Scanning of Large Statues”* [Levoy 2000], várias estátuas de Michelangelo foram digitalizadas durante um ano com um nível de detalhamento grande o suficiente para descobrir como Michelangelo criava as estátuas, observando as marcas deixadas pelas ferramentas.
- **Determinação do percentual de gordura:** o trabalho de França [França et al. 2004] descreve um método para se calcular o percentual de gordura corpórea, baseado em um sistema de aquisição 3D, utilizando triangulação laser.

## 2.5. Sistemas de Aquisição 3D

Nessa seção serão apresentados trabalhos relacionados usando variações das tecnologias de luz estruturada, estereoscopia e tempo de voo, discutidas no capítulo 2.

### 2.5.1. Luz Estruturada

O sistema baseado em um sensor de luz estruturada apresentado por Rusinkiewicz [Rusinkiewicz et al. 2002] projeta diferentes padrões de luz a 60Hz e faz a aquisição de modelos 3D em tempo-real.

Para obter uma imagem completa de profundidade é necessário varrer a superfície com um feixe de luz em forma de linha (como é feito em muitos scanners laser comerciais, com uma única linha) ou projetar múltiplas linhas. Dessa maneira, a aquisição de dados é mais rápida, porém, esse sistema deve ter algum método para

distinguir as linhas. As três maneiras mais conhecidas de se fazer isso são: 1- assumindo a continuidade da superfície de maneira que as linhas projetadas sejam adjacentes nas imagens da câmera [Proesmans et al. 1998], 2- diferenciando as linhas pelo uso de cores [Boyer e Kak 1987] e 3- codificando as linhas, variando sua iluminação com o passar do tempo. A primeira abordagem (assumindo continuidade) permite que a profundidade seja determinada com um quadro (*frame*), mas não funciona se a superfície tiver descontinuidades. Na abordagem, utilizando cor, é possível fazer a aquisição de superfícies mais complexas, mas haverá falha se a superfície for texturizada. A codificação temporal de linha é robusta para superfícies com textura moderada, mas necessita de muitos quadros para se calcular a profundidade e, dependendo da implementação do sistema, pode falhar, se o objeto se mover.

Em um modelo simples de sistema de coerência temporal, todo o processo de aquisição demanda um determinado número de quadros  $n$  e, em cada quadro, cada linha é ligada ou desligada. O padrão de liga-desliga de cada linha é escolhido para identificar, unicamente, aquela linha. Por exemplo, suponha-se que se deseje utilizar  $2^n$  linhas diferentes. Associa-se a cada linha um código de 0 a  $2^n - 1$  e observa-se cada um desses códigos como um número binário de  $n$ -bit. Projetam-se  $n$  padrões diferentes, em que, no padrão  $k$ , cada linha estará ligada se, e somente se, o  $k$ -ésimo bit desse código for 1. Então, só depois que cada dado *pixel* for observado para  $n$  quadros, a identidade da linha que iluminou aquele *pixel* pode ser determinada.

O tipo de sistema de luz estruturada codificada em tempo anteriormente tem uma desvantagem significativa: o objeto que está sendo digitalizado não deve se mover enquanto os  $n$  quadros codificados estão sendo capturados. Se o objeto se mover, o sistema pode obter um código de linha incorreto e conseqüentemente, dados 3D incorretos.

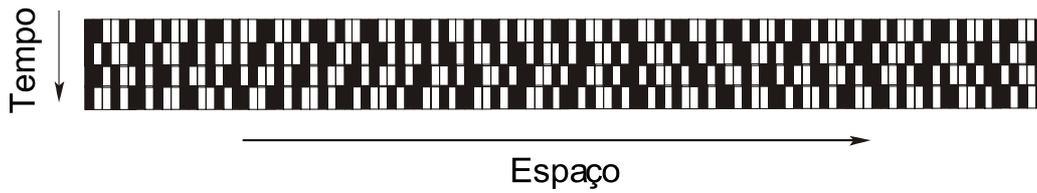
Para permitir movimentação do objeto durante a aquisição, é necessário adicionar rastreamento quadro-a-quadro no sistema de luz estruturada codificada em tempo. Em particular, se o objeto se move suficientemente lento para que as linhas possam ser rastreadas com o passar do tempo, também é possível acumular os bits do código com o passar do tempo, porque eles não provêm do mesmo *pixel* da câmera. Na prática, só se pode rastrear uma região pelas bordas. Em razão disso, Rusinkiewicz [Rusinkiewicz et al. 2002] desenvolveu um sistema baseado na idéia de rastrear as

bordas das linhas projetadas e usar essas bordas para transportar os código pelo tempo.

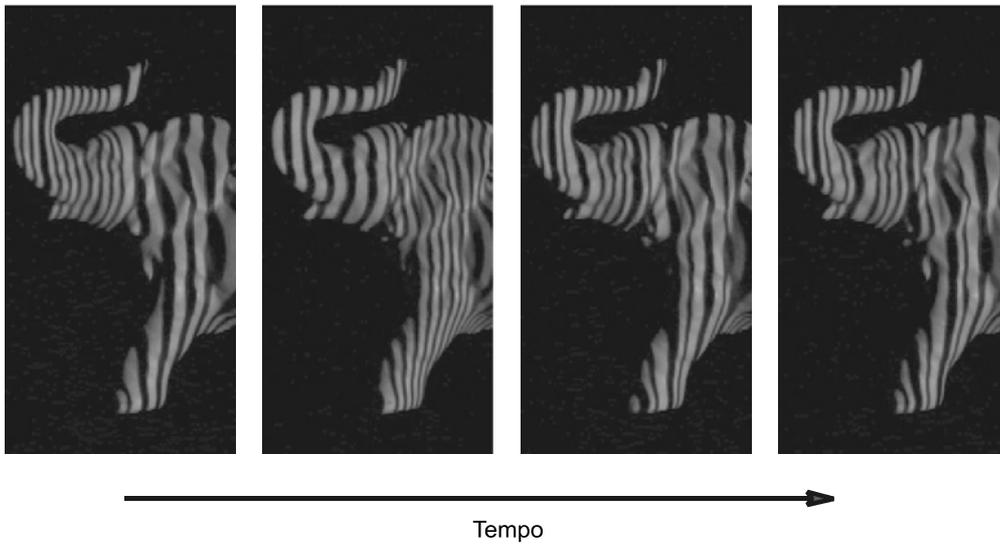
Em um sistema que rastreia as linhas de bordas à medida que elas se movem de quadro a quadro, é necessário um meio de codificar as linhas que seja otimizado para a aplicação. Em primeiro lugar, as linhas devem ser tão estreitas quanto for possível, de forma que o conjunto de bordas que estão sendo rastreadas seja o mais denso possível. Devido à técnica de simples codificação binária resultar em linhas muito largas em alguns quadros, essa codificação não é adequada para uma aplicação que necessite de execução em tempo real. Em segundo lugar, o foco abordado muda das linhas em si para as bordas, permitindo o transporte de informação de maneira mais eficiente: ao invés de obter um bit de informação por quadro, é possível obter dois bits de informação observando o estado ligado/desligado dos dois lados daquela borda. Então, utiliza-se uma codificação que em cada par de linhas adjacentes, tem-se um único código durante o tempo.

O critério final no desenvolvimento dos códigos é a observação de que, já que se está olhando os dois lados de uma borda de linha, essa borda não será visível em todos os quadros, já que as duas linhas de cada lado da borda podem estar ligadas ou desligadas. Por esse motivo foi desenvolvido um código que minimiza o número desses “fantasmas” invisíveis, para maximizar as chances de rastrear essas bordas mesmo se elas desaparecerem. Note-se que esse é só um caso especial da estratégia de se fazer as linhas tão estreitas quanto for possível. É possível desenvolver um código que nunca contém dois fantasmas adjacentes no espaço ou no tempo [Hall-Holt e Rusinkiewicz 2001].

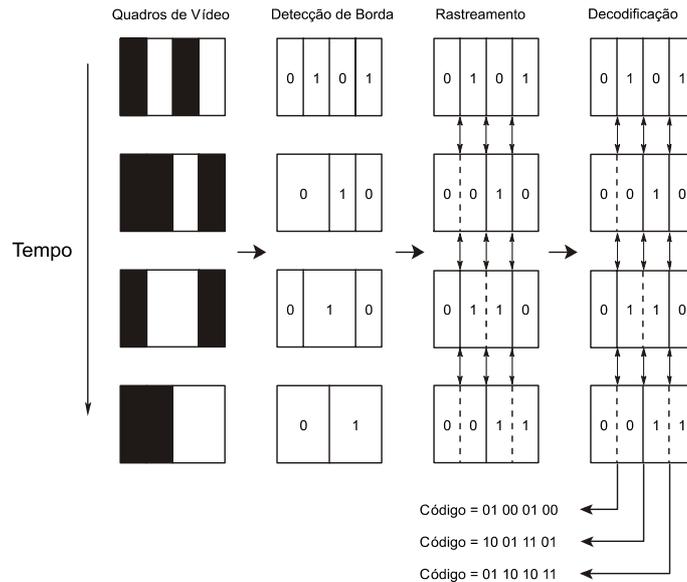
A Figura 2.31 mostra a seqüência de padrão de iluminação utilizada no sistema. Existem quatro padrões distintos, que podem unicamente codificar 110 linhas de borda. O sistema repete esses quadros a 60Hz, captura vídeo com uma câmera NTSC sincronizada (Figura 2.32), faz o rastreamento de bordas e decodificação (Figura 2.33), e retorna a imagem de profundidade de cada ponto instantaneamente. Como eles usam qualquer seqüência de quatro quadros consecutivos para obter a profundidade, há uma latência de 4 quadros para identificar uma nova linha de borda, mas a profundidade é obtida a cada quadro depois disso.



**Figura 2.31:** Uma seqüência de quatro quadros de padrões projetados. Cada padrão tem 110 linhas de borda (111 linhas), com a particularidade que cada linha de borda tem um único código (constituído de um histórico iluminação preto ou branco de cada lado da borda por uma seqüência de quatro quadros).



**Figura 2.32:** Uma seqüência de quadros de vídeo de um elefante iluminado com o código de borda de linha. O projetor fica alternando entre esses quatro padrões a 60Hz e o objeto iluminado é fotografado com uma câmera NTSC sincronizada ao projetor.

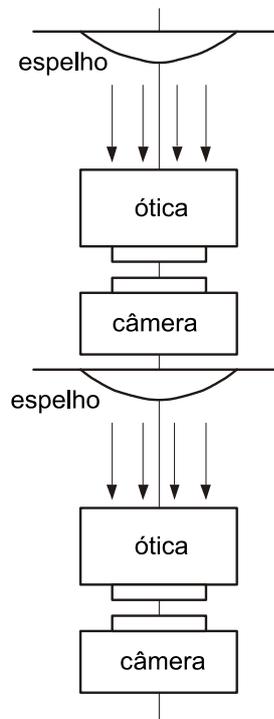


**Figura 2.33:** Rastreamento de linhas de borda. O sistema identifica os limites nos quadros de vídeo, rastreia a linha de borda pelo tempo, e propaga o histórico de iluminação de cada borda por quatro quadros para determinar o código de borda. Note-se que o estágio de rastreamento tem que inferir, na presença de bordas “fantasma” (mostrados em linhas tracejadas) em algumas das regiões P&B com largura dupla. Os padrões projetados são desenhados para minimizar a presença dessas bordas invisíveis e tem a propriedade de que nunca existem dois fantasmas adjacentes no espaço e tempo.

### 2.5.2. Estéreo Omnidirecional

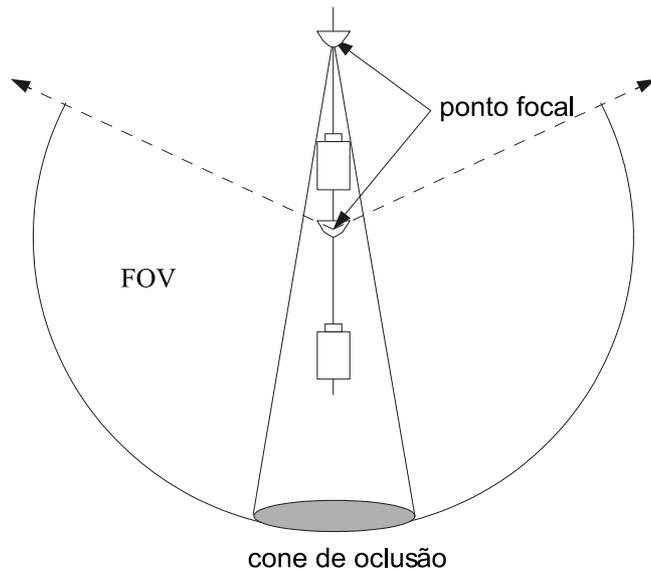
O sistema desenvolvido por Gluckman [Gluckman et al. 1998] utiliza o princípio da estereoscopia em câmeras omnidirecionais e é composto por duas câmeras omnidirecionais alinhadas verticalmente (coaxial Figura 2.34). Cada câmera omnidirecional consiste de uma câmera convencional, ótica telecêntrica, e um espelho parabólico. A ótica telecêntrica aproxima a projeção ortográfica que, junto com um espelho parabólico, forma um sistema com amplo ângulo de imageamento com um único centro de projeção localizado no ponto focal da parábola [Nayar 1997]. As razões para se alinhar os sensores verticalmente são: primeiro, ao se colocar um sensor sobre o outro, as singularidades (onde a profundidade não pode ser calculada perto dos epípolos) ocorrem onde as câmeras se ocluem; segundo, as imagens panorâmicas têm linhas epipolares paralelas correspondentes, o que as leva ao processamento estéreo em tempo-real.

Uma única câmera poderia ser utilizada, colocando-se um espelho menor no topo do outro [Southwell 1996]. Mas, a resolução dos dois panoramas seria significativamente diferente o que conduz a um casamento estéreo menos robusto.



**Figura 2.34:** Diagrama esquemático de uma câmera estereoscópica panorâmica compacta. Duas câmeras omnidirecionais são utilizadas para capturar os dados panorâmicos estereoscópicos. O imageamento omnidirecional é feito por projeção ortográfica de uma superfície parabólica reflexiva. A ótica telecêntrica é usada para fazer a projeção ortográfica enquanto mantém a câmera perto do espelho.

O campo de visão (FOV *field of view*) de cada câmera omnidirecional é determinado pela extensão da parábola. Se a parábola for além do ponto focal, é possível obter um FOV maior que um hemisfério. Cada câmera omnidirecional também produz um cone de oclusão, em que a câmera se “vê” no espelho. A Figura 2.35 mostra o campo de visão compartilhado pelas duas câmeras, em que a profundidade pode ser calculada. Quando as câmeras estão alinhadas verticalmente, o FOV compartilhado é o FOV da câmera inferior menos o cone de oclusão da câmera omnidirecional superior.



**Figura 2.35:** O campo de visão. Estendendo a parábola além do ponto focal, os espelhos parabólicos são capazes de capturar um campo de visão maior que um hemisfério. Um cone de oclusão é criado onde a câmera “se vê”.

O sistema gera panoramas de 360° completos projetando a imagem de entrada sobre um cilindro. Como mostra a Figura 2.36, o mapeamento é feito, primeiro projetando ortograficamente o ponto de imagem  $p$  sobre a parábola, e então projetando para o ponto  $p_0$  com um raio passando pelo ponto focal da parábola.

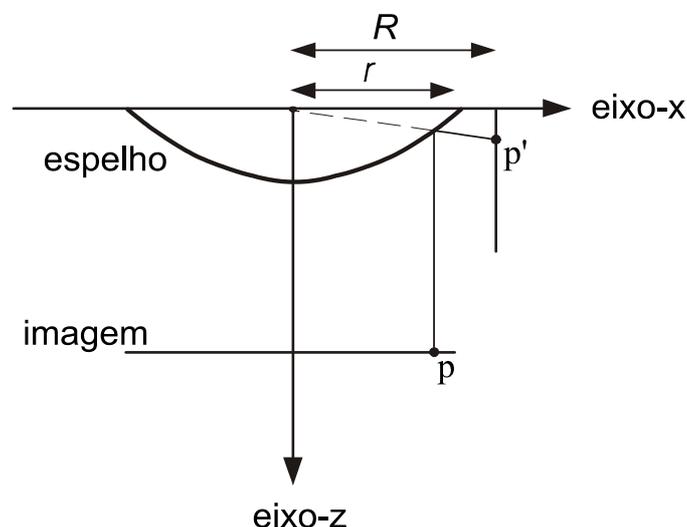
Na geração de panorama, cada *pixel* da imagem panorâmica é retroprojetado para um ponto na imagem de entrada e uma interpolação bilinear a fim de encontrar a intensidade naquele ponto. Usando-se a equação:

$$z = \frac{r^2 - x^2}{2r}, \quad (3.1)$$

para uma parábola (sistema de coordenadas localizado no ponto focal), a função de retroprojeção que relata a altura  $z'$  no panorama, para a distância  $x$  do centro da imagem se torna:

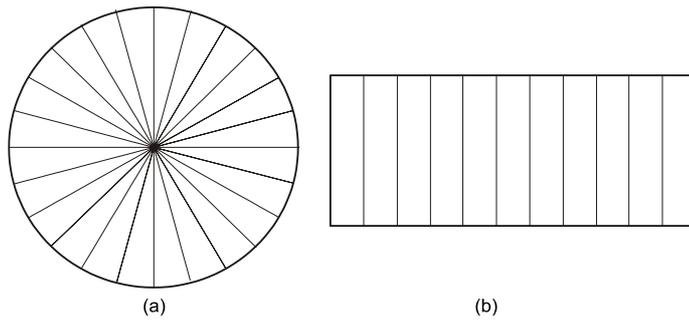
$$z' = \frac{R(r^2 - x^2)}{2rx}. \quad (3.2)$$

O radii da parábola e cilindro é denotado por  $r$  e  $R$ , os dois medidos nos pixels. Enquanto a equação (3.1) só define uma dimensão no panorama a outra dimensão é simplesmente a função de ângulo polar na imagem.



**Figura 2.36:** Mapeando um ponto da imagem para um panorama. Um ponto na imagem  $p$  é mapeado para o ponto  $p'$  no panorama.

Uma parte importante de qualquer sistema estéreo é a restrição epipolar. Essa restrição reduz o problema de encontrar pontos correspondentes para a procura 1-D. Para espelhos parabólicos, pontos correspondentes devem permanecer em cônicas (curvas epipolares). De qualquer forma, quando as parábolas estão verticalmente alinhadas de modo que os vértices e pontos focais sejam coaxiais, as curvas se reduzem a linhas verticais (veja a Figura 2.37). Uma vez que a imagem da parábola é projetada no cilindro (imagem panorâmica), as linhas epipolares se tornam paralelas. Se cada imagem no par estéreo for projetada no cilindro do mesmo tamanho, as linhas epipolares se casarão. Isso implica que  $R$  na equação (3.2) deve ser o mesmo para a função de projeção de cada imagem estéreo para assegurar o casamento das linhas epipolares. É desejável que as linhas epipolares sejam casadas porque métodos eficientes de busca podem ser usados para fazer a correspondência estereoscópica.



**Figura 2.37:** Geometria epipolar. (a) As linhas epipolares para um par de imagens omnidirecionais coaxiais são linhas radiais. (b) Quando as imagens são projetadas sobre um panorama as linhas epipolares se tornam paralelas.

Uma vez que a correspondência entre a imagem de pontos tenha sido estabelecida, o cálculo de profundidade é feito por triangulação. Como mostra a Figura 2.38, cada ponto da imagem correspondente é a projeção ortográfica de um ponto na parábola, que pode ser definido pelos ângulos  $\alpha$  e  $\beta$ . Pela lei dos senos, a profundidade  $d$  é calculada como:

$$d = \frac{\cos(\beta)}{\sin(\alpha - \beta)} b, \quad (3.3)$$

onde  $b$  é a linha base do sistema estéreo. Permitindo que  $x$  e  $x'$  sejam as distâncias dos pontos da imagem  $p$  e  $p'$  do centro das parábolas na imagem, então os ângulos  $\alpha$  e  $\beta$  são determinados por,

$$\tan(\alpha) = \frac{r^2 - x^2}{2rx} \quad (3.4)$$

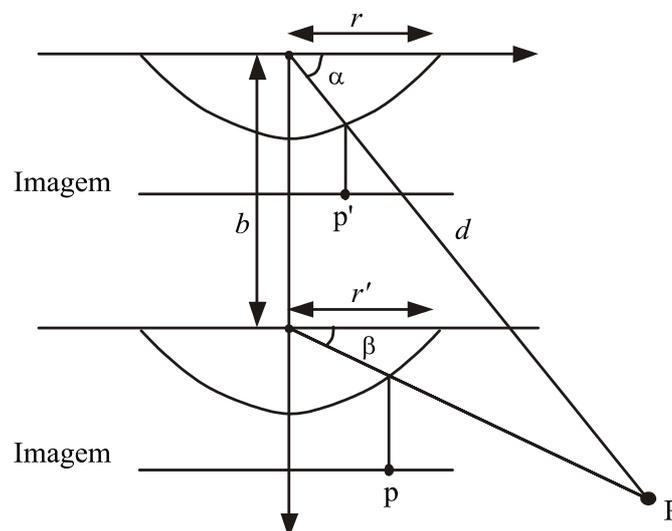
e

$$\tan(\beta) = \frac{r'^2 - x'^2}{2r'x'}, \quad (3.5)$$

onde  $r$  e  $r'$  são os raddis das parábolas. Note-se que em contraste com o estéreo tradicional, o cálculo da profundidade é independente do comprimento focal da câmera (devido à projeção ortográfica).

Alternativamente, a distância pode ser definida como uma função de disparidade na imagem panorâmica. Se  $x$  e  $x'$  são as alturas dos pontos correspondentes no topo e na parte inferior dos panoramas, a disparidade  $x - x'$  é inversamente relacionada a profundidade como:

$$\left( \frac{x - x'}{\sqrt{R^2 + x^2}} \right) b = \frac{1}{d} \quad (3.6)$$



**Figura 2.38:** Triangulação e computação de profundidade. Dado um par de pontos correspondentes  $p$  e  $p'$  das imagens, o ponto de cena  $P$  é encontrado por triangulação.

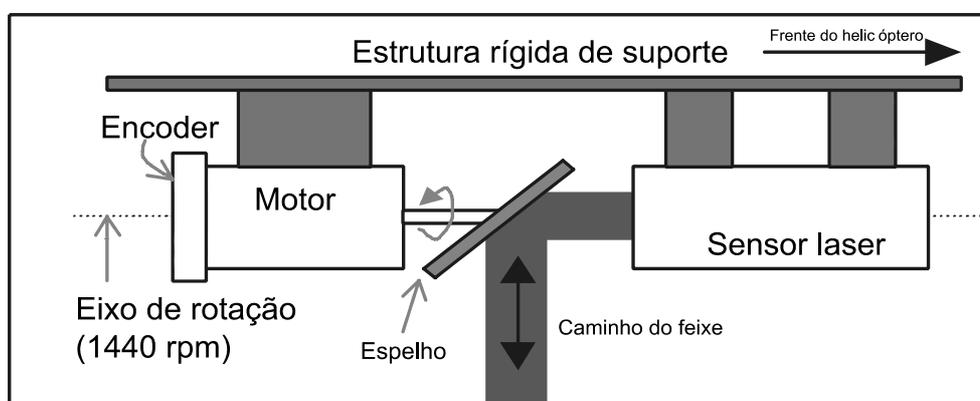
### 2.5.3. Tempo de Vôo Laser

No trabalho de Miller [Miller et al. 1999] um sensor de distância laser baseado no método de tempo de vôo TOF e um helicóptero autônomo foram usados em uma missão no ártico canadense que requeria a construção de um denso mapa topológico da cratera Houghton nas Ilhas Devon para estudo de ambientes análogos a Marte para cientistas da NASA.

Muitos trabalhos foram feitos com sensoriamento ativo de estruturas 3D por veículos aéreos e terrestres. A Lockheed Martin está desenvolvendo um sistema de mapeamento baseado em avião, que constrói mapa de elevação digital de grandes regiões usando radar de abertura sintética [Malliot 1996]. Esse sistema emprega o uso de veículos aéreos de grande altitude e tem por objetivo gerar mapas com pouca precisão (3m) de grandes áreas. Em contraste, o sistema de Miller [Miller et al. 1999] requer vôos mais próximos da área de interesse para criar mapas 3D de alta precisão (20cm). Um sistema de helicóptero autônomo foi especificamente desenhado para essas manobras aéreas, incluindo decolagem totalmente autônoma, rastreamento de caminho aéreo, vôo estacionário preciso (<20 cm) e aterrissagem [Amidi et al. 1998]. Essas características básicas permitem que o helicóptero seja usado como uma plataforma de sensoriamento de alta mobilidade; fornecendo acesso a localizações confinadas e remotas sem colocar pilotos humanos em risco.

O sistema de mapeamento de bordo faz o sensoriamento de distância do terreno com um scanner laser planar, que é integrado com os computadores do controle de vôo. O sistema integrado mede coordenadas 3D referenciadas a Terra em até 6000 pontos de terreno por segundo, com um alcance de 100 metros e precisão de 20 cm em relação à sua posição no espaço.

O desenho do scanner é mostrado na Figura 2.39. Ele possui um mecanismo com um único grau de liberdade, que faz a aquisição do ambiente refletindo o feixe laser do sensor de distância em um espelho rotativo de alumínio montado a 45 graus. O motor gira o espelho em um ciclo completo (360°) e o laser é redirecionado para fazer a aquisição de um plano perpendicular ao eixo de rotação.

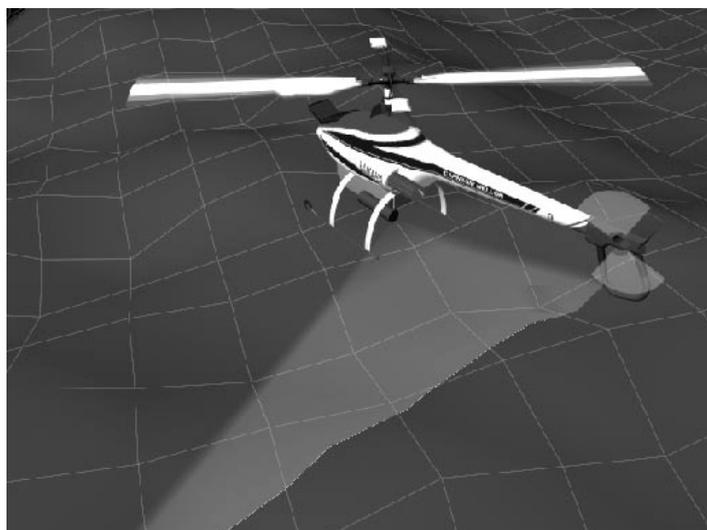


**Figura 2.39:** Scanner laser planar

O sensor utiliza o princípio do tempo-de-vôo (TOF time-of-flight) e tem precisão de 2 centímetros e alcance de 100 metros.

O sistema de varredura possui um motor equipado com um *encoder* de 2000 posições por rotação, que possibilita uma resolução de 0.18 graus, para controlar a posição do espelho. O laser é sincronizado para fazer somente uma leitura por pulso do *encoder*. O mecanismo de varredura é montado debaixo da fuselagem do helicóptero de maneira que o plano de varredura resultante é perpendicular à direção de vôo para frente (Figura 2.40). Com essa configuração, a combinação do movimento de varredura e o deslocamento para frente do helicóptero fazem o esquadramento do solo. Essa configuração para aquisição aérea é referenciada como *across-track* ou *whiskbroom scanning* [Samberg 1997].

Nessa configuração, o laser faz a varredura por 360 graus completos e parte desse tempo quando o laser é direcionado para o céu, nada significativo é medido. Essa ineficiência foi aceita em favor da simplicidade e baixo peso.



**Figura 2.40:** Mapeamento aéreo com o laser planar.

## 2.6. Considerações Finais

Nesse capítulo foram apresentadas sistemas de sensoriamento de distância, conceitos básicos de reconstrução, técnicas de filtragem espacial e detecção de

bordas, aplicações e sistemas de aquisição 3D, utilizando variações das técnicas tradicionais apresentadas na seção 2.1.

O sistema de aquisição de informações volumétricas usando o método de triangulação laser e campo de visão variável será apresentado no capítulo seguinte. Esse método é uma variação do método de triangulação laser que é um caso particular do sensoriamento ativo baseado em luz estruturada (seção 2.1.1). Também serão apresentadas as fases de desenvolvimento e implementações de sistemas de aquisição 3D baseados em triangulação laser simplificados.

## Capítulo 3

# Projeto e Implementações de um Sistema de Aquisição 3D

Neste capítulo será apresentado o modelo de aquisição por triangulação laser utilizado nas implementações deste trabalho. Em seguida a descrição do modelo de aquisição volumétrica com campo de visão variável aqui idealizada e as fases de seu desenvolvimento.

Esse trabalho iniciou-se com a criação de um modelo virtual com todos os parâmetros conhecidos. Posteriormente, três protótipos de sistemas de aquisição 3D foram implementados, sendo que cada um deles apresentava avanços sobre o hardware predecessor.

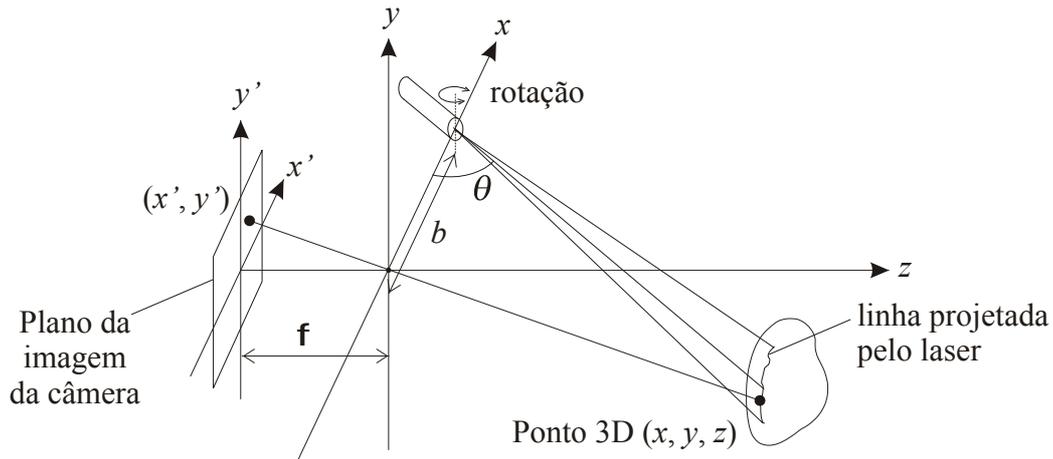
O sistema de varredura do primeiro protótipo implementado não possuía redutor mecânico. Nessa implementação houve uma familiarização com os sistemas de controle do motor.

Ao segundo protótipo foi adaptado um redutor mecânico de 3,9:1 e utilizou-se um laser de melhor qualidade. Essas adaptações implicaram em sua reconstrução quase total.

O terceiro protótipo implementado utiliza um motor de precisão acoplado a um redutor mecânico de 879:1. O *backlash* (folga entre o começo e o final do redutor) desse sistema é zero, o que propicia maior precisão, garantindo o posicionamento exato. A descrição de implementação apresentada será basicamente sobre o terceiro protótipo, havendo diferenciação quando se julgar necessário.

### 3.1. Aquisição 3D por Triangulação Laser

Esse trabalho baseia-se no método de triangulação laser, que consiste na projeção de um padrão de luz, no caso, uma linha vertical projetada pelo laser, sobre o objeto. Essa projeção é capturada pela câmera e a distância até o objeto é calculada por trigonometria, dado um conhecimento a priori das posições do laser e da câmera [Sinha e Jain 1994]. Uma configuração de sistema de triangulação é mostrada na Figura 3.1.



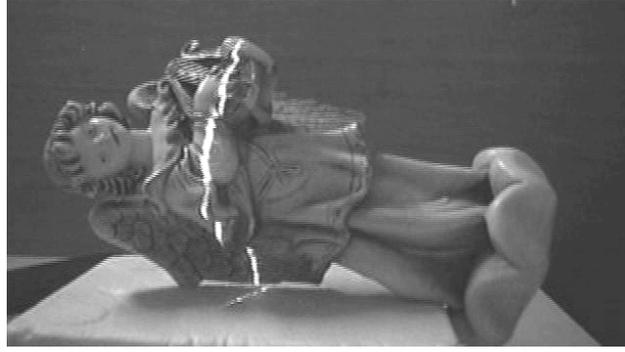
**Figura 3.1:** Geometria do sistema de aquisição por triangulação laser.

O centro das lentes está localizado na origem, com distância focal  $f$  até o plano da imagem e a distância entre o projetor e a câmera é a linha de base  $b$ . A projeção do laser forma o ângulo  $\theta$  em relação à linha base [Tiziani 1997] e o ponto 3D  $(x, y, z)$  no espaço real é projetado no pixel da imagem  $(x', y')$ .

O valor de  $\theta$  é conhecido pelo sistema de varredura e a distância focal  $f$ , é determinada pela calibração da câmera. A linha de base  $b$  é calculada posicionando-se o laser de maneira que sua projeção incida no centro da imagem, apontando para um objeto a uma distância conhecida  $z$ . Dessa forma tem-se que:

$$b = \frac{z}{\tan \theta} \quad (3.1)$$

Inicialmente é feita uma varredura girando-se o laser que projeta uma linha vertical, capturando imagens e armazenando os respectivos valores de  $\theta$  obtidos pelo sistema de varredura para cada imagem. Para cada imagem capturada, determinam-se os valores de  $x'$  e  $y'$  onde o laser foi refletido e então os pontos nas coordenadas reais são calculados através das equações (2.5), (2.6) e (2.7). A Figura 3.2 mostra um quadro da aquisição de vídeo onde a projeção do laser está presente. À medida que a linha projetada pelo laser se desloca para a esquerda, a distância  $z$  é menor.



**Figura 3.2:** Quadro de aquisição de vídeo.

### 3.2. Sistema de Aquisição de Informações Volumétrica Usando Método de Triangulação Laser e Campo de Visão Variável

O sistema de aquisição proposto nesse trabalho, difere dos sistemas tradicionais de triangulação em sua varredura e, por ter o campo de visão variável. Nos sistemas tradicionais a varredura é efetuada apenas pelo laser enquanto que nesse sistema a varredura é feita pela câmera e pelo laser. As equações originais de triangulação (2.5), (2.6) e (2.7), foram modificadas para admitirem rotação da câmera. Dessa forma tem-se:

$$x = \frac{x' \left( \frac{b \cdot \tan \theta \cdot \tan(90 - \alpha)}{\tan(90 - \alpha) + \tan \theta} \right) / \text{sen}(90 - \alpha)}{f \cot(\theta + 90 - \alpha) - x'} \quad (3.2)$$

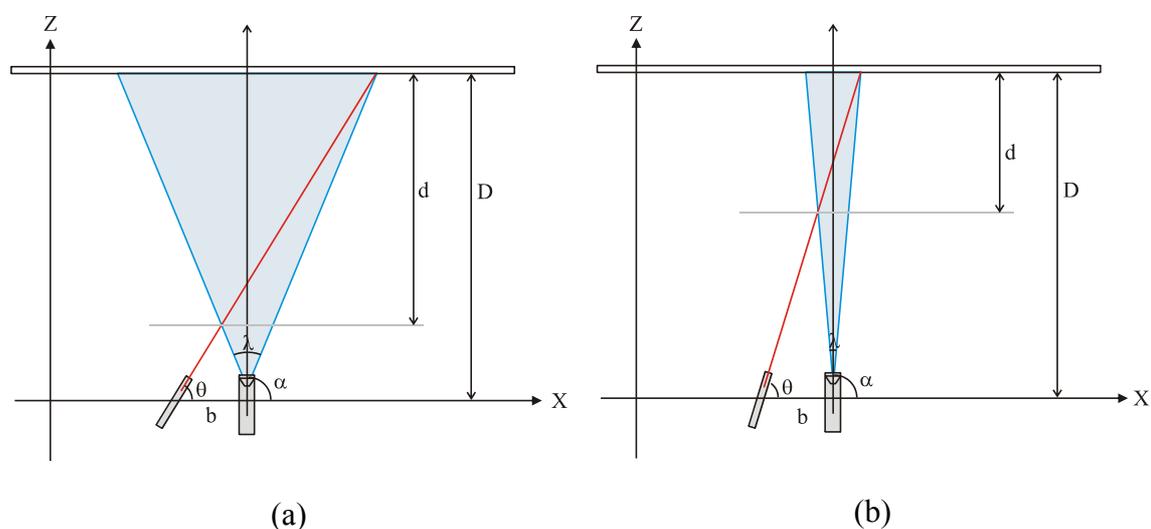
$$y = \frac{y' \left( \frac{b \cdot \tan \theta \cdot \tan(90 - \alpha)}{\tan(90 - \alpha) + \tan \theta} \right) / \text{sen}(90 - \alpha)}{f \cot(\theta + 90 - \alpha) - x'} \quad (3.3)$$

$$z = \frac{f \left( \frac{b \cdot \tan \theta \cdot \tan(90 - \alpha)}{\tan(90 - \alpha) + \tan \theta} \right) / \text{sen}(90 - \alpha)}{f \cot(\theta + 90 - \alpha) - x'} \quad (3.4)$$

onde o comprimento focal da câmera é  $f$  e suas lentes estão localizadas na origem. O ângulo da câmera em relação à linha de base  $b$  é  $\alpha$ .

O sistema de varredura deve cobrir toda a superfície da cena desejada. Para tanto, deve haver um controlador de movimento e um motor com *encoder*, para cada um dos três eixos de varredura descritos a seguir. A câmera e o laser possuem controles independentes de rotação no eixo- $Y$  (vertical) e giram em conjunto no eixo- $X$  (horizontal). O controlador de movimento comanda o posicionamento do motor. O sistema também controla o campo de visão da câmera (*field of view* - FOV).

Inicialmente é feita uma aquisição prévia com baixa resolução e FOV no máximo com a câmera apontando para o centro da cena para determinar a distância máxima  $D$  (entre a câmera e o objeto de interesse). Para a determinação dessa distância máxima a varredura é feita apenas pelo laser, mantendo a câmera parada. Em seguida o FOV é ajustado de acordo com a definição desejada pelo usuário (quanto menor for o FOV maior a precisão e menor a variação de profundidade) (Figura 3.3),



**Figura 3.3:** Em (a) o FOV  $\lambda$  e o campo de profundidade  $d$  são maiores que em (b), mas a precisão é maior em (b).

e então é feita a varredura no eixo- $Y$  (vertical) girando a câmera e o laser em conjunto (Figura 3.4 (a),(b) e (c)) de modo que o ângulo do laser fique relacionado com o ângulo da câmera, como mostra a equação (4.5).

$$\theta = \tan^{-1}\left(\frac{D}{D/(\tan(\alpha - (\lambda/2)) + b)}\right) \quad \text{para} \quad \frac{D}{D/(\tan(\alpha - (\lambda/2)) + b)} \geq 0$$

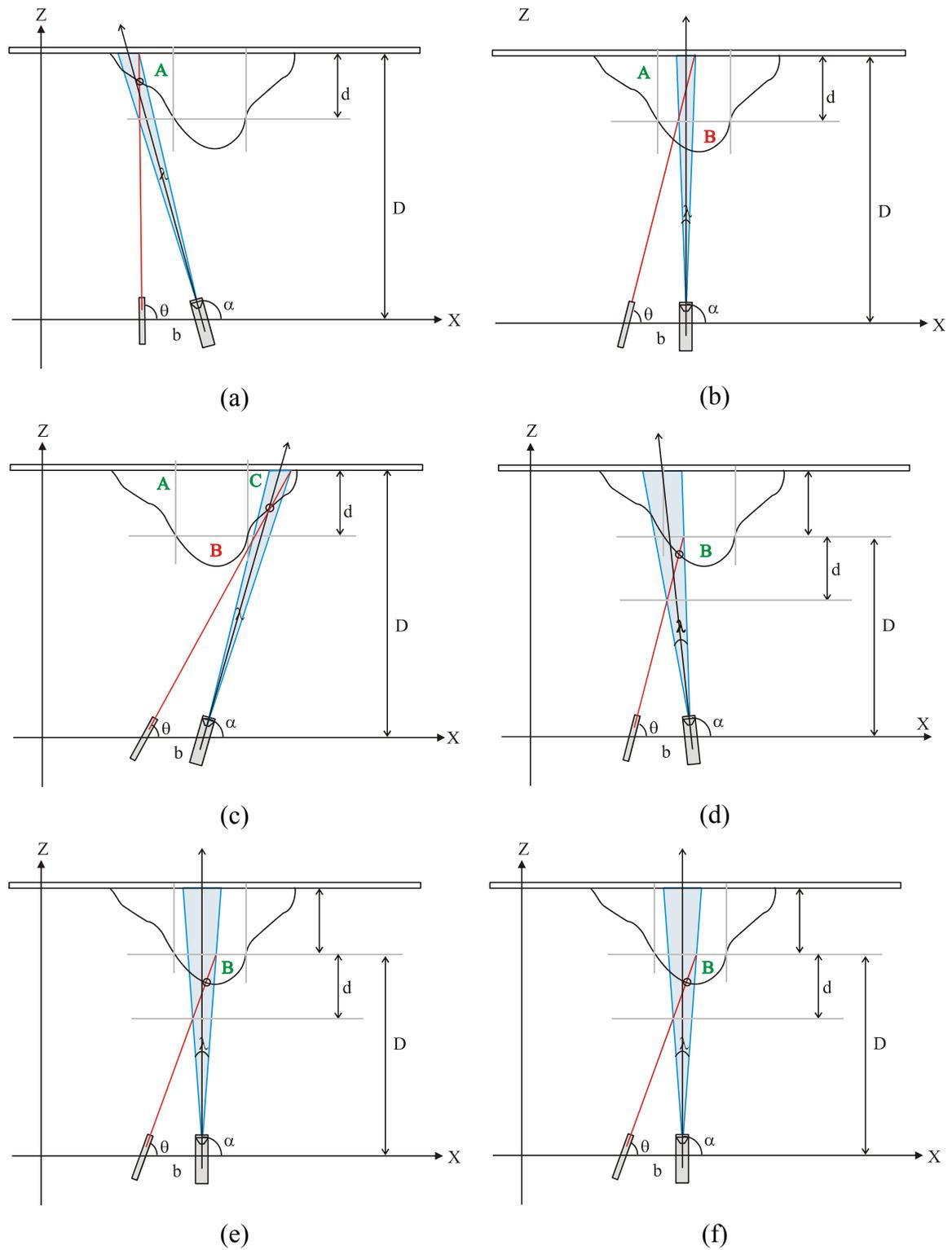
$$\theta = \tan^{-1}\left(\frac{D}{D/(\tan(\alpha - (\lambda/2)) + b)}\right) + 180 \quad \text{para} \quad \frac{D}{D/(\tan(\alpha - (\lambda/2)) + b)} < 0$$
(3.5)

onde  $b$  é a linha de base entre a câmera e o laser, e  $\lambda$  é o FOV. A movimentação é feita dessa forma para que o laser seja projetado nos *pixels* mais à direita da imagem (em um sistema com o laser do lado esquerdo e a câmera do lado direito). Como a distância máxima  $D$  é conhecida, o sistema nunca vai precisar fazer a aquisição em lugares mais distantes.

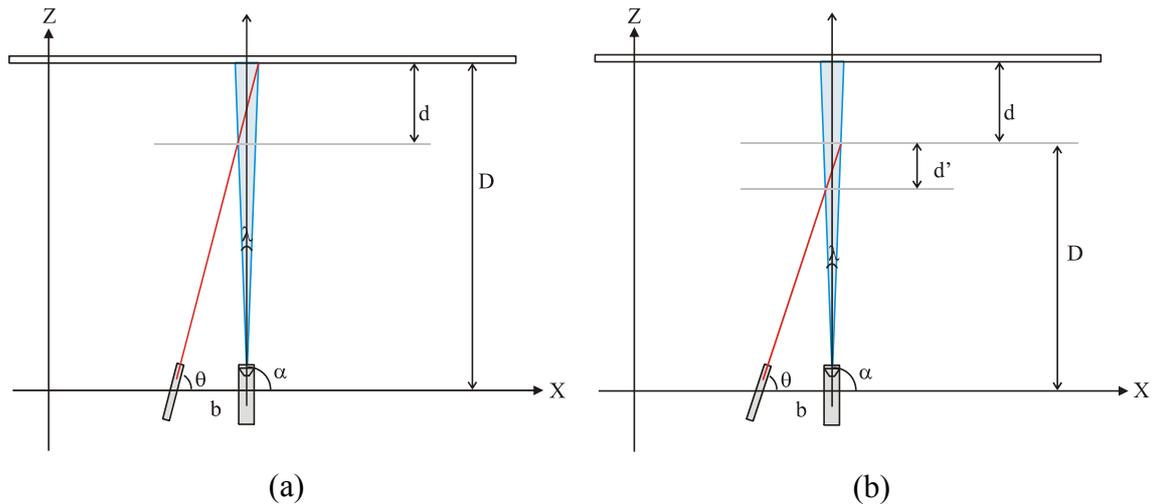
Após terminar a varredura de uma linha o sistema gira o conjunto laser-câmera no eixo- $X$  e continua fazendo a varredura no eixo- $Y$  até cobrir toda a área da superfície. Em seguida o sistema determina as regiões que excedem a variação de profundidade  $d$  (região B da Figura 3.4(a),(b) e (c)) e atualiza a distância máxima  $D$  para  $D-d$  ajustando o FOV para que o campo de profundidade continue sendo igual ao  $d$  anterior e ajusta o ângulo do laser  $\theta$  para que o laser continue no limite direito da imagem (Figura 3.4(d)). Se esse ajuste não fosse feito, a variação de profundidade seria cada vez menor (Figura 3.5).

O sistema refaz a aquisição no eixo- $Y$  nas áreas mais próximas que  $D-d$  (região B da Figura 3.4, (d), (e) e (f)). Esse processo se repete até que não haja mais regiões fora do campo de profundidade.

A aquisição é feita por camadas, aproveitando melhor a matriz de CCD (*Charged Couple Device*), obtendo-se maior precisão nas leituras de distância, maior alcance e maior variação de profundidade, além de ser necessário menor número de aquisições do que os sistemas de triangulação laser convencionais, possibilitando, assim, maior versatilidade.



**Figura 3.4:** Seqüência de aquisição inicial (a) (b) (c). Ajuste do FOV e ângulo do laser (d). Segunda aquisição (d) (e) (f).



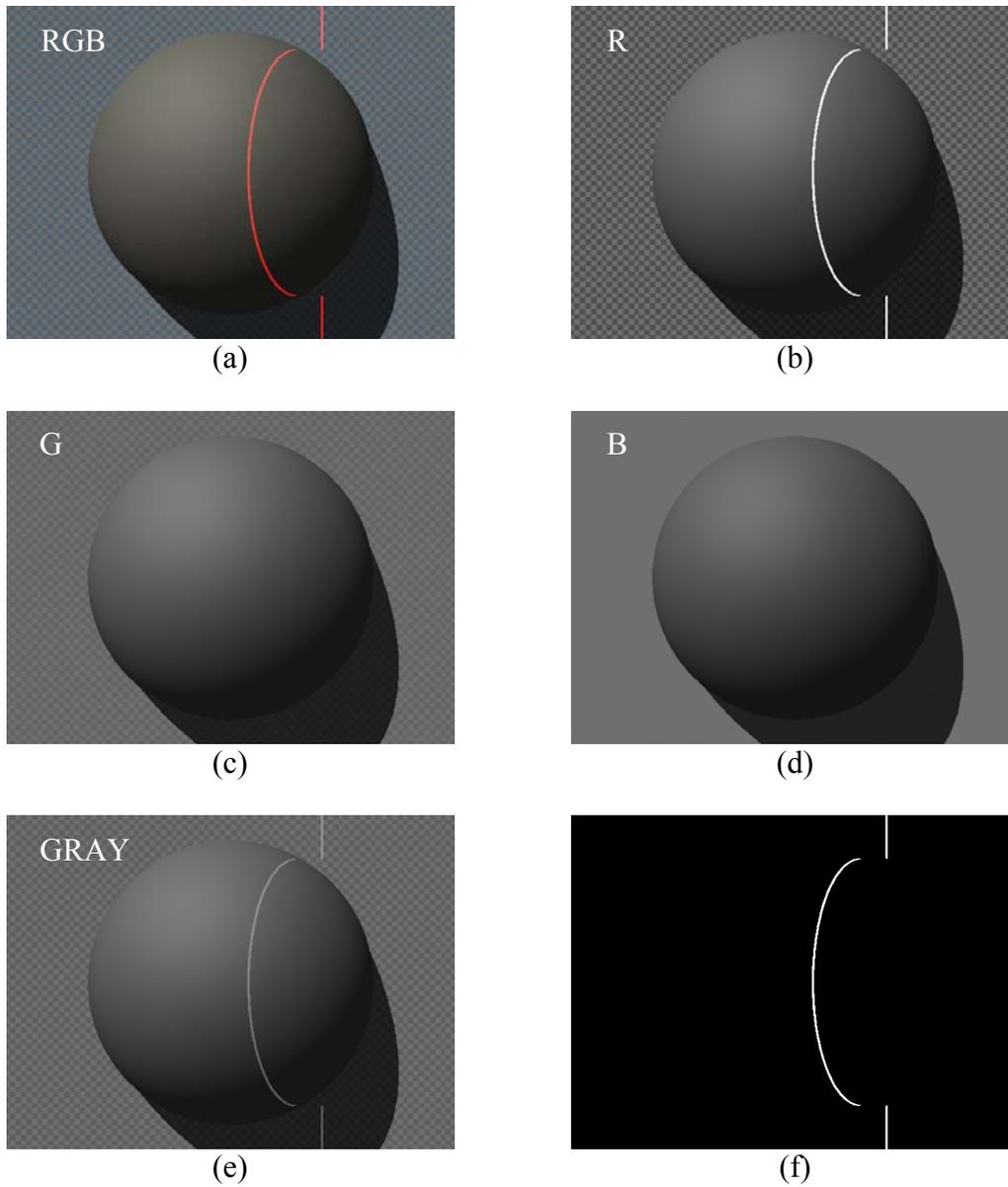
**Figura 3.5:** (a) Sistema ajustado para a primeira aquisição. (b) Segunda aquisição, ajuste do laser sem o ajuste do FOV ( $d' < d$ ).

### 3.3. Modelo Virtual

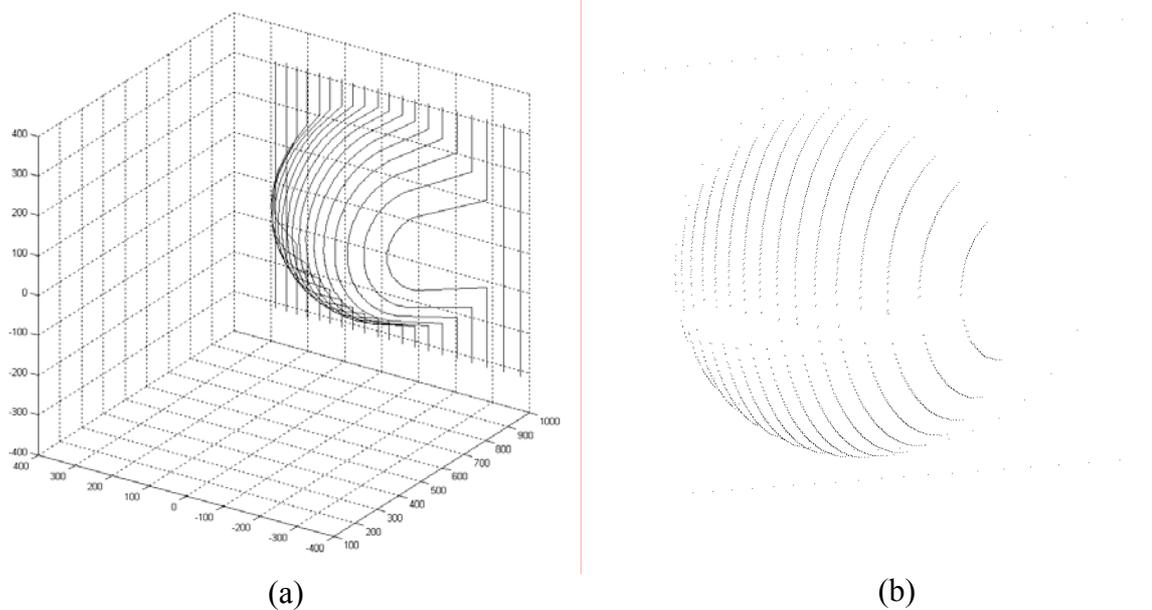
Nessa fase foi criado um sistema de aquisição 3D virtual com todos os parâmetros conhecidos, que foi usado como base para teste de algoritmos de processamento de imagens e estimação de distância, provendo condições ideais, livres de ruídos e de necessidade de calibração.

Utilizando a linha de base de 400 mm e comprimento focal de 8.6 mm, foi gerada digitalmente uma seqüência de imagens. A Figura 3.6 (a) mostra um quadro de seqüência de imagens geradas. Para segmentar a projeção do laser da imagem bastou fazer uma limiarização do canal vermelho (Figura 3.6(b)) como mostra a Figura 3.6 (f). Pode-se observar que a projeção do laser não está presente nos canais verde e azul (Figura 3.6(c) e (d)). Na imagem de intensidade (Figura 3.6 (e)), o laser não se destaca tanto quanto na imagem do canal vermelho (Figura 3.6 (b)).

Para estimação de distância aplicou-se as equações de triangulação (2.5), (2.6) e (2.7) em uma seqüência de imagens segmentadas, obtendo-se o objeto 3D apresentado na Figura 3.7.



**Figura 3.6:** (a) Imagem em RGB, (b) canal vermelho de (a), (c) canal verde de (a), (d) canal azul de (a), (e) intensidade de (a) e (f) limiarização de (b).



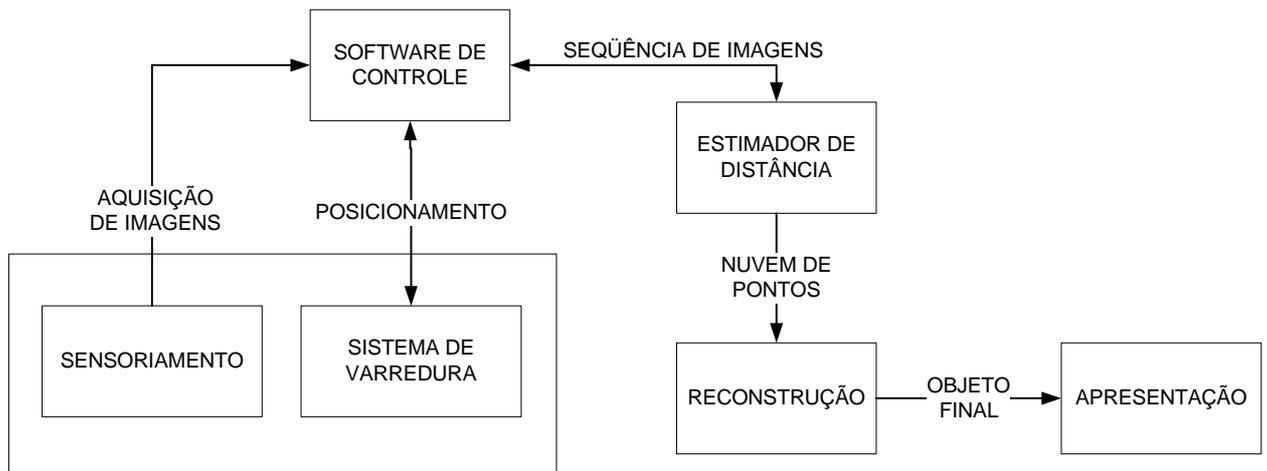
**Figura 3.7:** Objeto 3D gerado pela seqüência de imagens segmentadas. (a) Escala de distância em (mm). (b) Somente os pontos vizinhos com  $x$  diferente são apresentados.

### 3.4. Implementação do Sistema de Aquisição 3D

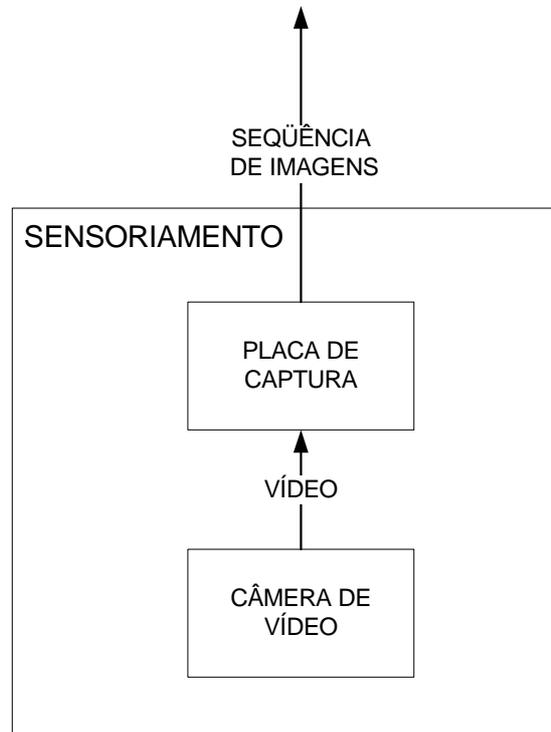
A estrutura básica do sistema de aquisição 3D é constituída de vários módulos, sendo que alguns são constituídos de hardware, e outros de software. Os módulos de hardware são responsáveis pelo sensoriamento e varredura, enquanto os módulos de software são responsáveis pelo controle dos módulos de hardware, estimação de distância, reconstrução e apresentação. A Figura 3.8 ilustra a estrutura básica do sistema de aquisição 3D.

#### 3.4.1. Sensoriamento

O sensoriamento é composto de uma câmera de vídeo e uma placa de captura que são responsáveis pela aquisição de uma seqüência de imagens que são utilizadas para estimação de distância. Nesse trabalho foi utilizada uma câmera NTSC com comprimento focal de 4.2~50.4 mm e CCD de 1/3". A placa de captura utilizada possui resolução de 640x480 *pixels* a 60 campos por segundo. A Figura 3.9 ilustra o módulo de sensoriamento.



**Figura 3.8:** Estrutura básica do sistema de aquisição 3D.

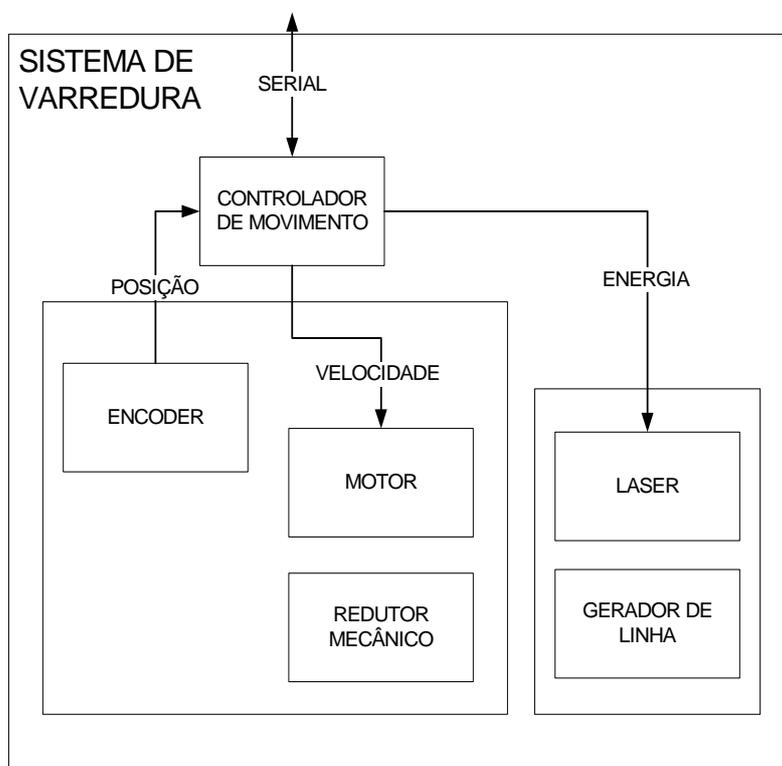


**Figura 3.9:** Módulo de sensoriamento.

O comprimento focal informado na documentação da câmera, assim como tamanho do CCD são aproximações e não puderam ser utilizados diretamente como parâmetros no sistema de estimação de distância.

### 3.4.2. Varredura

O módulo de varredura realiza o esquadramento da projeção do laser por toda a superfície da cena de interesse. Esse módulo é constituído de um hardware controlador de movimento, um motor com *encoder* e redutor mecânico, e laser com gerador de linha. O controlador de movimento comanda o posicionamento, a velocidade e a aceleração do motor, pela interface RS232-C. Além disso o controlador do motor também é responsável pelo acionamento do laser. O gerador de linha é uma lente cilíndrica que, ao ser transpassada pelo feixe de laser produz um padrão de linha. A Figura 3.10 ilustra o módulo de varredura.



**Figura 3.10:** Módulo de varredura.

No primeiro e segundo protótipos implementados, utilizou-se um motor danificado que não possuía atrito constante no período de uma revolução, o que dificultava a calibração dos termos proporcional e integral (seção 4.4.4). Os *encoders* utilizados nos três protótipos eram iguais.

O terceiro protótipo utiliza um motor ímã permanente DC acoplado a um redutor mecânico integrado e a um *encoder*. As Tabela 3.1, Tabela 3.2 e Tabela 3.3 mostram as características do motor, redutor mecânico e *encoder* respectivamente.

**Tabela 3.1:** Informações técnicas do motor.

Característica	Valor	Unidade
Tensão	12	V
Corrente	570	mA
Torque máximo recomendado	5	mN.m
Torque de fricção	0.2	mN.m
Constante de velocidade	657	rpm/V
Resistência interna	8,71	Ohm
Constante de torque	14.50	mN.m/A
Aceleração angular	0,074	rad/s <sup>2</sup>
Constante de back-EMF	0,001520	V/rpm
Velocidade máxima recomendada	8000	Rpm

**Tabela 3.2:** Informações técnicas do redutor mecânico.

Característica	Valor	Unidade
Taxa de redução	879:1	
Velocidade máxima recomendada	4000	rpm
Backlash	0	°
Eficiência	26	%
Torque de saída para operação contínua	270	mN.m
Torque de saída para operação intermitente	400	mN.m
Direção de rotação reversível	=	

**Tabela 3.3:** Informações técnicas do *encoder*.

Característica	Valor	Unidade
Linhas por revolução	512	
Sinal de saída, onda quadrada	2	Canais
Alcance de frequência, até	160	kHz
Largura de pulso	180±45	°
Deslocamento de fase, canal A para B	90±45	°

Para comandar o motor nas três implementações, foi utilizado um controlador de movimento com as seguintes características apresentadas na Tabela 3.4.

A projeção de luz do primeiro protótipo, era efetuada utilizando-se um diodo laser, adaptado de um laser apontador de baixa qualidade que, possuindo um espalhamento muito grande, prejudicava a segmentação da imagem. No segundo e terceiro protótipos, foi utilizado um diodo laser de melhor qualidade, de 5 mW 650 nm e gerador de linha de 60°.

**Tabela 3.4:** Informações técnicas do controlador de movimento.

Característica	Valor	Unidade
Tensão	12 ... 28	V
Corrente máxima	5	A
Corrente máxima de pico	10	A
Velocidade	10 ... 30000	rpm
Frequência máxima de sinal de <i>encoder</i>	200	kHz
Porta serial RS232-C	1200, 2400, 4800, 19200	Baud
Memória de programação Serial EEPROM	7936	Bytes
Três entradas de nível lógico	baixo 0... 0,5 / alto 4 ... 30	V
Uma saída	Tensão de entrada / 30 mA	V

É recomendável que os componentes de hardware não excedam os limites de suas especificações. Assim, alguns cálculos devem ser efetuados para se saber se as exigências da aplicação, excedem as limitações do hardware. Dessa maneira o torque do motor é definido em função da corrente pela equação:

$$M_e = K_M \cdot I_e \quad (3.6)$$

onde  $M_e$  é o torque em mN.m,  $K_M$  é a constante de torque e  $I_e$  é a corrente. No sistema de varredura ocorre um pico de corrente de 0.098 A, que corresponde ao torque de 1.421 mN.m no motor. Ao passar pelo redutor mecânico esse torque é multiplicado pelo inverso da taxa de redução 879 e pela eficiência do redutor 0,26. Então, tem-se que o pico de 0,098 A, representa o torque de saída de 324.75 mN.m e quando o motor está estabilizado a 4000 rpm, sua corrente é de 0,075 A que corresponde ao torque de 248.53 mN.m, sendo que esses valores estão abaixo dos limites de torque do redutor mecânico.

A velocidade máxima do *encoder* é definida pela seguinte equação:

$$\text{Velocidade (rpm)} = f \text{ (Hz)} \times 60/N \quad (3.7)$$

onde N é o número de linhas por volta do *encoder*. A frequência máxima do *encoder* utilizado é de  $16 \cdot 10^4$  Hz e o número de linhas por volta é de 512, então aplicando a equação (4.7) tem-se que a velocidade máxima do *encoder* é de 18750 rpm. Sendo que esse valor é superior à velocidade máxima utilizada nessa aplicação (4000 rpm).

Para determinação do posicionamento em ambos os sentidos de rotação e são necessários dois canais no *encoder* e sua resolução é de 2048 incrementos por rotação (4x o número de linhas do *encoder*).

A frequência máxima suportada pelo controlador de movimento para o sinal de *encoder* é de 200 kHz. Isso significa que a velocidade máxima que ele suporta é dada pela seguinte equação:

$$\text{Resolução do } encoder * \text{ velocidade máxima da aplicação} / 60 \quad (3.8)$$

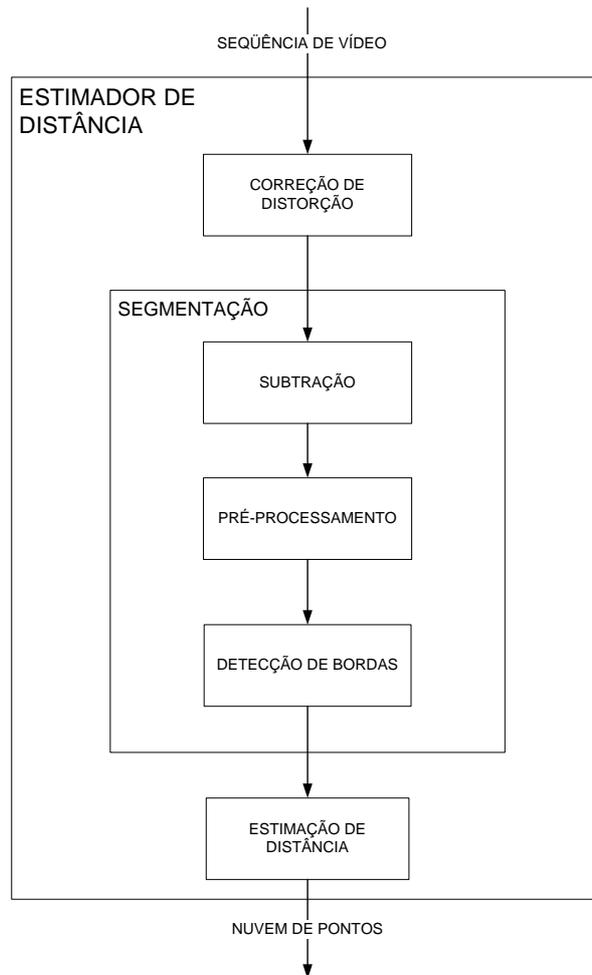
A resolução do *encoder* é de 2048 pulsos e a velocidade máxima da aplicação é de 4000 rpm. Assim, tem-se que a frequência máxima de sinal de *encoder* gerada por essa aplicação é de 136.53 kHz, que é inferior à capacidade máxima do controlador de movimentos (200 kHz).

Dessa forma conclui-se que o hardware utilizado se adequa às exigências da aplicação.

### **3.4.3. Estimador de distância**

O módulo de estimação de distância tem como entrada a seqüência de imagens com as projeções do laser, capturada pelo módulo de sensoriamento e esquadrihada pelo módulo de varredura. Cada imagem da seqüência passa por um processo de correção de distorções, causadas pelos parâmetros intrínsecos da câmera e são, então, processadas digitalmente para que a projeção do laser possa ser segmentada da imagem. Finalmente a distância é calculada a partir da imagem segmentada.

Uma nuvem de pontos com coordenadas  $(x, y, z)$  é gerada após todas as imagens terem suas distâncias calculadas. A Figura 3.11 ilustra o módulo de estimação de distância.

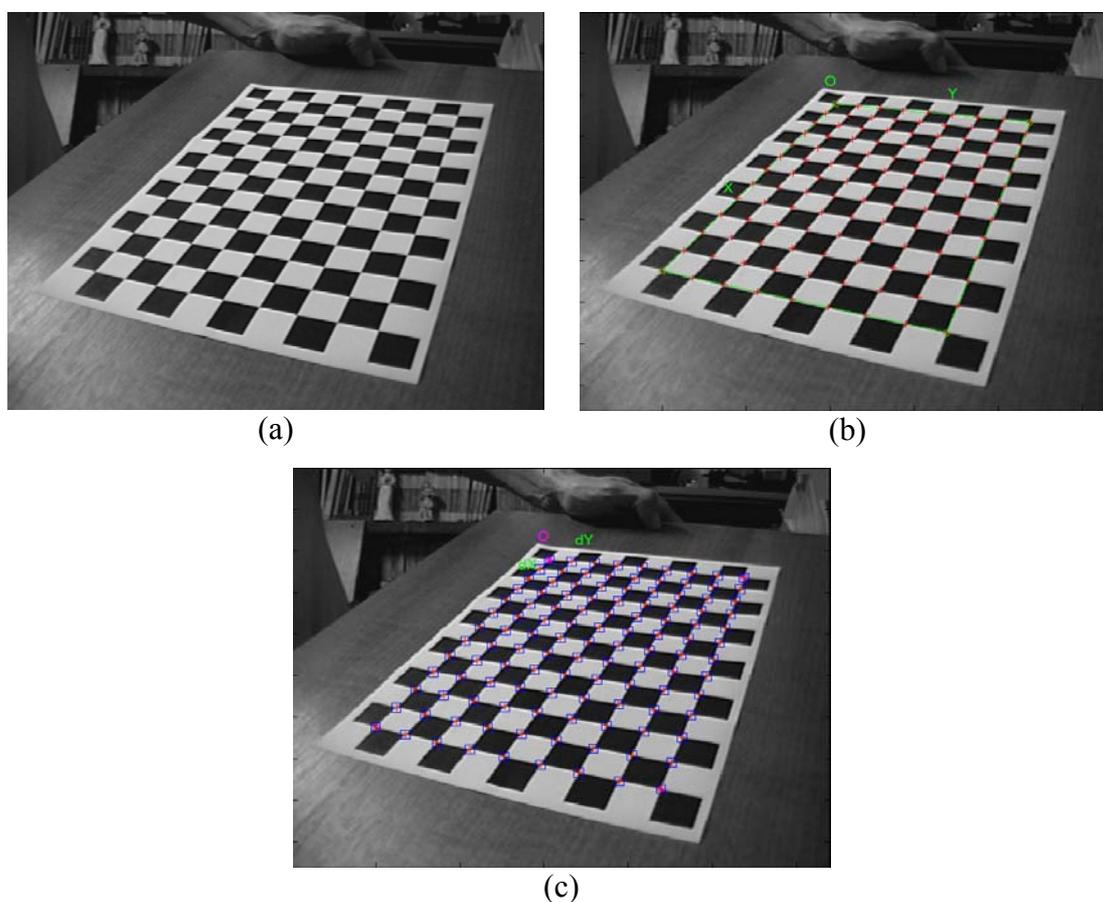


**Figura 3.11:** Módulo de estimação de distância.

### **Correção de distorção**

Antes que as imagens possam ter sua distorção corrigida, um processo de calibração deve ser efetuado. Nesse trabalho foi utilizada uma biblioteca de calibração de câmera criada por Bouguet [Bouguet 2004], que é baseada no procedimento de quatro-passos com correção de imagem implícita descrito por Heikkilä e Silven [Heikkilä e Silven 1997]. Esse procedimento pode ser utilizado em várias aplicações de visão computacional, mas seu maior benefício é para o uso de medições baseadas em câmeras; e também em visão robótica onde há necessidade de alta precisão geométrica. Esse procedimento utiliza métodos de calibração explícitos para mapeamento de coordenadas 3D na imagem e uma abordagem implícita para correção de distorção da mesma [Heikkilä e Silven 2004].

Para a calibração da câmera foram utilizadas 75 imagens capturadas de um padrão xadrez com dimensões conhecidas (nesse caso o tamanho do padrão foi de 284.2x203.0 mm com 14x10 quadrados de 20.3 mm de lado) em ângulos diferentes. A Figura 3.12 (a) mostra uma das imagens utilizadas para calibração. O procedimento de calibração inicia-se com a seleção dos quatro cantos do padrão (formando o retângulo em verde da Figura 3.12 (b)), e depois, escolhe-se um valor inicial para o fator de distorção radial. As cruces em vermelho mostradas na Figura 3.12 (b) indicam a estimativa de onde os cantos de cada quadrado estão posicionados, considerando a distorção radial.



**Figura 3.12:** (a) Padrão utilizado na calibração da câmera, (b) as cruces em vermelho mostradas indicam a estimativa de onde os cantos de cada quadrado estão posicionados. (c) correção automática dos cantos dos quadrados, analisando as janelas de 5x5 pixels, mostradas em azul.

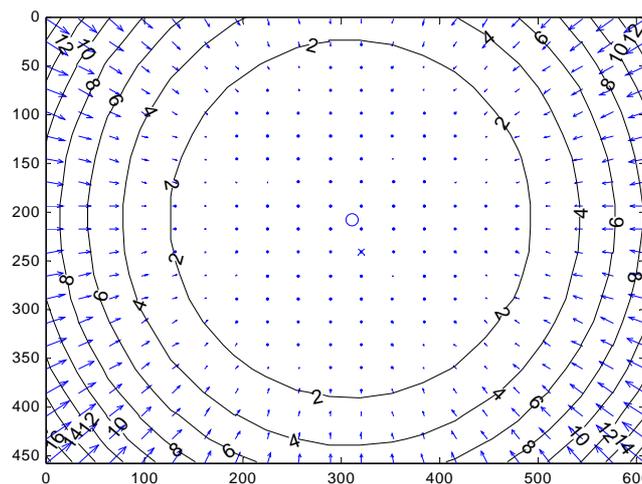
Após essa operação, os cantos de cada quadrado são corrigidos automaticamente, sendo analisados por uma janela de 5x5 *pixels* (quadrados em azul Figura 3.12 (c)) em

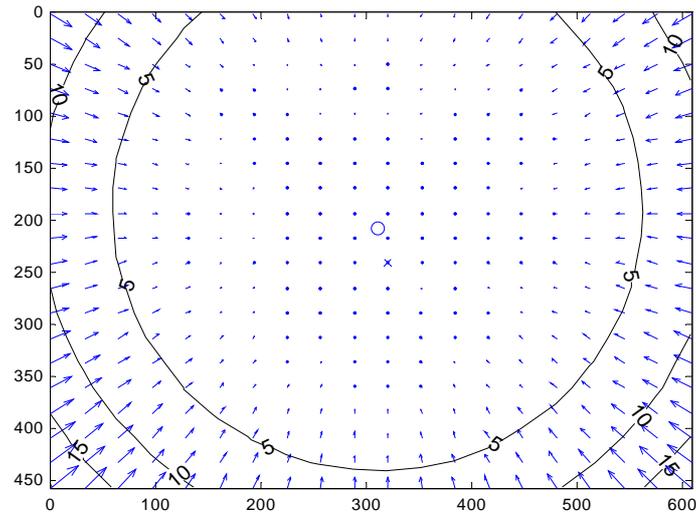
volta de cada canto. A nova posição dos cantos é mostrada em vermelho na Figura 3.12 (c) e a precisão em sua determinação é de 0.1 pixel.

A distorção radial das lentes faz com que um determinado pixel da imagem seja deslocado radialmente no plano da imagem [Slama 1980]. A Figura 3.13 mostra a componente radial do modelo de distorção. Normalmente um ou dois coeficientes são suficientes para compensar a distorção.

O centro de curvatura da superfície das lentes nem sempre é estritamente colinear, introduzindo um outro tipo de distorção decentralizadora que tem duas componentes, radial e tangencial. A Figura 3.14 mostra a componente tangencial do modelo de distorção.

Um modelo apropriado para uma calibração precisa é derivado combinando-se o modelo de furo de agulha com a correção das distorções radial e tangencial. A Figura 3.15 mostra o modelo de distorção completo.





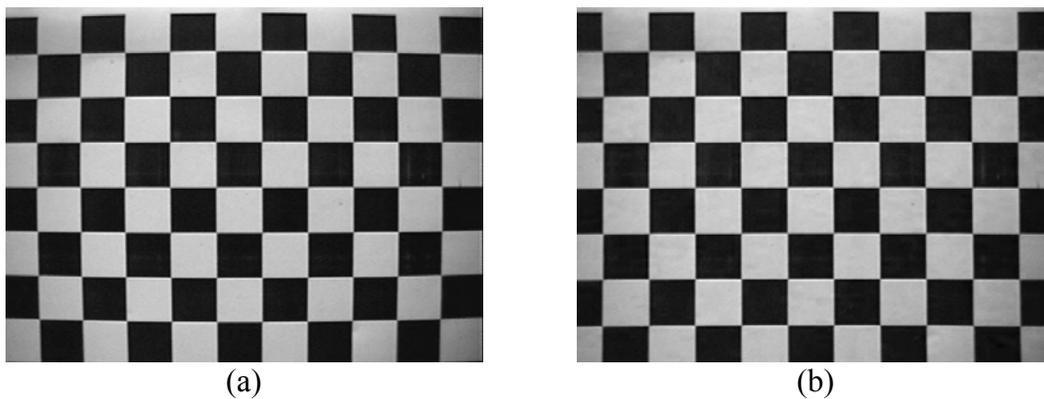
**Figura 3.15:** Modelo de distorção completo.

A Tabela 3.5 mostra os parâmetros obtidos pela calibração da câmera utilizada.

**Tabela 3.5:** Parâmetros de calibração da câmera.

Parâmetro	Valor	Erro
Comprimento focal	(772.403, 771,057)	+/- [0.7874 0.8587]
Ponto principal	(310.46, 206.498)	+/- [2.026, 1.645]
Coefficiente radial	(-0.1955, 0.07877, 0)	+/- [0.007243, 0.04346,0]
Coefficiente tangencial	(-0.004331, 0.0002312)	+/- [0.0004151, 0.0004249]

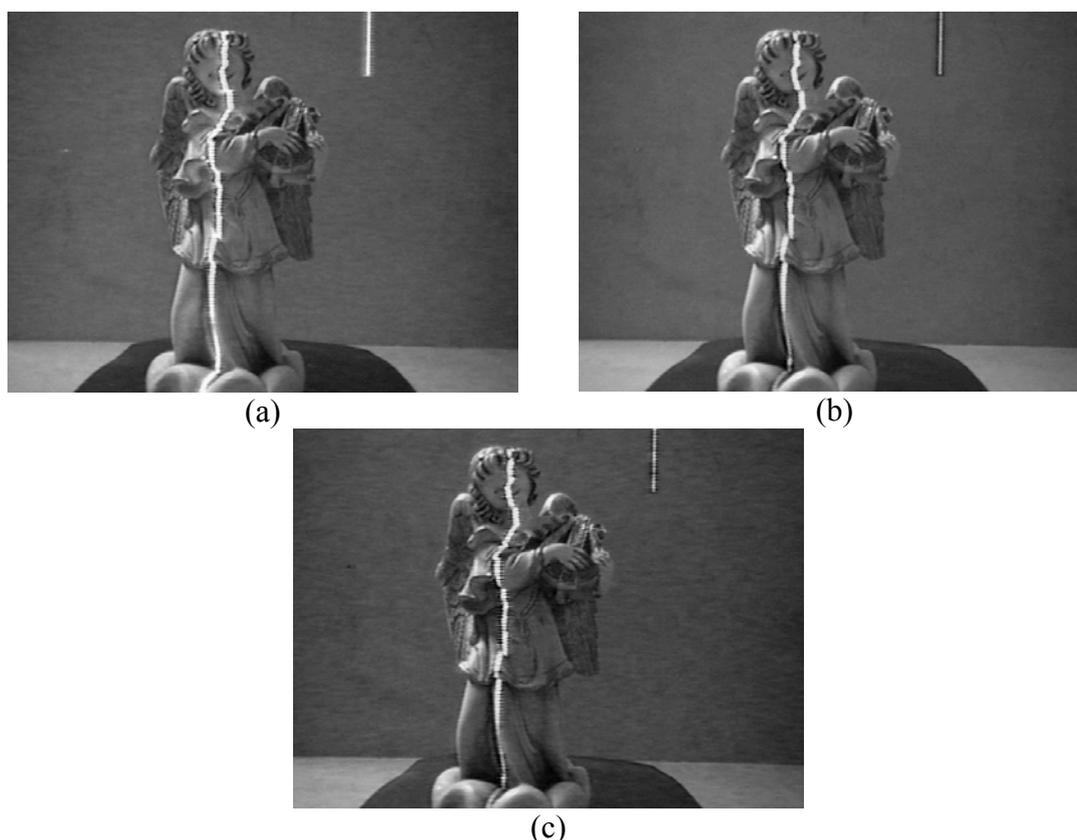
A Figura 3.16 mostra uma imagem antes de depois da correção de distorção.



**Figura 3.16:** (a) Imagem original, (b) imagem com correção de distorção.

## Segmentação

Para segmentar a projeção do laser da cena de interesse, optou-se pela utilização da imagem de intensidade porque a limiarização do canal vermelho não produziu resultados satisfatórios como na simulação e a projeção do laser está presente em todos os canais.

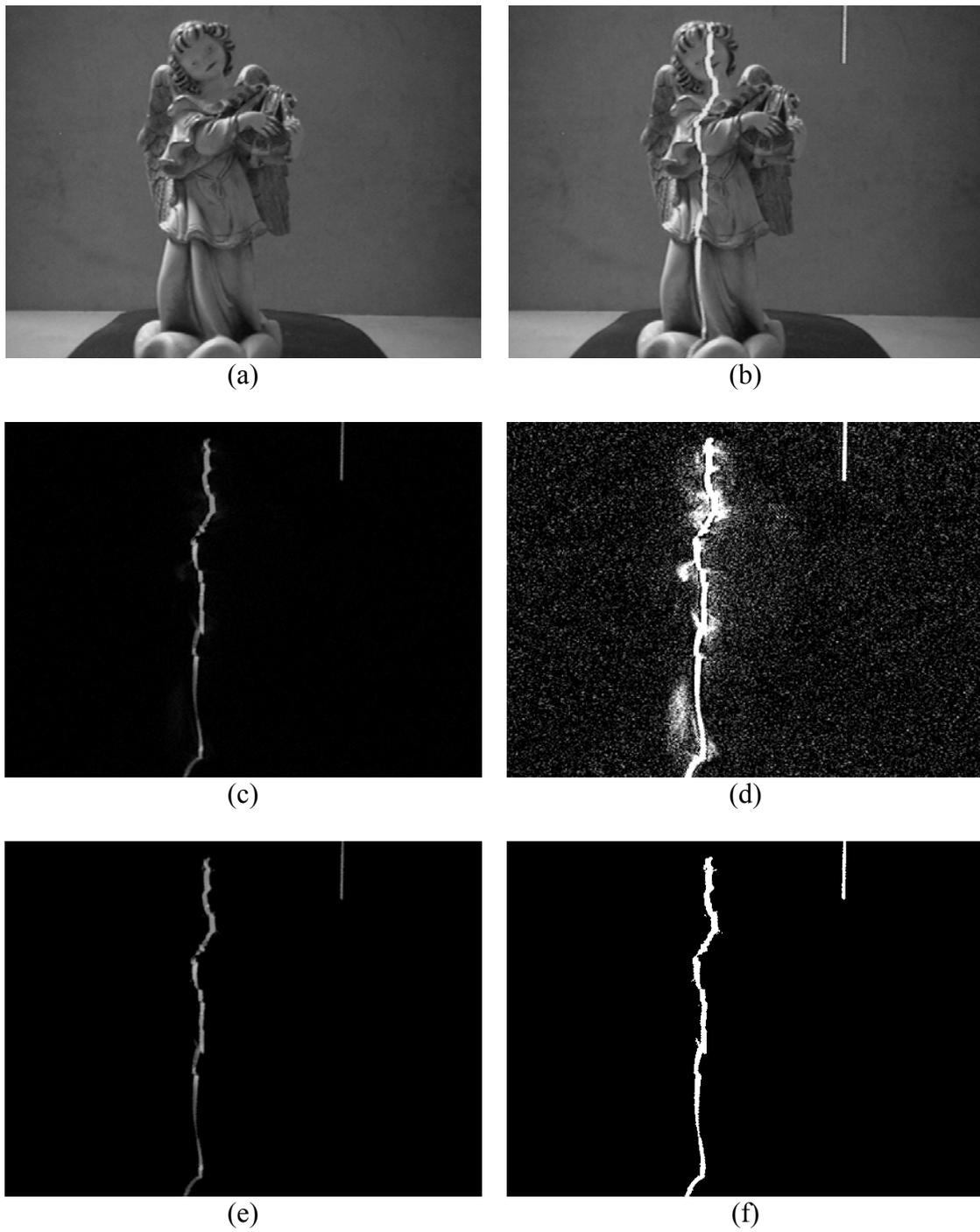


**Figura 3.17:** (a) Canal vermelho, (b) canal verde e (c) canal azul.

Devido ao fato da câmera utilizada não possuir três CCDs, os canais separados são muito ruidosos inviabilizando sua utilização individual. A Figura 3.17 (a), (b) e (c) mostra os canais vermelho, verde e azul, onde pode-se notar o ruído.

A segmentação de imagens inicia-se com a subtração de uma imagem da seqüência de aquisição (Figura 3.18 (a)) de uma imagem de referência sem o laser ligado (Figura 3.18 (b)). A Figura 3.18 (c) mostra a imagem resultante e (d) mostra a imagem (c) multiplicada por 20 para destacar o ruído.

Para eliminação de ruído é feito um pré-processamento onde ocorre a remoção de pixels que não são 8-conectados (Figura 3.18 (e) e (f)) e filtragem mediana (Figura 3.18 (g) e (h)) utilizando-se uma máscara de 5x5 para suavizar a imagem.



**Figura 3.18:** (a) Imagem de referência, (b) imagem da seqüência de aquisição, (c) subtração de (b) e (a), (d) imagem (c) multiplicada por 20, (e) eliminação de pixels que não são 8-conectados de (c), (f) imagem (e) multiplicada por 20, (g) filtragem mediana de (e), (h) imagem (g) multiplicada por 20.



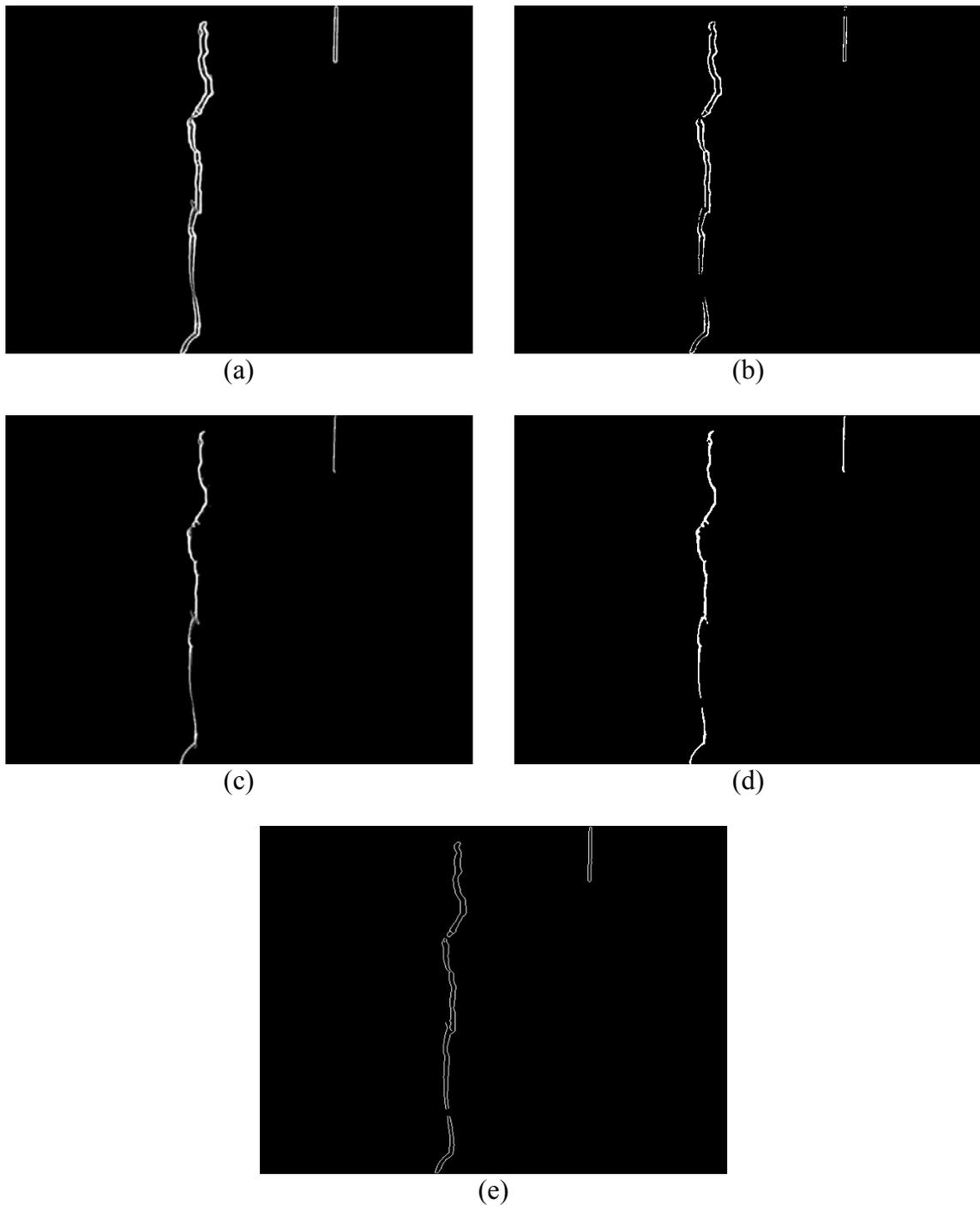
**Figura 3.18:** (Continuação)

Após a imagem ser suavizada, é feita a detecção de bordas e limiarização. Para essa aplicação foi utilizado o operador de Prewitt e Canny. A Figura 3.19 (a) mostra a detecção de bordas utilizando o operador de Prewitt, a Figura 3.19 (b) é a limiarização de (a) com  $T=200$ . Para evitar que, após a limiarização, a distância da borda de coluna da direita seja calculada nas regiões onde há lacuna na primeira borda, só foi considerada a parte positiva (borda da esquerda) referente à borda de coluna no cálculo da magnitude de Prewitt. Dessa forma, para calcular a magnitude, foi utilizada a equação (3.9):

$$|p_1| + p_2 \quad (3.9)$$

onde  $p_1$  é a máscara de linha e  $p_2$  é a máscara de coluna. A Figura 3.19 (c) mostra essa detecção de bordas e a Figura 3.19 (d) mostra a limiarização com  $T=110$ . Opcionalmente, pode-se utilizar a detecção de bordas Canny (Figura 3.19 (e)), que produz resultados melhores, mas com um alto custo computacional.

O algoritmo de segmentação de imagens é apresentado a seguir e ilustrado na Figura 3.20.



**Figura 3.19:** Detecção de bordas. (a) magnitude de Prewitt, (b) limiarização de (a)  $T=200$ , (c) magnitude de Prewitt sem a segunda borda de coluna, (d) limiarização de (c)  $T=110$  e (e) Canny.

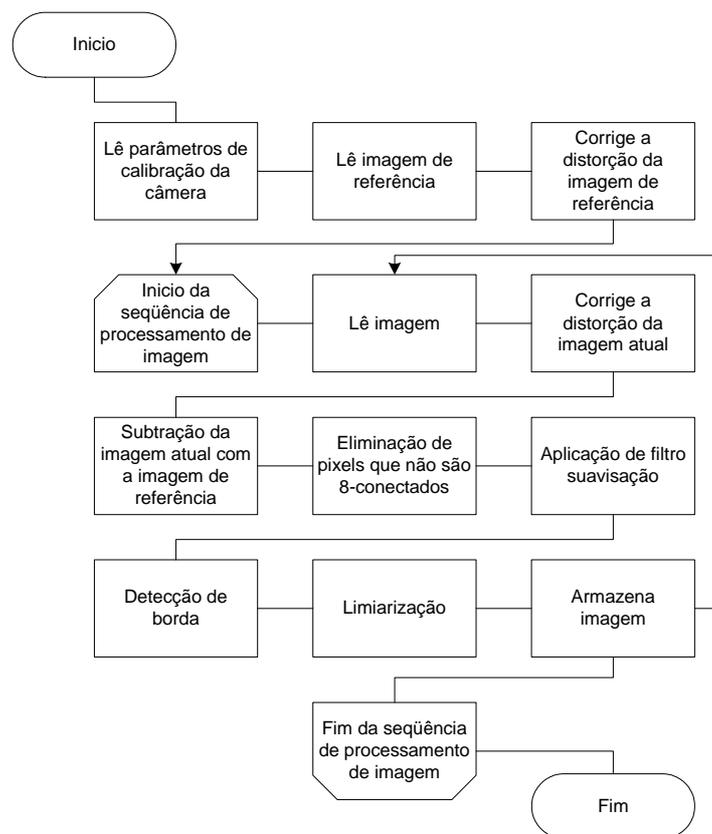
---

**Algoritmo 2** Segmentação de imagens

---

- 1 INICIO
- 2 Lê parâmetros de calibração da câmera
- 3 Lê imagem de referência
- 4 Corrige distorção da imagem de referência

- 5 Para cada imagem da seqüência de aquisição
  - 6 Lê imagem
  - 7 Corrige distorção da imagem atual
  - 8 Subtração da imagem atual com a imagem de referência
  - 9 Eliminação de pixels que não são 8-conectados
  - 10 Aplicação de filtro de suavização
  - 11 Detecção de bordas
  - 12 Limiarização
  - 13 Armazena imagem
  - 14 Fim (para)
  - 15 FIM
- 



**Figura 3.20:** Esquemático do algoritmo de segmentação de imagens.

### Estimação de Distância

Para a estimação de distância, utiliza-se as equações de triangulação para sensoriamento ativo (2.5), (2.6) e (2.7). O comprimento focal utilizado é proveniente da calibração, e está em unidade de pixel, podendo ser utilizado diretamente sem conversão para unidade métrica porque somente o relacionamento entre as coordenadas 3D de referência e as coordenadas 2D da imagem são necessários. Dessa forma  $u$  e  $v$  das equações (2.5), (2.6) e (2.7) também são utilizados em pixel, mas devem ser convertidos de forma que a origem esteja no ponto principal e não na

origem da imagem, que é o pixel que está no canto superior esquerdo. Assim  $u$  e  $v$  são descritos pela equação abaixo:

$$u=x' - x_0' \quad (3.10)$$

$$v=y' - y_0' \quad (3.11)$$

onde  $x'$  e  $y'$  são as coordenadas dos pixels de interesse na imagem e  $x_0'$  e  $y_0'$  são as coordenadas do ponto principal, provenientes da calibração (310.46, 206.498).

O algoritmo 3 tem como entrada uma imagem binária proveniente da limiarização de uma imagem de detecção de bordas. Cada linha da imagem é percorrida da esquerda para direita e quando o primeiro pixel com valor 1 é encontrado, estima-se sua distância. Se a distância for menor que um limiar de distância informado pelo usuário o ponto  $(x,y,z)$  fará parte da nuvem de pontos. Analisa-se a próxima linha.

---

**Algoritmo 3:** Estimação de Distância

---

```

1  INICIO
2  Para r =1 até a ultima linha da imagem
3      c=1
4      Enquanto o valor do pixel I(r,c) < 1
5          c = c + 1
6      Fim enquanto
7      Se o valor do pixel > 0
8          Estima a distância
9          Se z < limiar de distância
10             Inclui o ponto(x,y,z) na nuvem de pontos
11         Fim (se)
12     Fim (se)
13 Fim (para)
14 FIM

```

---

O tempo total do módulo de estimação de distância utilizando detecção de bordas Prewitt é de 708 ms, enquanto que, utilizando Canny, é 2441 ms. A Tabela 3.6 mostra o tempo de cada operação por imagem. Os tempos foram obtidos pela média de 200 imagens, utilizando-se um microcomputador pessoal com processador Athlon XP 2600+, 512 Mb de memória RAM de 333 Mhz.

**Tabela 3.6:** Tempo das operações do módulo de estimação de distância.

Operação	Tempo (ms)
Correção de distorção	281
Subtração	16
8-conectados	47
Filtragem mediana	32
Detecção de bordas Prewitt	297
Detecção de bordas Canny	2030
Estimação de distância	35

---

### 3.4.4 Software de Controle

O software de controle é responsável pela inicialização do sistema de varredura e aquisição 3D. Na inicialização, os parâmetros do motor são informados ao controlador e o sistema de varredura é posicionado de forma que o laser fique paralelo à câmera.

Para que a movimentação do motor ocorra adequadamente, os termos proporcional e integral devem ser calibrados previamente assim como uma calibração entre a câmera e o laser para garantir que se saiba o posicionamento do motor sempre que o sistema da varredura for ligado.

Na aquisição o software de controle gerencia a captura de imagens e o controle de varredura.

#### Inicialização do sistema de varredura

O processo de inicialização é formado por uma etapa de configuração, um sub-sistema de proteção e uma etapa de posicionamento. Na etapa de configuração os parâmetros operacionais do sistema de varredura são informados ao controlador de movimentos. No sub-sistema de proteção, limites de varredura são estabelecidos para que o sistema não choque com barreiras físicas e a corrente é limitada para, caso isso ocorra, o sistema não danifique ou destrua a redução mecânica ou até mesmo o laser. A etapa de posicionamento é realizada para que o sistema de varredura seja posicionado sempre no mesmo lugar, quando for ligado. Quando o sistema de varredura é ligado, a posição do motor lida pelo *encoder* será sempre zero, independentemente de onde o motor estiver. Para evitar esse problema, uma seqüência de eventos é realizada para que o motor seja posicionado de maneira que o laser esteja, precisamente, em paralelo à câmera, sempre que o sistema for ligado.

O algoritmo 4 é responsável pela inicialização do sistema de varredura e sua ilustração é representada na Figura 3.21.

---

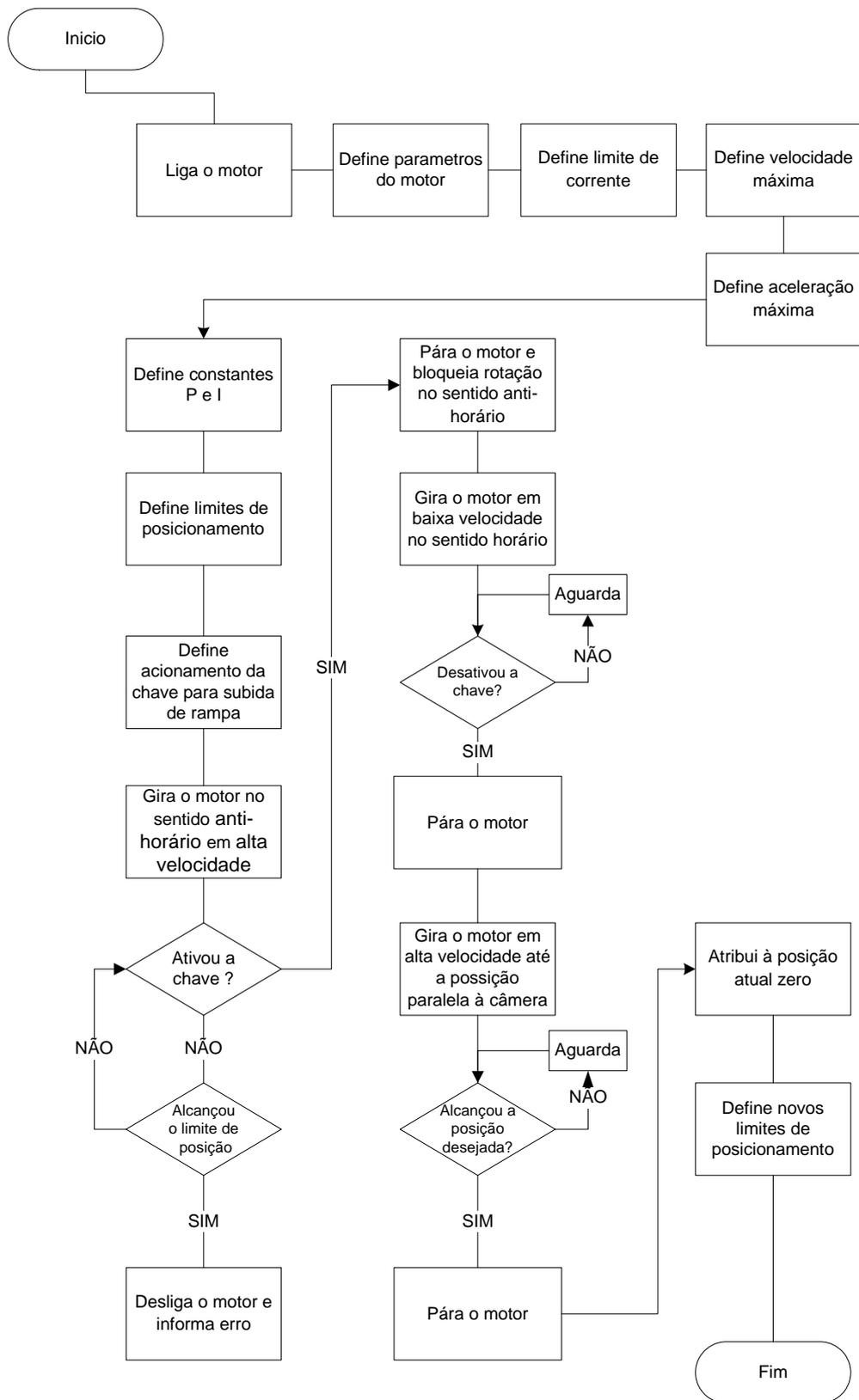
#### **Algoritmo 4:** Inicialização do Sistema de Varredura

---

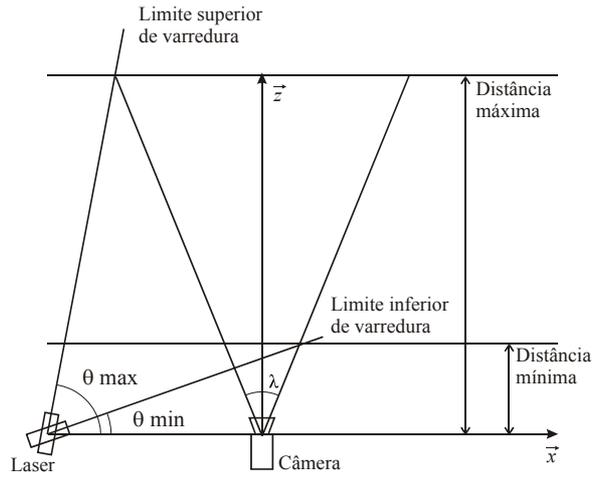
- 1 INÍCIO
- 2 Liga o motor
- 3 Os parâmetros do motor, constante de velocidade e resistência interna, e resolução do encoder, são informados ao controlador do motor.
- 4 É estabelecido um limite de corrente de pico e um limite de corrente contínua para o motor.

- 5 Estabelecimento de velocidade máxima
  - 6 Estabelecimento de aceleração máxima
  - 7 Definição de constante proporcional e integral. Para determinação dessas constantes um outro algoritmo é utilizado previamente e será apresentado posteriormente.
  - 8 Definição de limites de posicionamento. Aqui os limites são estabelecidos para que o sistema de varredura possa girar 360° em ambas as direções.
  - 9 Gira o motor no sentido anti-horário em alta velocidade. Inicialmente procura-se pela chave em alta velocidade para que pouco tempo seja gasto, mas isso não garante que a chave seja acionada sempre no mesmo instante.
  - 10 Enquanto a chave não for ativada
  - 11     Se o motor alcançou o limite de posição (a chave está danificada)
  - 12         Desliga o motor
  - 13         Envia notificação de erro
  - 14     Fim (se)
  - 15 Fim (enquanto)
  - 16 Pára o motor
  - 17 Bloqueia o sentido de rotação anti-horário enquanto ela estiver ativada
  - 18 Gira o motor em baixa velocidade no sentido horário
  - 19 Enquanto a chave não for desativada
  - 20     Aguarda
  - 21 Fim (enquanto)
  - 22 Pára o motor
  - 23 Gira o motor em alta velocidade até a posição onde o laser está paralelo à câmera. Essa posição é determinada através de um algoritmo de calibração entre a câmera e o laser e será apresentado posteriormente
  - 24 Enquanto a posição desejada não for alcançada
  - 25     Aguarda
  - 26 Fim (enquanto)
  - 27 Pára o motor
  - 28 Atribui à posição atual o valor zero
  - 29 Define novos limites de posicionamento. Esses limites são determinados de forma que o limite inferior seja o limite de alcance do campo de visão da câmera para objetos distantes e o limite superior seja o limite de alcance do campo de visão da câmera para objetos próximos. A Figura 3.22 ilustra esses limites, onde  $\theta_{min}$  é o limite inferior,  $\theta_{max}$  é o limite superior e  $\lambda$  é o campo de visão da câmera.
  - 30 FIM
- 

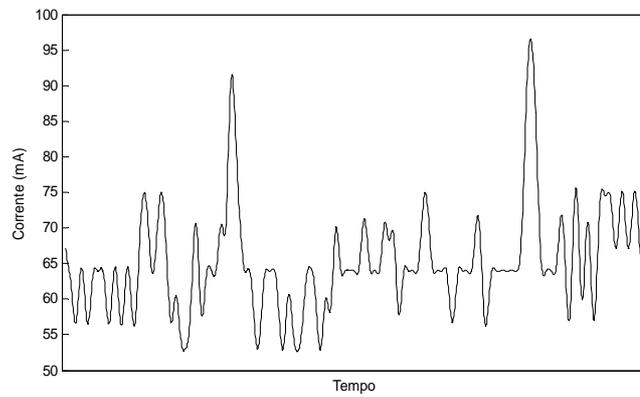
Para essa aplicação foi utilizada a velocidade máxima de 4000 rpm e aceleração máxima de 500 revoluções/s<sup>2</sup>. Os limites de corrente foram estabelecidos em 110 mA para pico e 80 mA para operação contínua. A Figura 3.23, mostra a plotagem da corrente e da velocidade em função do tempo, para o motor passando do estado parado para o estado de velocidade máxima (4000 rpm), com aceleração máxima (500 revoluções/s<sup>2</sup>) e depois voltando para o estado parado com velocidade e aceleração máxima.



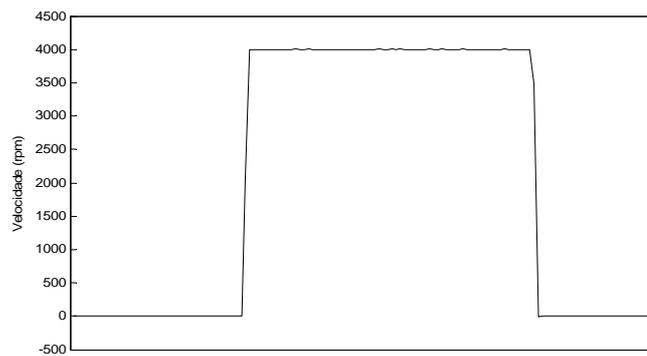
**Figura 3.21:** Esquemático do algoritmo de inicialização do sistema de varredura.



**Figura 3.22:** Limites de varredura.



(A)



(B)

**Figura 3.23:** (a) plotagem da corrente em função do tempo. O primeiro pico indica a mudança de velocidade de 0 para 4000 rpm e o segundo pico indica a mudança de velocidade de 4000 para 0 rpm. (b) plotagem de velocidade em função do tempo com aceleração de 30000 revoluções/s<sup>2</sup>.

O algoritmo 5 é responsável pela calibração do termo proporcional e integral e sua representação é ilustrada na Figura 3.24.

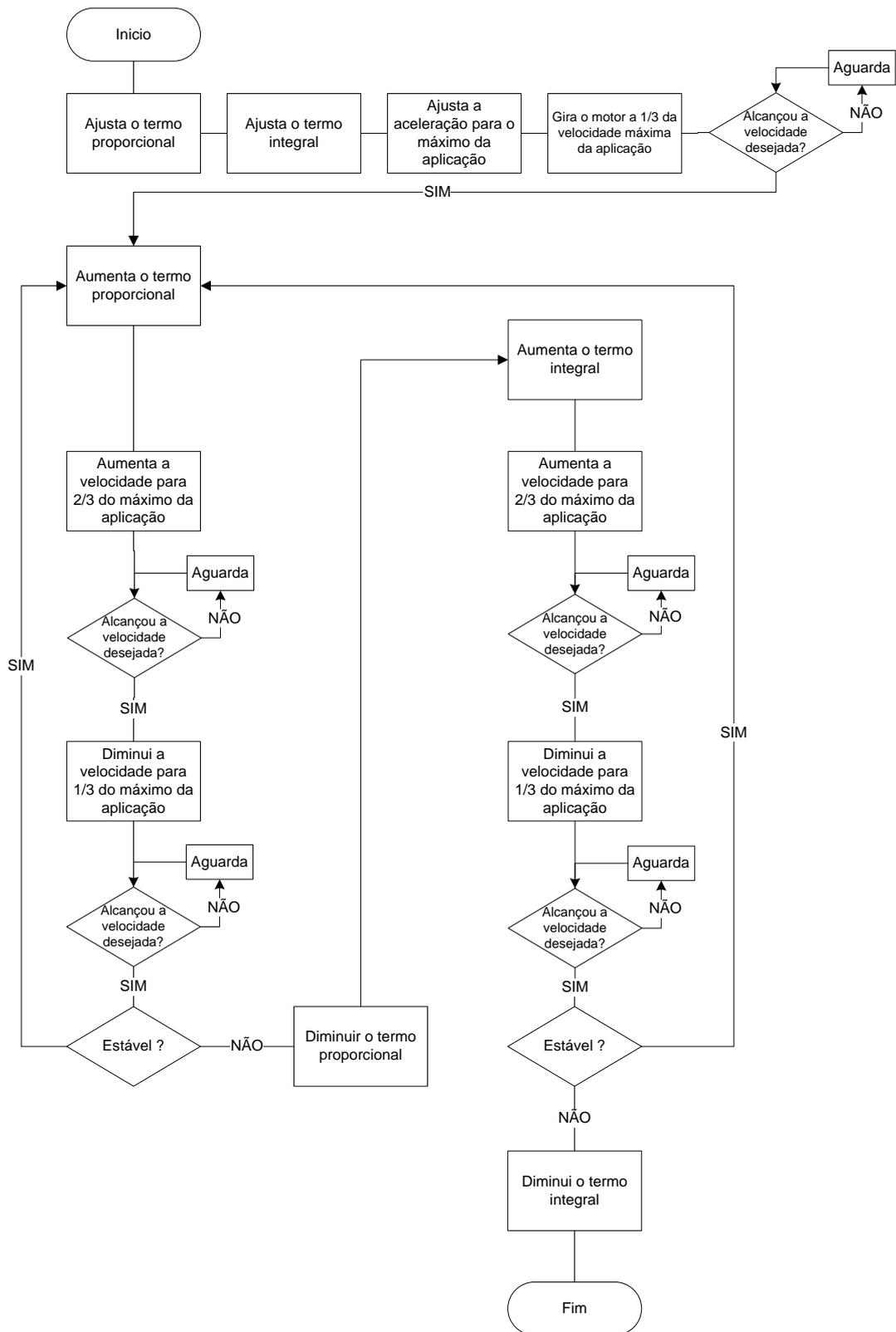
---

**Algoritmo 5:** Calibração do Termo Proporcional e Integral

---

- 1 INÍCIO
- 2 Ajusta o termo proporcional com um valor baixo
- 3 Ajusta o termo integral para zero
- 4 Ajusta a aceleração para o valor máximo da aplicação
- 5 Gira o motor com 1/3 da velocidade máxima da aplicação
- 6 Enquanto a velocidade desejada não for alcançada
- 7     Aguarda
- 8 Fim (enquanto)
- 9 Aumenta o termo proporcional
- 10 Aumenta a velocidade do motor para 2/3 da velocidade máxima da aplicação
- 11 Enquanto a velocidade desejada não for alcançada
- 12     Aguarda
- 13 Fim (enquanto)
- 14 Diminui a velocidade para 1/3 do máximo da aplicação
- 15 Enquanto a velocidade desejada não for alcançada
- 16     Aguarda
- 17 Fim (enquanto)
- 18 Se estiver estável sem oscilação
- 19     Volta para (8)
- 20 Senão
- 21     Diminui o termo proporcional
- 22     Aumenta o termo integral
- 23     Aumenta a velocidade para 2/3 do máximo da aplicação
- 24     Enquanto a velocidade desejada não for alcançada
- 25         Aguarda
- 26     Fim (enquanto)
- 27     Diminui a velocidade para 1/3 do máximo da aplicação
- 28     Enquanto a velocidade desejada não for alcançada
- 29         Aguarda
- 30     Fim (enquanto)
- 31     Se estiver estável sem oscilação
- 32         Volta para (8)
- 33     Senão
- 34         Diminui o termo integral
- 35     Fim (se)
- 36 Fim (se)
- 37 FIM

---



**Figura 3.24:** Esquemático do algoritmo de calibração do termo proporcional e integral.

## Calibração Laser-Câmera

A calibração da posição do laser em relação à câmera é efetuada para que se determine precisamente a posição em que o laser esteja paralelo à câmera. Para isso procura-se a posição em que o laser aponta para o centro da imagem. Inicialmente o laser passa rapidamente pelo centro da imagem e pára, embora o laser não esteja precisamente no centro da imagem. A partir daí, inicia-se um processo de refinamento em que, tanto o desvio em relação ao centro da imagem, quanto o ângulo de rotação do motor vão sendo reduzidos. Para que o laser fique paralelo à câmera é necessário conhecer a priori a distância entre o sensor e um objeto de referência. O algoritmo 6 é responsável pela calibração entre o laser e a câmera e sua representação é mostrada na Figura 3.25.

A calibração entre o laser e a câmera poderia ser feita analisando-se a posição do laser em qualquer posição em que ele estivesse sendo refletido no objeto de referência. Mas, se algum outro parâmetro, como, comprimento focal, tamanho do sensor ótico (CCD ou CMOS) ou distorções óticas, estivesse incorreto, haveria um cascadeamento de erros. A calibração implementada aqui, funciona para qualquer comprimento focal e tamanho de sensor ótico.

---

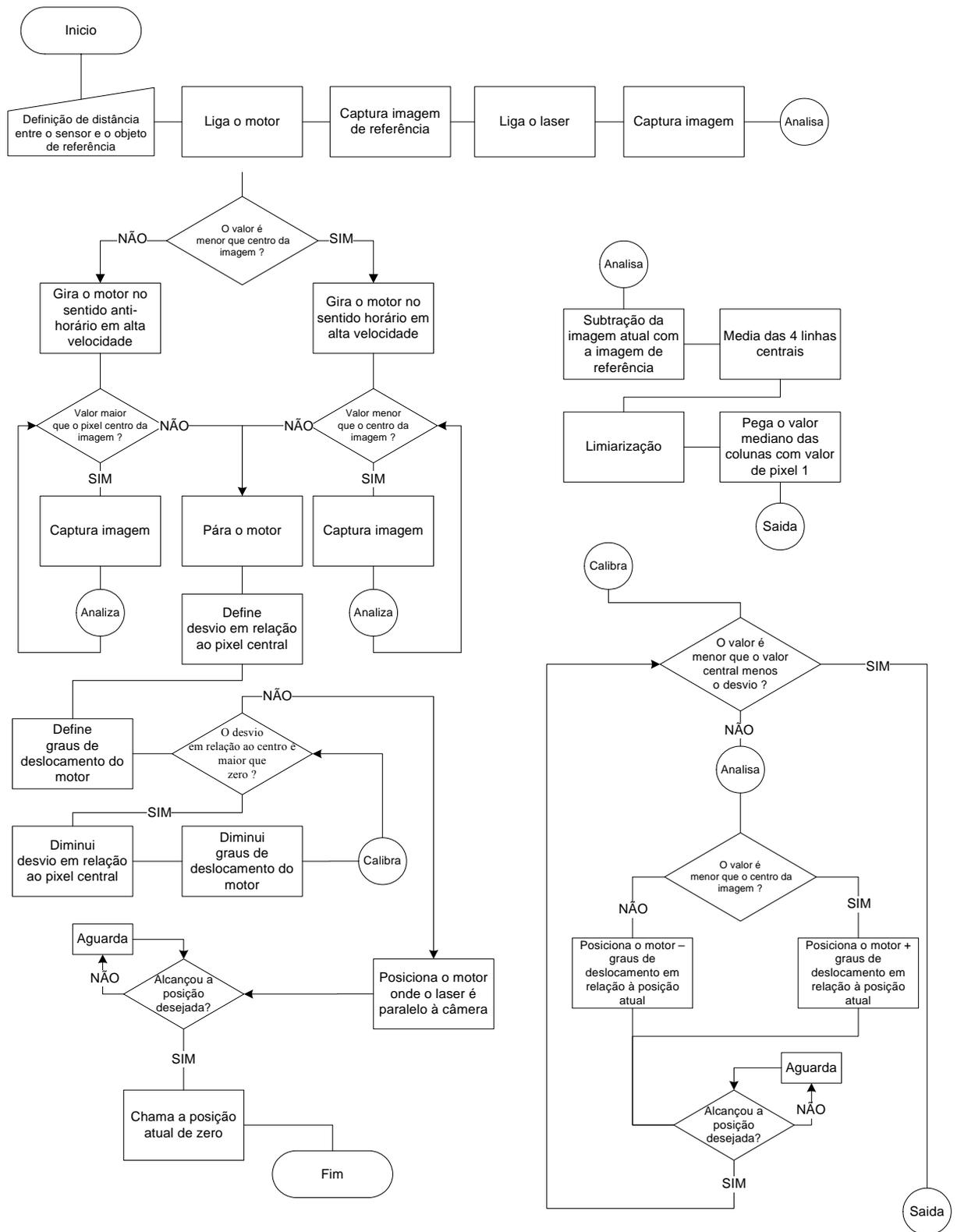
### Algoritmo 6: Calibração Laser Câmera

---

```
1 INÍCIO
2 Definição de distância entre o sensor e o objeto de referência
3 Liga o motor
4 Captura imagem de referência
5 Liga o laser
6 Captura imagem
7 Analisa a imagem. Subtração da imagem de atual com a imagem de
  referência; determinação da mediana entre as 4 linhas centrais; limiarização;
  determinação da mediana das colunas com valor de pixel 1
8 Se o valor mediano for menor que centro da imagem
9   Gira o motor no sentido horário em alta velocidade
10  Enquanto o valor analisado for maior que o centro da imagem
11    Captura imagem
12    Analisa imagem
13  Fim (enquanto)
14 Senão
15  Gira o motor no sentido anti-horário em alta velocidade
16  Enquanto o valor analisado for menor que o centro da imagem
17    Captura imagem
18    Analisa imagem
19  Fim (enquanto)
20 Fim (se)
```

21 Pára o motor  
22 Define o desvio em relação ao pixel central  
23 Define graus de deslocamento do motor  
24 Enquanto desvio em relação ao pixel central for maior que zero; início do ciclo de refinamento de calibração  
25     Diminui o desvio em relação ao pixel central  
26     Diminui graus de deslocamento do motor  
27     Enquanto o valor não for menor que o valor central menos o desvio  
28         Captura imagem  
29         Analisa a imagem  
30         Se o valor for menor que o centro  
31             Posiciona o motor na posição atual + os graus de deslocamento  
32             Enquanto a posição desejada não for alcançada  
33             Aguarda  
34             Fim (enquanto)  
35         Senão  
36             Posiciona o motor na posição atual - os graus de deslocamento  
37             Enquanto a posição desejada não for alcançada  
38             Aguarda  
39             Fim (enquanto)  
40         Fim (se)  
41     Fim (enquanto)  
42 Fim (enquanto)  
43 Posiciona o motor onde ele é paralelo à câmara  
44 Enquanto a posição desejada não for alcançada  
45     Aguarda  
46 Fim (enquanto)  
47 Chama a posição atual de zero  
48 FIM

---



**Figura 3.25:** Esquemático do algoritmo de calibração entre a câmera e o laser.

## Aquisição

Três sistemas de aquisição foram implementados. O primeiro não possui redutor mecânico e utiliza um motor, que não tem atrito uniforme ao longo de sua revolução. Por esse motivo não foi possível fazer uma calibração precisa dos termos proporcional e integral. O segundo sistema de aquisição possui um redutor mecânico de 3,9:1 e utiliza o mesmo motor do primeiro. O terceiro sistema possui um motor integrado a um redutor mecânico de 879:1 com folga zero entre o começo e o final da redução.

Para o primeiro e segundo sistemas foi utilizado um algoritmo de aquisição de posicionamento que garante que, a cada imagem capturada, a varredura está igualmente espaçada, isso é, se em um determinado ciclo de aquisição o motor está na posição  $p$ , no ciclo seguinte a posição é  $p$ +deslocamento do motor especificado pelo usuário. Para o primeiro sistema, sem redução, tem-se que, aplicando-se a equação (3.12), um pulso de *encoder* representa aproximadamente  $0.1758^\circ$  e para o segundo  $0.0451^\circ$ .

$$R = \frac{360}{Enc.Res.*Redução}, \quad (3.12)$$

onde  $R$  é a resolução de varredura em graus e *Enc.Res.* é a resolução do *encoder* (2048 para os três sistemas implementados). Em ambos os casos, é importante que as aquisições sejam igualmente espaçadas.

Para o terceiro sistema foram implementados dois algoritmos de aquisição. O primeiro algoritmo é semelhante ao algoritmo do primeiro sistema, mas sem a necessidade das aquisições serem igualmente espaçadas, pois se o motor não parasse na posição desejada estaria a, no máximo, 10 pulsos de *encoder* fora da posição desejada, sendo que nesse sistema, um pulso representa aproximadamente  $0.0002^\circ$ . O segundo modo de aquisição parte da posição inicial e segue com velocidade constante até a posição final. Ao longo do caminho as posições são capturadas e armazenadas.

Os três algoritmos possuem uma fase inicial que é igual para os três modos de aquisição, onde os limites de aquisição são definidos e o tipo de aquisição é determinada. Os algoritmos 7,8,9 e 10 são responsáveis pela aquisição e suas respectivas ilustrações são apresentadas nas Figura 3.26, Figura 3.27, Figura 3.28 e Figura 3.29.

---

**Algoritmo 7: Aquisição**

---

- 1 INÍCIO
- 2 Define limites de varredura
- 3 Define modo de aquisição; três modos de aquisição podem ser escolhidos, sendo que os dois primeiros são de posicionamento e o terceiro modo é de velocidade
- 4 Se o modo escolhido for posicionamento
- 5     Define graus de deslocamento do motor
- 6 Senão se o modo escolhido for velocidade
- 7     Define velocidade
- 8 Fim (se)
- 9 Captura imagem de referência
- 10 Liga o motor
- 11 Posiciona o motor na posição inicial de varredura
- 12 Enquanto a posição desejada não for alcançada
- 13     Aguarda
- 14 Fim Enquanto
- 15 Armazena a posição atual
- 16 Liga o laser
- 17 Se o modo for “1”
- 18     Inicia o modo de aquisição “posicionamento igualmente espaçado”
- 19 Senão se o modo de aquisição for “2”
- 20     Inicia o modo de aquisição “posicionamento”
- 21 Senão se o modo de aquisição for “3”
- 22     Inicia o modo de aquisição “velocidade”
- 23 Fim Se
- 24 FIM

---

---

**Algoritmo 8: Aquisição, modo de posicionamento igualmente espaçado**

---

- 1 INÍCIO
- 2 Para 1 até (limite final – limite inicial) / graus de deslocamento do motor
- 3     Posiciona o motor na posição desejada
- 4     Enquanto a posição desejada não for alcançada. Observe que se o motor alcançou a posição desejada não significa que ele esteja na posição desejada só indica que o motor passou por lá.
- 5         Aguarda
- 6     Fim (enquanto)
- 7     Enquanto não estiver na posição desejada. Ao passar pela posição desejada o motor leva um determinado tempo até se estabilizar.
- 8         Aguarda
- 9     Fim (enquanto)
- 10    Para o motor
- 11    Se o motor estiver na posição desejada. Após ter parado o motor pode estar antes ou depois da posição desejada, mas não muda mais de posição, por isso mais uma vez a posição é perguntada.
- 12    Armazena a posição atual

- 13 Captura imagem
  - 14 Incrementa a posição desejada. Os incrementos são os graus de deslocamento do motor, definidos anteriormente pelo usuário.
  - 15 Senão
  - 16 Volta para (2)
  - 17 Fim (se)
  - 18 Fim (para)
  - 19 Desliga o motor
  - 20 FIM
- 

---

**Algoritmo 9:** Aquisição, modo de posicionamento

---

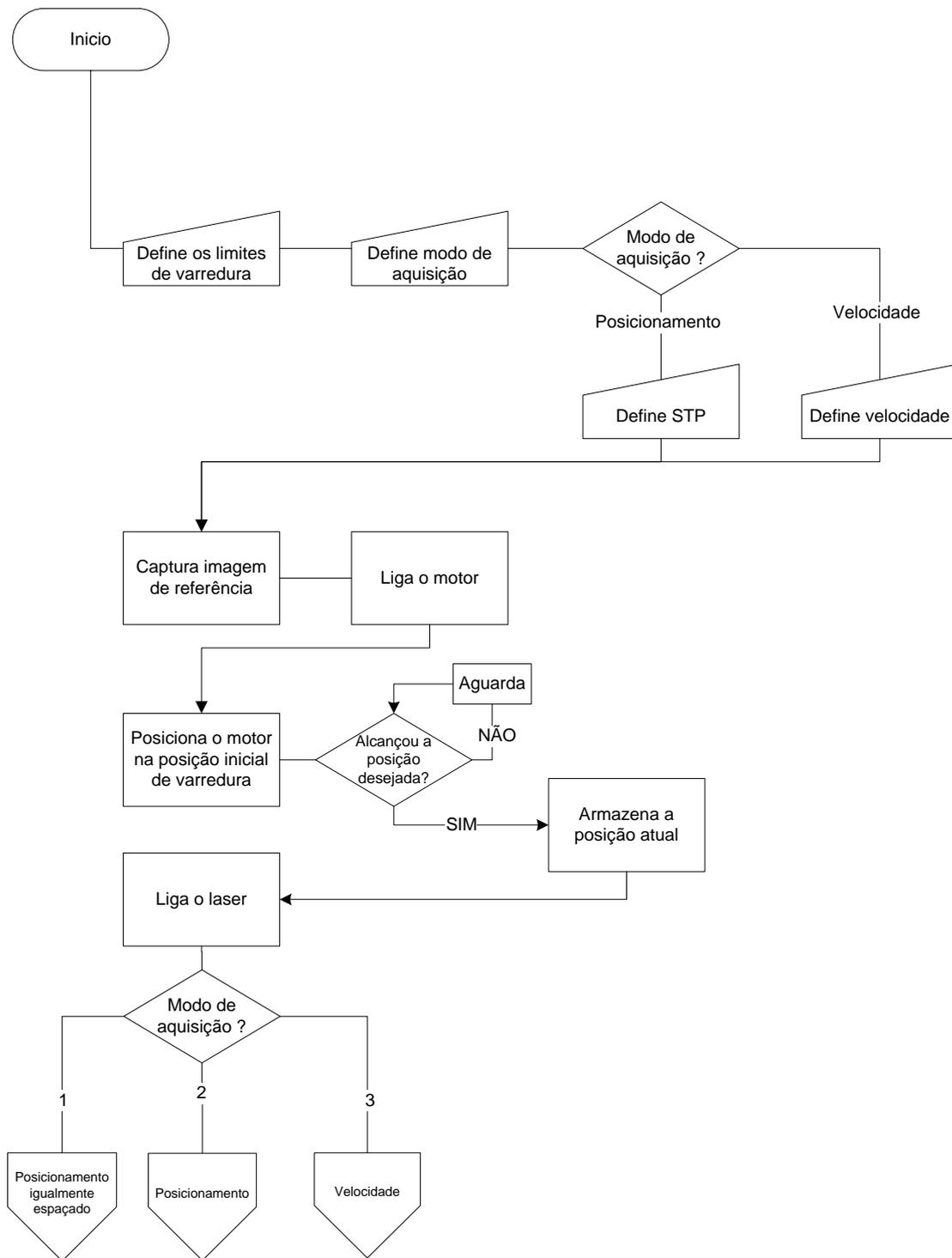
- 1 INÍCIO
  - 2 Para 1 até (limite final – limite inicial) / graus de deslocamento do motor
  - 3 Posiciona o motor na posição desejada.
  - 4 Enquanto não alcançar a posição desejada
  - 5 Aguarda
  - 6 Fim (enquanto)
  - 7 Para o motor
  - 8 Pergunta a posição atual
  - 9 Armazena a posição atual
  - 10 Captura imagem
  - 11 Incrementa a posição desejada. Os incrementos são os graus de deslocamento do motor, definidos pelo usuário.
  - 12 Fim (para)
  - 13 Desliga o motor
  - 14 FIM
- 

---

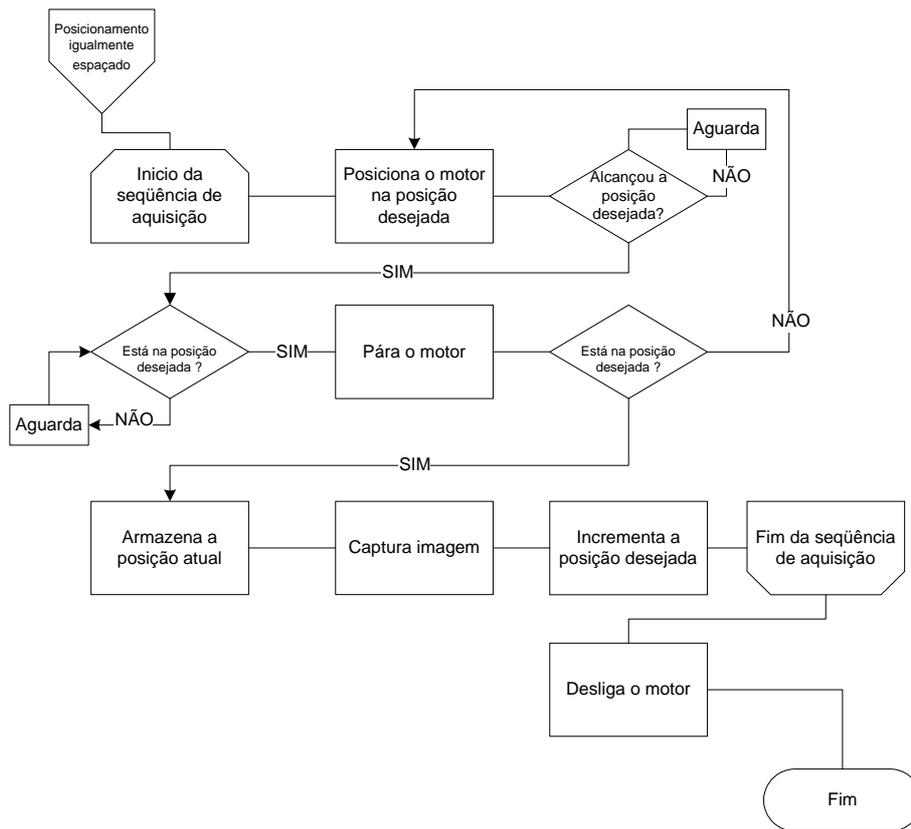
**Algoritmo 10:** Aquisição, modo de velocidade

---

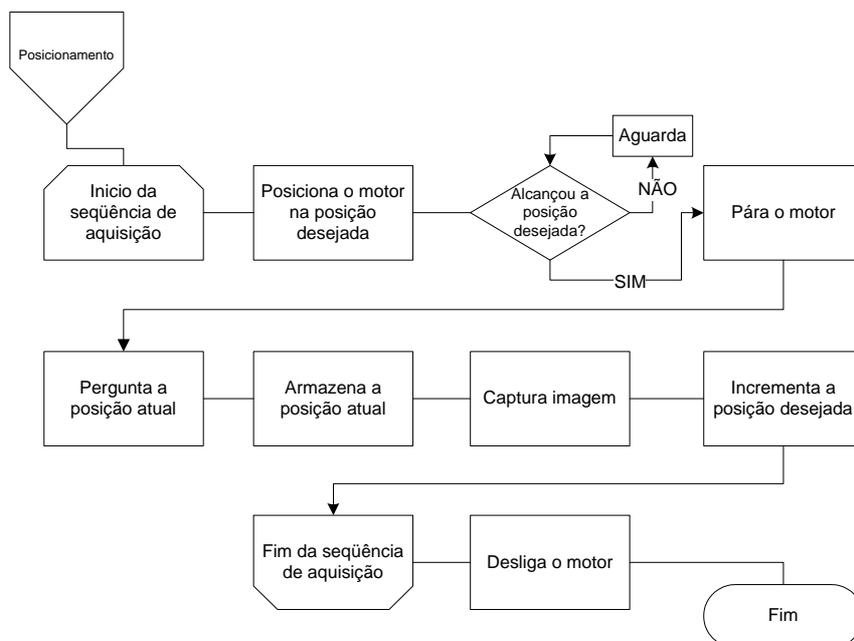
- 1 INÍCIO
  - 2 Enquanto a posição atual for menor que a posição do limite superior de varredura
  - 3 Gira o motor na velocidade predefinida
  - 4 Pergunta a posição atual
  - 5 Armazena a posição atual
  - 6 Captura imagem
  - 7 Fim (enquanto)
  - 8 FIM
-



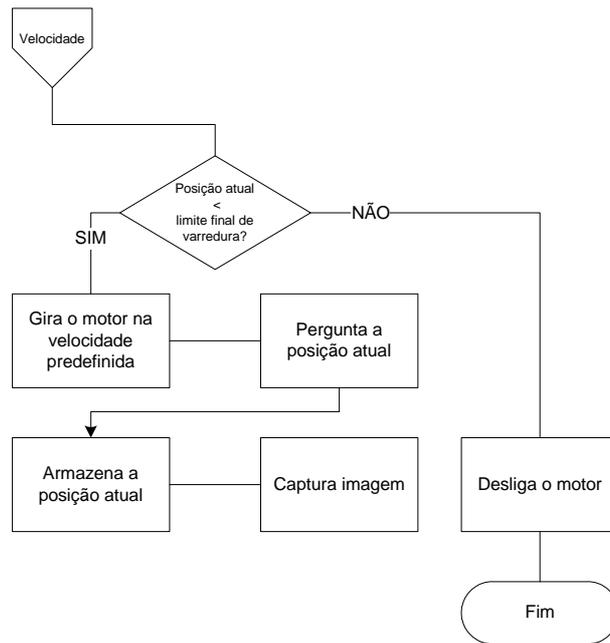
**Figura 3.26:** Esquemático da primeira parte do algoritmo de aquisição.



**Figura 3.27:** Esquemático do algoritmo de aquisição com modo de posicionamento igualmente espaçado.



**Figura 3.28:** Esquemático do algoritmo de aquisição, modo de posicionamento.



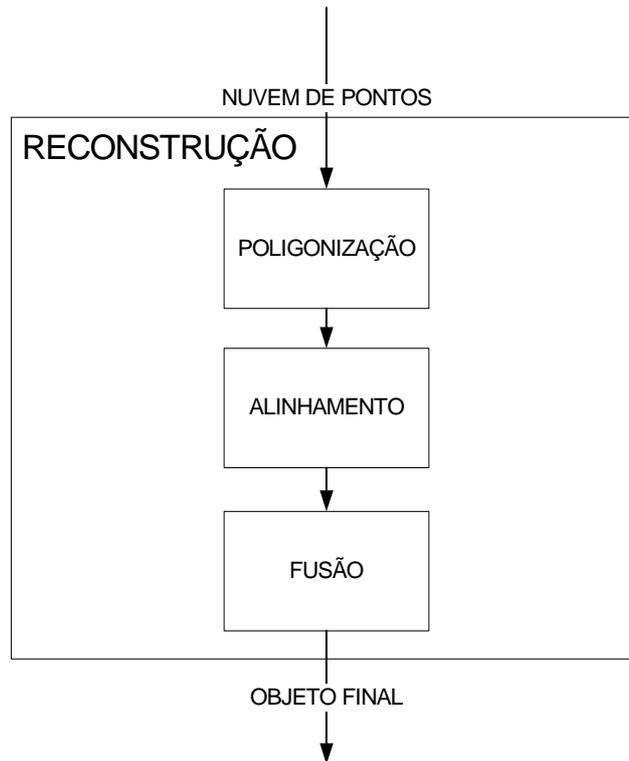
**Figura 3.29:** Esquemático do algoritmo de aquisição, modo de velocidade.

### 3.4.5 Reconstrução

O módulo de reconstrução, (Figura 3.30) tem a função de transformar uma ou mais nuvens de pontos, provenientes do módulo de estimação de distância, em um objeto único com malha de superfície triangular.

A poligonização gera a malha triangular. Um estudo e comparação de várias técnicas para geração de malha triangular foi feito por Gois [Gois 2004]. A biblioteca CGAL - *Computational Geometry Algorithms Library* [CGAL 2004] contém diversas triangulações implementadas de maneira eficiente, robusta e flexível a alterações e adaptações e está disponível *on-line* no *site* <http://www.cgal.org>.

Como o sistema de aquisição só faz coleta de dados onde o objeto está no campo de visão da câmera, várias aquisições podem ser necessárias para que um objeto original seja representado integralmente. As nuvens de pontos geradas por essas aquisições ou as malhas triangulares de superfície devem ser alinhadas e fundidas em um único objeto 3D. Uma técnica de alinhamento e fusão é apresentada no capítulo 2.

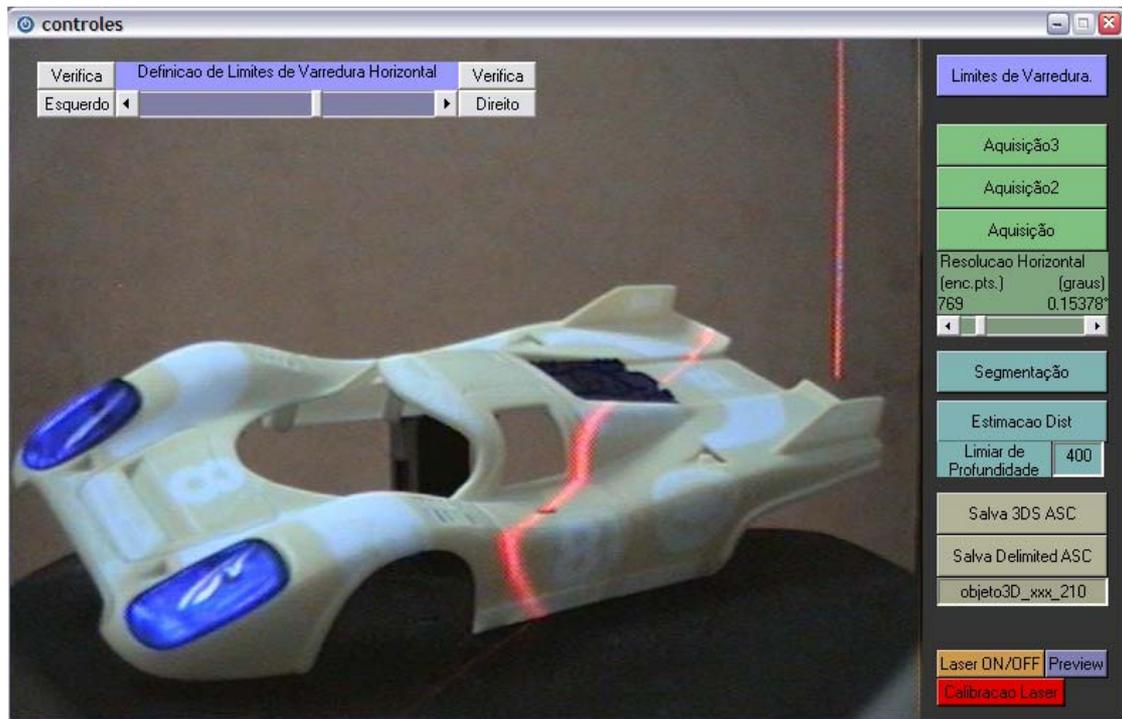


**Figura 3.30:** Módulo de reconstrução.

### 3.4.6 Interface Gráfica

A interface gráfica permite que o usuário: determine os limites de varredura utilizando a barra de rolagem e os verifique, fazendo com que o laser seja posicionado nesses limites; escolha o modo de aquisição e determine a resolução de varredura horizontal; faça a segmentação de imagens; determine o limiar de profundidade e faça a estimação de distância; salve o objeto 3D nos formatos 3D Studio ASCII ou Delimited ASCII (formato compatível com a maioria dos softwares de geração de malhas de superfície); ligue e desligue o laser manualmente para testar a intensidade do mesmo (durante a aquisição, o laser é ligado e desligado automaticamente e fica ligado pelo tempo mínimo necessário); ligue ou desligue o *preview* de vídeo (o *preview* é desligado automaticamente na aquisição, segmentação e estimação de profundidade para otimização de performance); e faça a calibração entre o laser e a câmera (isso só é feito uma vez depois que o sistema é implementado e está presente para testes). A Figura 3.31 mostra a interface gráfica.

A aquisição, a segmentação e a estimação de distância foram implementados separadamente para que, a aquisição pudesse ser executada mais rapidamente e para que, fazendo apenas uma aquisição de imagens, diferentes procedimentos de segmentação de imagens pudessem ser testados e monitorados. Esse tipo de implementação otimiza a performance ao custo de uma maior utilização de memória.



**Figura 3.31:** Interface gráfica do sistema de aquisição.

## Capítulo 4

### Resultados Obtidos

Os protótipos implementados apresentam diferentes características. Os resultados obtidos nos respectivos experimentos, sua descrição e sua análise representam o objetivo desse capítulo.

As Figura 4.1 e Figura 4.2 mostram objetos obtidos pelo primeiro protótipo implementado. A caveira da Figura 4.1 (a) foi a primeira aquisição feita nesse trabalho. O objeto original da aquisição da Figura 4.1 (b) era predominantemente marrom dificultando sua aquisição. O anjo da Figura 4.1 (c) não possui boa definição devido ao fato deste primeiro sistema possuir resolução de varredura de apenas  $0.18^\circ$ . As aquisições apresentadas na Figura 4.1 (a), (b) e (c) foram feitas como os objetos na posição vertical. Uma aquisição melhor de (c) poderia ser feita colocando-se o objeto original na posição horizontal. A aquisição do boneco de palha da Figura 4.1 (d) foi feita com o objeto na posição horizontal, mas a qualidade não foi boa devido a excessiva oclusão causada pelo próprio objeto. Uma melhor aquisição poderia ter sido feita posicionando-se o objeto em outro ângulo.

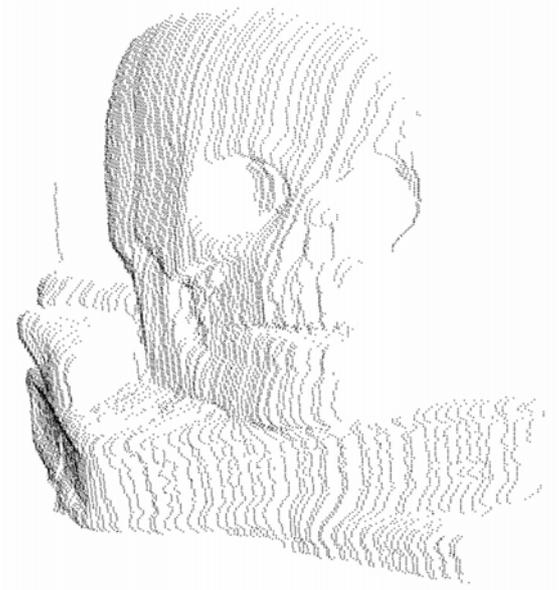
As nuvens de pontos obtidas nas aquisições efetuadas na Figura 4.2 foram feitas com os objetos originais na posição horizontal. Na Figura 4.2 (a), algumas regiões da nuvem de pontos não estão presentes; isso ocorre porque essas regiões eram de baixa refletância no objeto original. A Figura 4.2 (b) mostra a aquisição de uma estátua de anjo e as Figura 4.2 (c) e (d) apresentam aquisições do mesmo anjo utilizado como objeto original na Figura 4.1 (c), mostrando resultados melhores que a aquisição feita na Figura 4.1 (c).

As Figura 4.3 e Figura 4.4 mostram objetos obtidos pela segunda implementação. As aquisições feitas na Figura 4.3 (a), (b), (c) e (d) foram feitas em diferentes ângulos para que pudessem ser alinhadas (Figura 4.3 (e)). A Figura 4.3 (f) apresenta as aquisições alinhadas com superfície triangular que foram fundidas em um único objeto integrado (Figura 4.3 (g)). A Figura 4.4 (a) mostra a aquisição feita na Figura 4.3 (a) com superfície triangular e aplicação de textura e a Figura 4.4 (b) mostra a aquisição feita na Figura 4.3 (b) com mapeamento de cor em função da distância no eixo-z. A Figura 4.4 (c) mostra o efeito de incrustamento causado pela textura do objeto original.

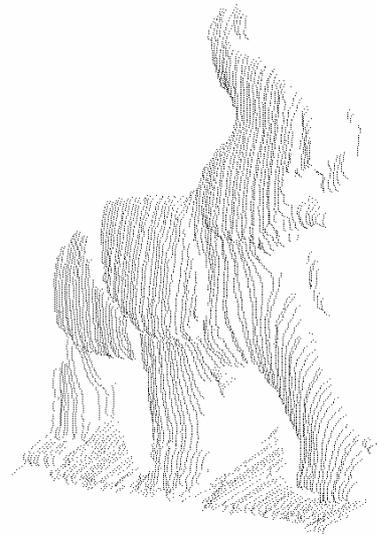
A Figura 4.5 (a), (b) e (c) mostra as nuvens de pontos obtidas pela terceira implementação do sistema de aquisição 3D. Na Figura 4.5 (a) e (b) a aquisição foi feita no modo de posicionamento, sendo que a resolução de varredura (b) é maior que a resolução de (a). A Figura 4.5 (c) mostra a aquisição feita no modo de velocidade e a Figura 4.5 (d) mostra a superfície triangular da Figura 4.5 (b).

Pode-se observar que as linhas correspondentes à posição do laser na aquisição do carro da Figura 4.5 (a) são mais precisas que as linhas da nuvem de pontos obtida na segunda implementação Figura 4.4 (c). Isso ocorre porque o sistema de varredura da terceira implementação possui *backlash* 0°. Esse efeito também pode ser notado comparando-se a Figura 4.5 (d) com a Figura 4.3 (f).

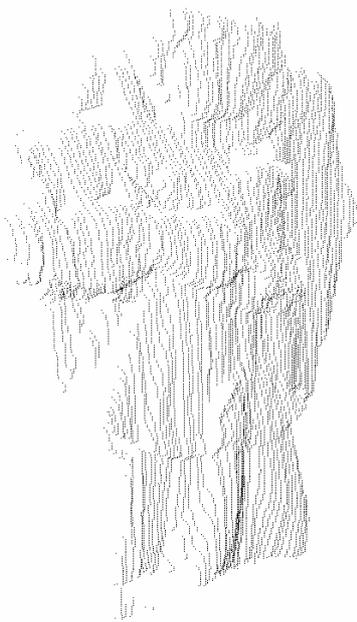
A Figura 4.6 mostra os sistemas de varredura do segundo (a) e terceiro (b) protótipos.



(a)



(b)



(c)



(d)

**Figura 4.1:** Nuvens de pontos obtidas pela primeira implementação: (a) caveira (15419 pontos), (b) elefante (20870 pontos), (c) anjo (24885 pontos) e (d) boneco de palha (22900 pontos).



(a)



(b)

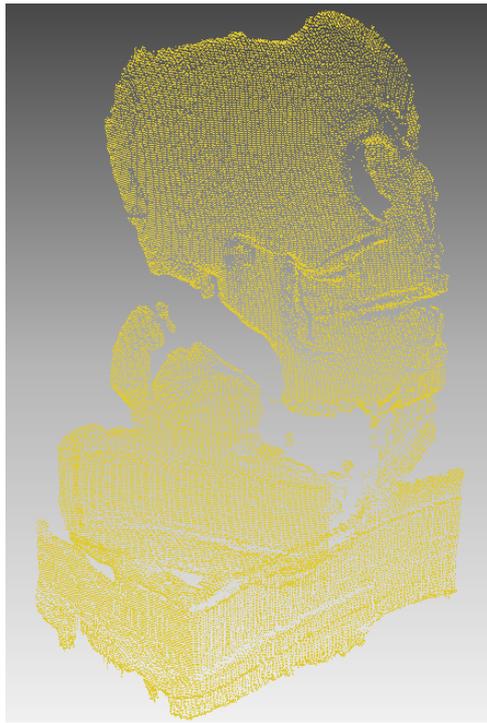


(c)



(d)

**Figura 4.2:** Nuvens de pontos obtidas pela primeira implementação com os objetos na horizontal: (a) astroboy (18579 pontos), (b) anjo de porcelana (27515 pontos), (c) anjo frente (43797 pontos) e (d) anjo costas (43333 pontos).



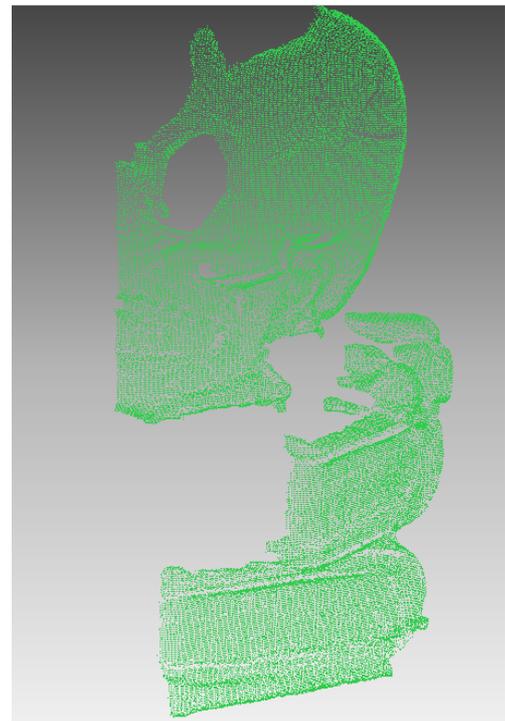
(a)



(b)

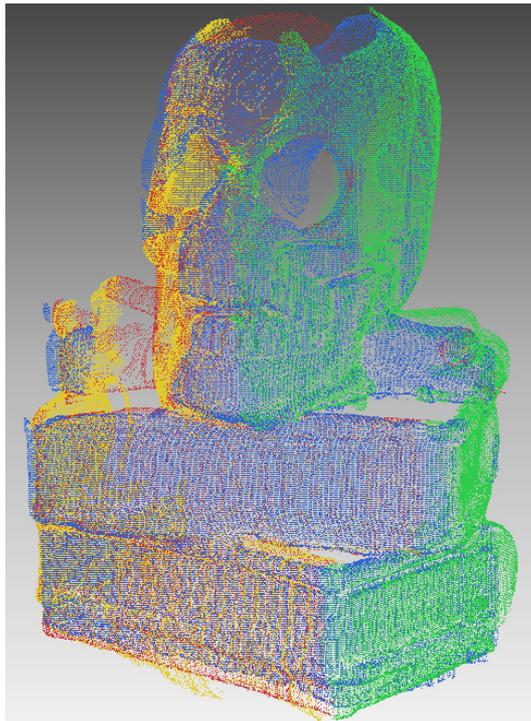


(c)

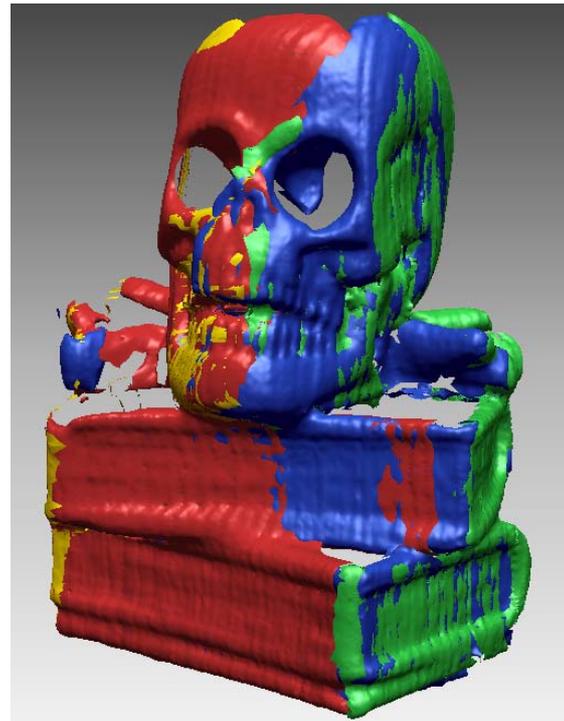


(d)

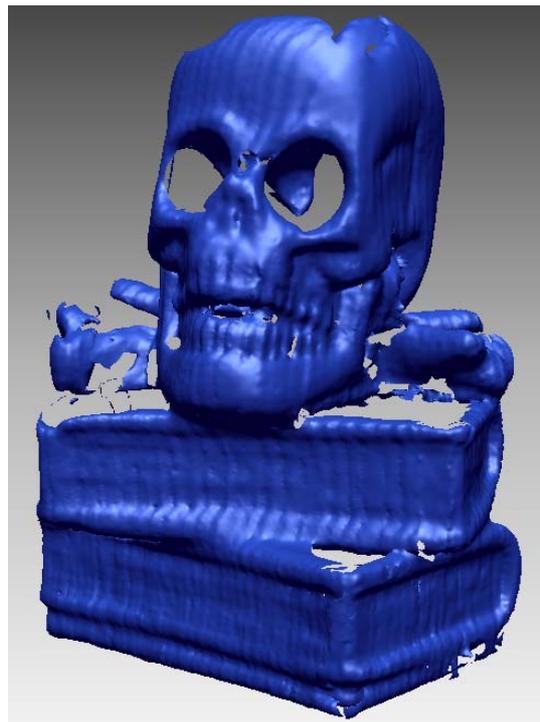
**Figura 4.3:** Nuvens de pontos obtidas pela aquisição feita em diferentes ângulos pela segunda implementação. (a) (80522 pontos), (b) (87093 pontos), (c) (92396 pontos) e (d) (69320 pontos).



(e)

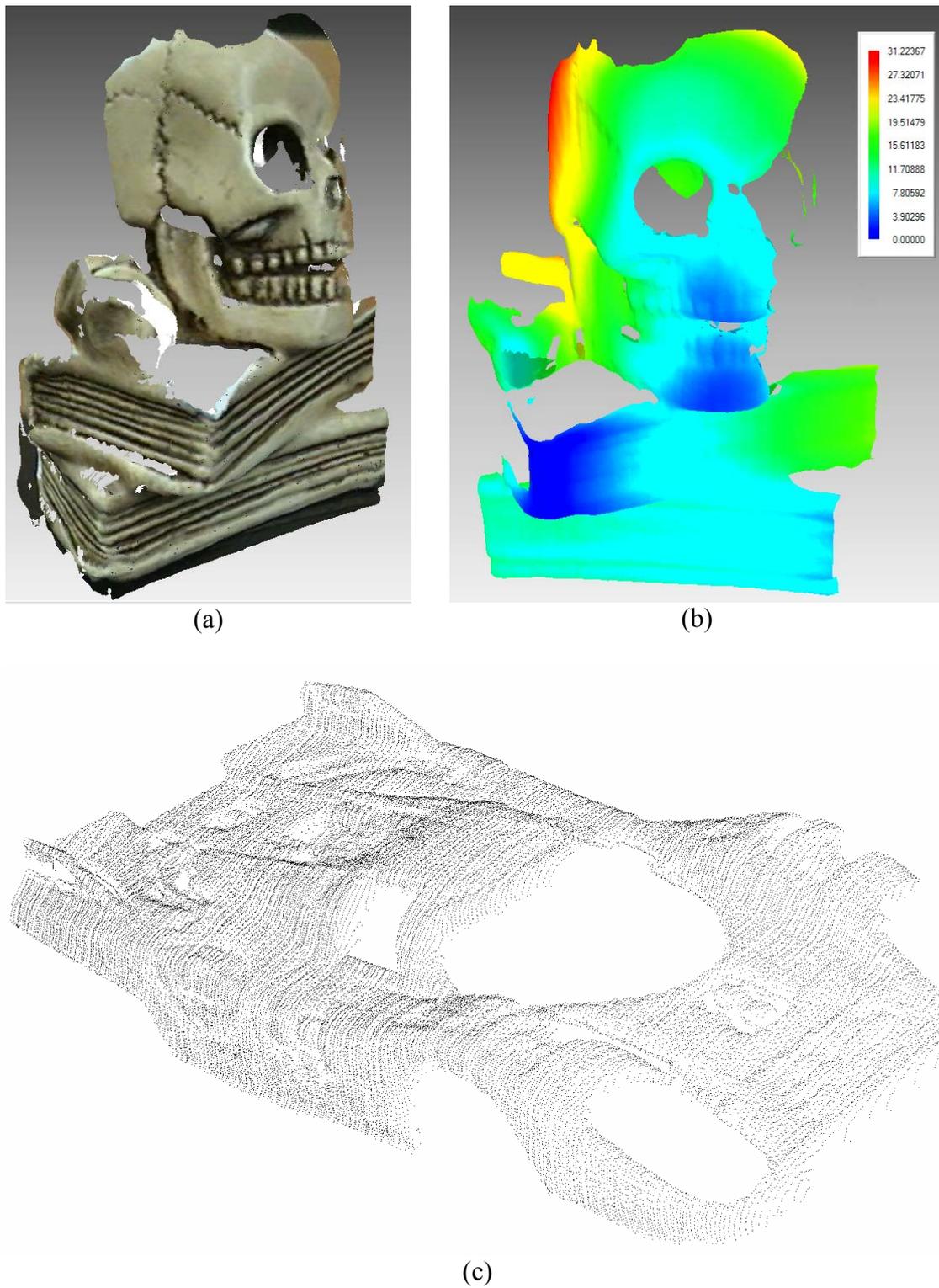


(f)

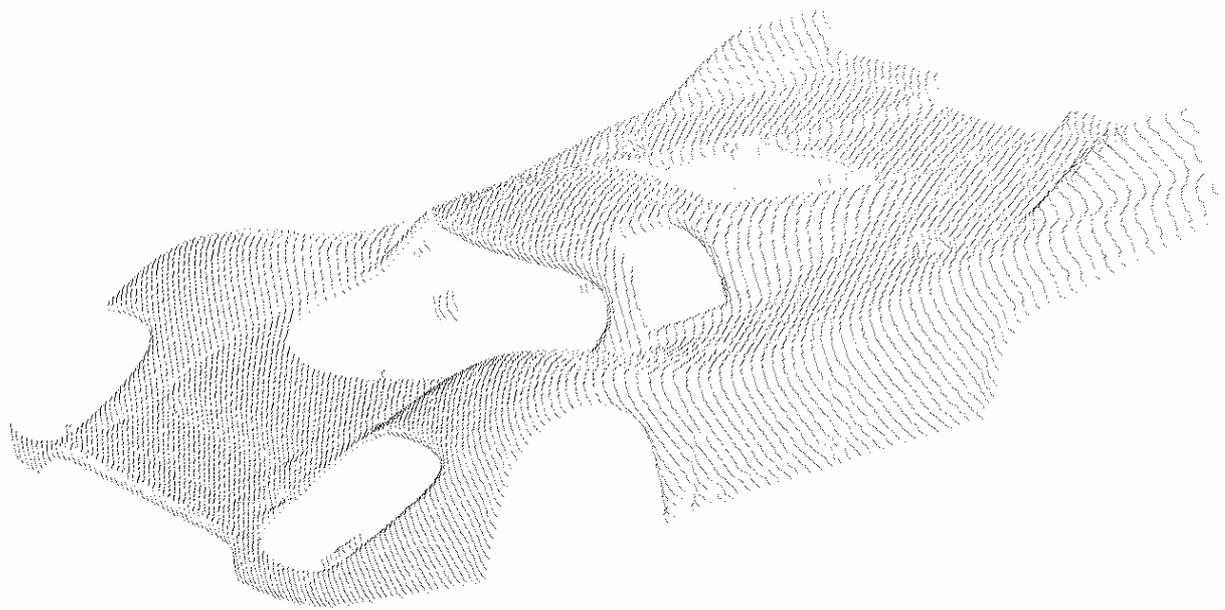


(g)

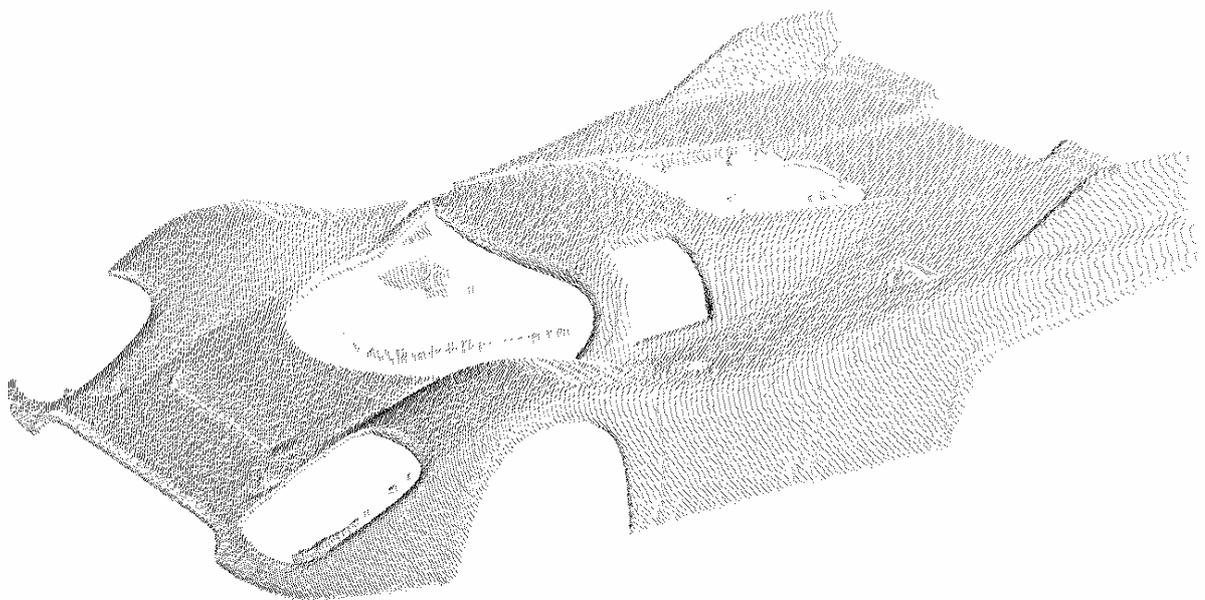
**Figura 4.3 (continuação):** (e) alinhamento entre as aquisições feitas em (a), (b), (c) e (d). (f) criação de superfície triangular de (a), (b), (c) e (d). (g) fusão entre as superfícies triangulares.



**Figura 4.4:** (a) objeto com a aplicação de textura, (b) objeto com mapeamento de cor em função da distancia z. (c) carro (57083 pontos) efeito de incrustamento causado pela textura do objeto original na aquisição.

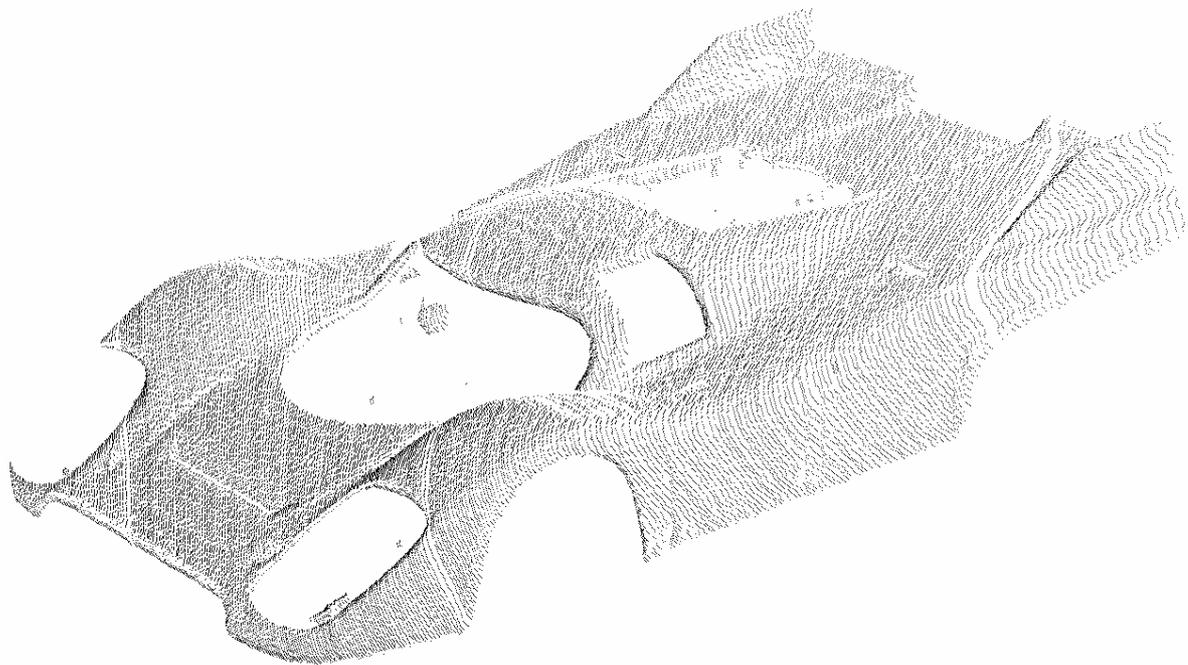


(a)

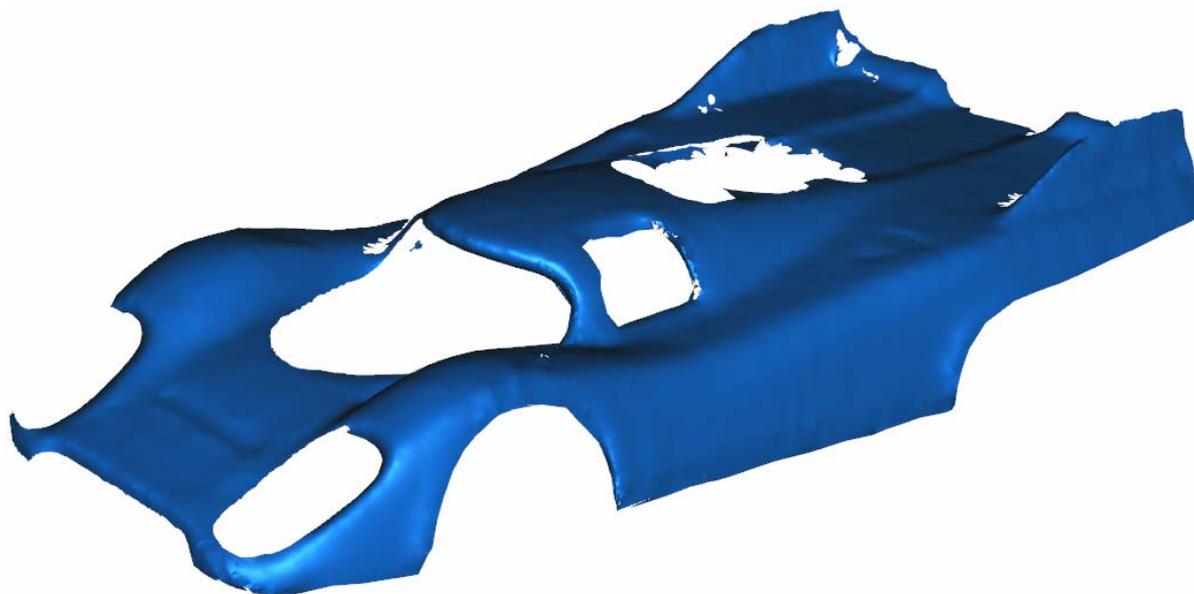


(b)

**Figura 4.5:** Nuvens de pontos obtidas pela aquisição da terceira implementação. (a) Carro (49083 pontos) e (b) (136352 pontos), aquisições feitas no modo de posicionamento. (c) carro (98223 pontos) aquisição feita no modo de velocidade e (d) superfície triangular de (b).

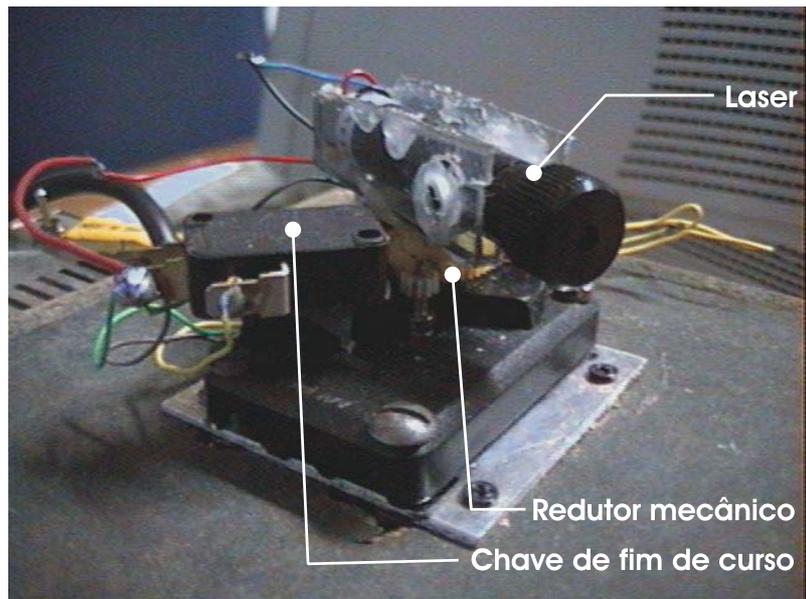


(c)

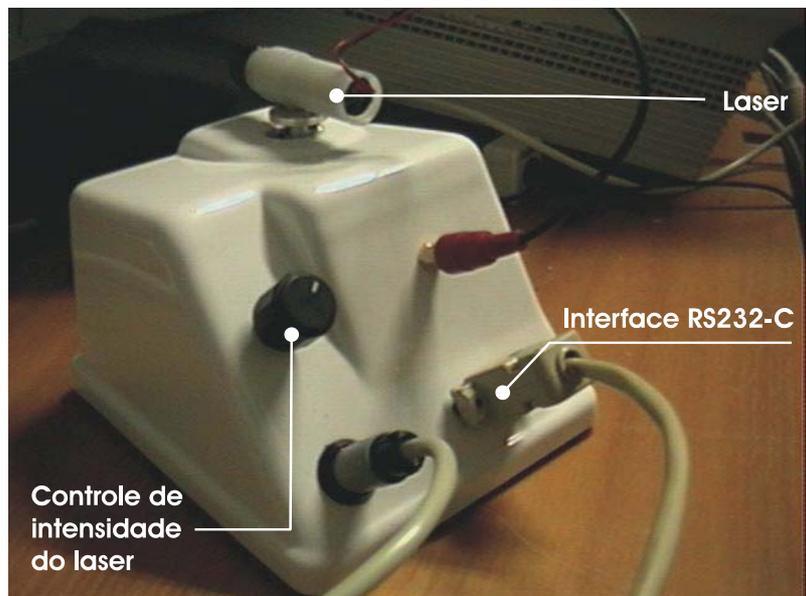


(d)

**Figura 4.5 (continuação)**



(a)



(b)

**Figura 4.6:** Sistema de varredura do segundo (a) e terceiro protótipo (b).

## Capítulo 5

### Conclusões

A área de aquisição 3D está presente em várias aplicações (por exemplo: visão robótica, levantamento topográfico e inspeção espacial), o que justifica o desenvolvimento dessa área por vários pesquisadores e grupos de pesquisa, para os mais variados objetivos.

Devido à complexidade desse trabalho não foi possível fazer um aprofundamento em todas as áreas envolvidas. Muito ainda pode ser aprimorado na ótica do módulo de sensoriamento, com a utilização de filtros e câmeras específicas para visão computacional com maior resolução e velocidade de aquisição. O módulo de estimação de distância e o software de controle podem ser implementados em FPGA, deixando todo o processamento fora do PC.

É possível melhorar a qualidade dos dados obtidos fazendo filtragens e interpolações na nuvem de pontos para redução ou eliminação de ruídos inerentes ao sistema.

Nesse trabalho, o módulo de reconstrução não foi implementado. Os sub-sistemas envolvidos nesse módulo constituem uma grande área de estudos que vem sendo desenvolvida em várias instituições de pesquisa e podem ser incorporadas ao sistema de aquisição 3D aqui desenvolvido.

Ao final desse trabalho, chegou-se à conclusão de que não é possível implementar um sistema de aquisição 3D de propósito geral. No caso da triangulação laser, cada aplicação requer uma determinada linha de base que seja adequada à faixa de operação. O sistema de aquisição de informações volumétricas, utilizando o princípio da triangulação laser e campo de visão variável, expande esses limites mas, nem por isso, torna-se um sistema de aquisição 3D de propósito geral.

É possível implementar um sistema de aquisição 3D, de baixo custo, com a finalidade de popularizar sua utilização, a exemplo dos scanners de mesa atualmente.

Eliminando-se a utilização do controlador de movimento e motores de precisão com *encoder*, pode-se estimar o ângulo de incidência do laser, utilizando-se um objeto plano de referência que esteja sempre no campo de visão da câmera, a uma distância conhecida. O objeto de interesse deve ser posicionado na frente do objeto de referência, sem bloquear, totalmente, sua visibilidade para a câmera. As primeiras

linhas na imagem da câmera devem conter, somente, as reflexões do laser na superfície do objeto de referência. Essas reflexões são utilizadas para a determinação do ângulo de incidência do laser.

## Bibliografia

---

- Amidi, O., Kanade, T., Miller, R. (1998) "Vision-based Autonomous Helicopter Research at Carnegie Mellon Robotics Institute." In Proc. Heli Japan '98. Gifu, Japan, April.
- Bouguet, J. (2004) "Camera Calibration Toolbox for Matlab", Computational Vision Group - California Institute of Technology Publications, January. [http://www.vision.caltech.edu/bouguetj/calib\\_doc](http://www.vision.caltech.edu/bouguetj/calib_doc), acessado em 04/05/04.
- Boyer, K. L., Kak, A. C. (1987) "Color-Encoded Structured Light for Rapid Active Ranging," Trans. PAMI, Vol. 9, No. 1.
- CGAL (2004) Cgal - computational geometry algorithms library, <http://www.cgal.org>, acessado em 12/07/2004.
- Cody, W. J., Waite W. (1980) "Software Manual for the Elementary Functions", Prentice-Hall, Englewood Cliffs, NJ.
- EDG (2004), "EDG Consulting Engineers", [http://www.edgweb.com/III\\_capabilities/iic4\\_cyra.html](http://www.edgweb.com/III_capabilities/iic4_cyra.html), acessado em 26/07/2004.
- França, J. G. et al. (2004) "Scanner Antropométrico para Determinação do Percentual de Gordura", Congresso Latino-Americano de Engenharia Biomédica, Setembro.
- Gluckman, J., Nayar, S. K., Thoresz, K. (1998) "Real-Time Omnidirectional and Panoramic Stereo", In Proc. of Image Understanding Workshop, volume 1, pages 299-303.
- Gois J. P., Castelo Filho, A. (2004) "Reconstrução de superfícies a partir de nuvens de pontos", 131p. Dissertação (Mestrado) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.
- Gonzalez, R. C., Woods, R. E. (1993) "Image Segmentation", Digital Image Processing, Rafael C. Gonzalez, Addison-Wesley Publishing Company, p. 413-470.

- Hall-Holt, O. and Rusinkiewicz, S. (2001) "Stripe Boundary Codes for Real-Time Structured-Light Range Scanning of Moving Objects," Proc. ICCV.
- Heikkilä J., Silven O. (1997) "A four-step camera calibration procedure with implicit image correction", In Proc. IEEE Conference on Computer Vision and Pattern Recognition, June 17-19, San Juan, Puerto Rico, 1:1106-1112.
- Heikkilä, J., Silven O. (2004) "University of Oulu: Machine Vision Group", <http://www.ee.oulu.fi/mvg/mvg.php>, acessado em 05/06/2004.
- Herm, D., Baker, M. P. (1997), "Computer Graphics: C version", Prentice Hall, Inc, Upper Saddle River, New Jersey.
- Jain, R., Kasturi, R., Schunck, B. G. (1997) "Edge Detection", Machine Vision, Eric M. Munson, McGraw-Hill, Inc. p. 144-185
- Jain, R., Kasturi, R., Schunck, B. G. (1995) "Image Filtering", Machine Vision, Eric M. Munson, McGraw-Hill, Inc. p. 112-139.
- Kohl, T.R. (1992) "VisionTutor: Lab Guide", Amerinex Artificial Intelligence, Inc., Amherst, Massachusetts.
- Levoy, M. et al. (2000) "The Digital Michelangelo Project: 3D Scanning of Large Statues", Computer Graphics Proceedings, Kurt Akeley, ACM Siggraph, p. 131-144.
- Malliot, H. A. (1996) "Digital Terrain Elevation Mapping System.", IEEE Aerospace Applications Conference.
- Miller, R., Amidi, O., Delouis, M. (1999) "*Arctic Test Flights of the CMU Autonomous Helicopter*" in Proc. Association for Unmanned Vehicle Systems, 26th Annual Symposium, July.
- Mount, D. M. "ANN Programming Manual," Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, Maryland.
- Nayar, S.K. (1997) "*Omnidirectional video camera*". In *Proc. DARPA Image Understanding Workshop*, May.

- Neptec (2004) "Neptec Design Group Ltd.", [http://www.ccc.ca/eng/suc\\_neptec.cfm](http://www.ccc.ca/eng/suc_neptec.cfm), acessado em 21/06/2004.
- O'Rourke, J. (1998) "Computational Geometry in C (Second Edition)", Cambridge University Press.
- Proesmans, M. Van Gool, L., and Defoort, F. 1998. "Reading Between the Lines: A Method for Extracting Dynamic 3D with Texture," Proc. ICCV 1998.
- Rusinkiewicz, S., Hall-Holt, O. & Levoy, M. (2002) "*Real-Time 3D Model Acquisition*", ACM Trans. On Graphics (SIGGRAPH).
- Samberg, A. (1997) "Mathematical model for time-varying system employing across track scanning." In Proc. SPIE, Vol 3065, pages 214-23.
- Samson, C. et al. (2002) "Imaging and tracking elements of the International Space Station using a 3d auto-synchronized scanner.", Proceedings of SPIE's 16 th Annual International Symposium on Aerospace/Defense Sensing, Simulations, and Controls (AeroSense 2002), Orlando, FL, NRC 44920.
- Sedgewick, R. (1988) "Algorithms", Addison-Wesley, second edition. p353.
- Sinha, S. S., Jain R. (1994) "Range Image Analysis", Handbook of Pattern Recognition and Image Processing: Computer Vision Vol.2, Young, T. Y., Academic Press Inc, p. 186-195.
- Siri, W. E., "Body composition from fluid space and density" in J. Brozek & A. Hanschel (Eds.): *Techniques for measuring body composition*, Washington, DC: National Academy of Science, pp. 223-244, 1961
- Slama, C. C. (ed.) (1980) Manual of Photogrammetry, 4<sup>th</sup> ed., American Society of Photogrammetry, Falls Church, Virginia.
- Southwell, D. et al. (1996) "Panoramic stereo". In Proc. Int'l Conference on Pattern Recognition
- Sun, Y., Dumont C., Abidi M. (2000) "Mesh-based Integration of Range and Color Images", SPIE's 14th International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, 4051, 2000, pp 110--117.

Tiziani, H. J. (1997) "Optical Metrology of Engineering Surfaces-Scope and Trends", Optical Measurement Techniques and Applications, Rastogi, P. K., Norwood, Artech House Inc., p. 15-22.

Umbaugh, S. E. (1998) "Computer Vision and Image Processing: A Practical Approach Using CVIPtools", Prentice Hall.