

JULIO TABEGNA

**RECUPERAÇÃO DE FORMA DE PEÇAS
ESTAMPADAS POR LUZ ESTRUTURADA**

**Dissertação apresentada à Escola
Politécnica da Universidade de
São Paulo para a obtenção do
Título de Mestre em Engenharia.**

**Área de Concentração:
Engenharia Mecatrônica**

**Orientador:
Prof. Dr. Oswaldo Horikawa**

**São Paulo
2004**

AGRADECIMENTOS

Agradeço primeiramente à General Motors do Brasil por permitir não somente a execução dos ensaios e utilização dos equipamentos da empresa, como também por disponibilizar carga horária e regimento flexível de trabalho, sem os quais não seria possível a conclusão do curso de mestrado.

Ao professor Dr. Oswaldo Horikawa pelas colocações e questionamentos que sempre acrescentaram ao desenvolvimento desta dissertação.

À minha namorada Fabiana e à minha família por toda a paciência e incentivo nos momentos de dificuldade.

Por fim, agradeço a todos os amigos que de alguma maneira contribuíram para a realização deste trabalho.

RESUMO

É proposto um método para a recuperação de forma de componentes automobilísticos estampados. O método utiliza a luz estruturada com padrão de projeção binário. São apresentados o equacionamento matemático e uma simplificação para a obtenção das coordenadas tridimensionais de um objeto. A implementação do método é feita com o uso de uma máquina fotográfica digital, um projetor de multimídia e um microcomputador permitindo portabilidade ao sistema. A simulação digital é utilizada para analisar os erros obtidos na medição dos posicionamentos de projetor e câmera. Os resultados dos ensaios preliminares mostraram a falta de precisão na metade da imagem capturada do lado oposto ao da câmera, bem como a ocorrência das áreas de oclusão. A fim de superar estas limitações, uma nova abordagem foi estudada. Trata-se da captura de informações do objeto sob posicionamento oposto de câmera em relação ao projetor, da seleção das informações mais precisas e da combinação de resultados em prol do aperfeiçoamento da reconstrução em três dimensões. O sistema é calibrado por um processo numérico baseado em algoritmo genético. Ao final, todo o desenvolvimento é aplicado à recuperação de forma de peças estampadas. O resultado é comparado à medição feita por uma máquina de medição por coordenadas. Verificou-se que o sistema é capaz de recuperar a forma de peças estampadas com rapidez e com um erro máximo de 2,1 mm.

ABSTRACT

A method for the shape recovery of stamped parts for vehicles is proposed. The method uses the structured light with a binary pattern. The mathematical equationing and a simplification to the tridimensional coordinates acquisition of an object are presented. The implementation of this method is done with a digital camera, a LCD projector and a microcomputer allowing portability to the system. The digital simulation is used to analyse the errors from the projector and camera positioning measurements. Test results show the lack of precision on the half of the image captured in the opposite side of the camera, as well as occurrence of hiding areas. In order to overcome these limitations, a new approach was studied. It treats on the capture of the object information in the opposite side of the camera in respect to the projector, the more precise information filtering and the results combination to reach the improvement of the three dimensional reconstruction. The system is calibrated by a numeric process based on genetic algorithm. Finally, the whole development is applied to the shape recovery of stamped parts. The result is compared to the measurement made by a coordinate measurement machine. It was found that the system is capable of shape recovery of stamped parts quickly and with 2,1 mm of maximum error.

SUMÁRIO

RESUMO

ABSTRACT

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CAPÍTULO I - INTRODUÇÃO	1
1.1 Motivação.....	1
1.2 Métodos de Recuperação de Forma.....	7
1.2.1 Métodos Contactantes	8
1.2.1.1 Instrumentos Convencionais	9
1.2.1.2 Máquina de Medição por Coordenadas (MMC).....	10
1.2.2 Métodos Ópticos	11
1.2.2.1 Triangulação	11
1.2.2.2 Interferometria	12
1.2.2.3 Holografia.....	13
1.2.2.4 Processamento de Imagens.....	14
1.2.2.4.1 Métodos Estáticos	14
1.2.2.4.2 Métodos Dinâmicos	15
1.2.2.5 Tomografia	16
1.2.2.6 Franjas de Moiré	17
1.2.2.7 Luz Estruturada.....	19
1.3 Particularidades dos Componentes Estampados	23

1.4 Síntese Inicial	26
1.5 Objetivo.....	27
1.6 Metodologia.....	29
1.7 Inovações.....	30
CAPÍTULO II – DESCRIÇÃO DO MÉTODO DE RECUPERAÇÃO DE FORMA EMPREGADO	32
2.1 Composição do Sistema.....	32
2.2 Modelagem do Sistema.....	32
2.2.1 Posicionamento Genérico dos Equipamentos	33
2.2.2 Modelo de Câmera.....	33
2.2.3 Equacionamento do Sistema	34
2.2.4 Simplificação da Triangulação	41
CAPÍTULO III – IMPLEMENTAÇÃO DO MÉTODO DE RECUPERAÇÃO DE FORMA ATRAVÉS DE ENSAIOS PRELIMINARES.....	46
3.1 Equipamentos do Sistema	46
3.2 Objetos a Ensaiar	47
3.3 Captura das Imagens.....	48
3.4 Tratamento das Imagens	50
3.4.1 Pré-processamento	50
3.4.2 Processamento	52
3.5 Análise de Resultados.....	54
3.5.1 Análise de Erro	55
3.5.1.1 Simulação	56
3.5.2 Áreas de Oclusão	58
3.5.3 O Efeito do Ângulo da Câmera	58

3.6 Limitações do Sistema	59
CAPÍTULO IV – NOVA ABORDAGEM	60
4.1 Descrição da Nova Abordagem.....	60
4.2 Calibração do Sistema.....	62
4.3 Ensaio da Nova Abordagem.....	64
CAPÍTULO V – RECUPERAÇÃO DE FORMA DE PEÇAS	
ESTAMPADAS	67
5.1 Peças Estampadas Utilizadas.....	67
5.2 Preparação dos Ensaios.....	67
5.3 Resultados de Recuperação de Forma	68
5.3.1 Comparação de Medição.....	69
5.3.2 Comparação de Forma	73
6 Conclusões	74
ANEXO A – CÁLCULOS DE PONTO FOCAL.....	76
ANEXO B – EQUACIONAMENTO DO SISTEMA	77
B.1 Determinação do Ponto S	77
B.2 Determinação do Ponto Q	78
B.3 Determinação do Ponto T	82
B.4 Determinação do Ponto R	85
B.5 Determinação do Termo Δx	86
B.6 Determinação do Ponto P	86
B.7 Conversão de Sistema de Coordenadas.....	87
B.7.1 Rotação	87
B.7.2 Translação.....	88

B.8 Determinação do Ponto Q'	89
B.9 Determinação do Ponto T'	89
ANEXO C – CÓDIGO FONTE	91
C.1 Detecção de Vértices e Simplificação da Triangulação	91
C.2 Filtro	95
C.3 Simulação de Paralelepípedo	98
C.4 Simulação de Projeção	100
C.5 Simulação de Captura	103
C.6 Calibração	109
C.7 Processamento	115
C.8 Biblioteca	119
REFERÊNCIAS BIBLIOGRÁFICAS	125

LISTA DE FIGURAS

Figura 1-1 Fluxo de informações no desenvolvimento de um novo veículo.....	2
Figura 1-2 Complementos de estampabilidade	3
Figura 1-3 Sequência de desenvolvimento de complementos	3
Figura 1-4 Resultado da análise de estampagem por elementos finitos de um componente	4
Figura 1-5 Etapas no projeto de ferramenta de estampagem.....	4
Figura 1-6 Usinagem de modelo de isopor.....	5
Figura 1-7 Preparação para a fundição de ferramenta	5
Figura 1-8 Modificações dos complementos de um componente.....	6
Figura 1-9 Modelo físico e digital de um mesmo componente	6
Figura 1-10 Métodos de recuperação de forma	8
Figura 1-11 Medição com gabarito de raios.....	9
Figura 1-12 Medição de peça com MMC.....	10
Figura 1-13 Representação esquemática da técnica de triangulação	12
Figura 1-14 Tipos de reflexão de luz.....	13
Figura 1-15 Representação das fases de gravação e reconstrução em holografia.....	13
Figura 1-16 Representação esquemática do método de rotação do objeto	15
Figura 1-17 Esquema do princípio da técnica de tomografia	16
Figura 1-18 Representação esquemática do método das Franjas de Moiré	17
Figura 1-19 Representação esquemática do método de luz estruturada	19
Figura 1-20 Estampa de projeção com codificação binária.....	20
Figura 1-21 Estampa de projeção com codificação por intensidade luminosa	21
Figura 1-22 Simetria de informações em componente estampado	23
Figura 1-23 Formas de componente estrutural e estético	24
Figura 1-24 Fenômeno do retorno elástico de um componente.....	25
Figura 2-1 Posicionamento genérico dos equipamentos	33
Figura 2-2 Representação do modelo de câmera.....	34

Figura 2-3 Esquema óptico geométrico do sistema.....	35
Figura 2-4 Representação do termo Δx	39
Figura 2-5 Imagem com aspecto trapezoidal.....	39
Figura 2-6 Projeção das borda da imagem no plano π'	41
Figura 2-7 Projeção das bordas da imagem transformada no plano π	41
Figura 3-1 Equipamentos do sistema de recuperação de forma.....	47
Figura 3-2 Objetos utilizados para a recuperação de forma com dimensões em milímetros.....	48
Figura 3-3 Posicionamento dos equipamentos.....	49
Figura 3-4 Imagens capturadas pela câmera dos objetos.....	49
Figura 3-5 Rotina do pré-processamento.....	50
Figura 3-6 Imagens após a simplificação da triangulação.....	51
Figura 3-7 Imagens após a aplicação de filtro.....	52
Figura 3-8 Reconstrução de paralelepípedo e meia esfera.....	53
Figura 3-9 Corte aplicado ao longo do comprimento do paralelepípedo.....	54
Figura 3-10 Aumento de precisão para as regiões próximas à câmera.....	55
Figura 3-11 Imagem de paralelepípedo criado por simulação.....	56
Figura 3-12 Imagem de simulação de projeção sobre o paralelepípedo.....	57
Figura 3-13 Imagem de simulação de captura da câmera.....	57
Figura 3-14 Área de oclusão com aumento de altura do objeto.....	58
Figura 3-15 Área de oclusão devido a variação do ângulo α	59
Figura 4-1 Esquema óptico geométrico do sistema na nova abordagem.....	60
Figura 4-2 Formação trapezoidal da imagem na nova abordagem.....	61
Figura 4-3 Esquema de calibração por algoritmo genético.....	63
Figura 4-4 Imagens capturadas pela câmera posicionada à direita e à esquerda do projetor.....	65
Figura 4-5 Reconstrução da meia esfera ensaiada na nova abordagem.....	66
Figura 5-1 Componentes estampados adotados para a reconstrução.....	67
Figura 5-2 Imagens capturadas pela câmera do suporte da estrutura do amortecedor.....	68
Figura 5-3 Reconstrução do suporte da estrutura do amortecedor.....	69
Figura 5-4 Referência adotada e seção na peça na medição pela MMC.....	70

Figura 5-5 Imagem de captura dos pontos adquiridos pela MMC num software CAD	70
Figura 5-6 Esquema de corte na medição por luz estruturada na peça	71
Figura 5-7 Comparação de medição	71
Figura 5-8 Reconstrução e imagem real do inserto da lanterna traseira	73
Figura 5-9 Reconstrução e imagem real da caixa da roda superior dianteira	73
Figura A-1 Esquema para cálculo de ponto focal.....	76
Figura B-1 Esquema para determinação da equação do plano π'	78
Figura B-2 Representação da rotação do sistema de coordenadas	87
Figura B-3 Representação da translação do sistema de coordenadas.....	88

LISTA DE TABELAS

Tabela 2-1 Coordenadas geométricas dos pontos da figura 2-7	42
Tabela 3-1 Valores obtidos para os posicionamentos do projetor e da câmera	53
Tabela 3-2 Resultados numéricos obtidos para a seção do paralelepípedo	54
Tabela 4-1 Valores medidos para os posicionamentos do projetor e da câmera	64
Tabela 4-2 Valores obtidos por calibração para os posicionamentos do projetor e da câmera.....	65
Tabela 5-1 Valores obtidos por calibração no posicionamento do projetor e da câmera para os componentes estampados	68

LISTA DE ABREVIATURAS E SIGLAS

2D.....duas dimensões

3D.....três dimensões

CAD“Computer Aided Design” (Projeto Assistido por Computador)

CAM.....Computer Aided Manufacturing” (Fabricação Assistida por Computador)

CCD.....“Charge-couple Device” (Dispositivo de Carga Acoplada)

HSI“Hue, Saturation and Intensity” (Matiz, Saturação e Intensidade)

ISO“International Organization for Standardization” (Organização Internacional para a Normalização)

MMCMáquina de Medição por Coordenadas

CAPÍTULO I – INTRODUÇÃO

1.1 Motivação

A alta competição industrial do mundo globalizado tem colocado prazos cada vez mais curtos no desenvolvimento de novos produtos. Ao mesmo tempo, há exigências no mercado de produtos no sentido de que estes agreguem mais funções e sejam cada vez mais eficientes. Desta forma, tornaram-se emergentes novas tecnologias e métodos de desenvolvimento de novos produtos.

Foi neste momento que as metodologias tradicionais de engenharia direta cederam espaço a uma nova abordagem: a engenharia reversa.

“Engenharia reversa é a análise individual de um objeto para verificar seu funcionamento a fim de duplicá-lo ou aprimorá-lo” [1]. Isto significa partir do produto final com o objetivo de melhorá-lo. Trata-se de uma metodologia que corresponde ao caminho inverso do processo de desenvolvimento direto.

Historicamente, utilizou-se a engenharia reversa no Japão para evitar gastos com novos projetos. A partir da análise de forma, função, tolerância de montagem e processos de fabricação de produtos existentes, houve um melhor entendimento da funcionalidade, permitindo que um produto de melhor qualidade fosse criado [2].

A literatura disponível que trata de engenharia reversa é abundante. Dentre ela, Varady; Marty; Coxt [3] apresentam as vantagens e desvantagens de vários métodos de aquisição de dados, bem como as limitações do uso de engenharia reversa. Dankwort; Podehl [4] expõem as dificuldades nos métodos tradicionais de desenvolvimento de estilo no segmento automobilístico, da interação com o modelo físico à manipulação das superfícies em CAD, e também as iniciativas em engenharia reversa. A metodologia de engenharia reversa é revista por Germani; Mandorli; Berti [5] nos aspectos custo, tempo e habilidade de operação frente aos métodos tradicionais de engenharia.

Neste cenário, tecnologias que permitam a recuperação da forma do produto físico final e que, de alguma maneira, o compare com as informações matemáticas de projeto, tornaram-se parte importante no desenvolvimento de novos produtos.

A necessidade da recuperação da forma do produto final é verificada no processo de fabricação de componentes automobilísticos metálicos estampados na General Motors do Brasil.

O desenvolvimento de um novo veículo inicia-se na área de “Design” a partir da definição das formas do automóvel. Nesta fase, criam-se arquivos matemáticos CAD que reproduzem o estilo do carro. Na seqüência, são desenvolvidos os componentes do veículo pela Engenharia de Produtos. Os componentes de formas definidas são dimensionados a fim de atender as solicitações de trabalho no automóvel, como por exemplo, de caráter estrutural, térmico e acústico. A seguir, a Engenharia de Manufatura recebe cada componente do veículo nos mesmos formatos CAD e estabelece a maneira que eles serão fabricados. No caso dos componentes automobilísticos metálicos estampados, são definidas informações como o material do componente, a quantidade de operações de estampagem, a força necessária para a estampagem e o posicionamento do componente nas prensas para a fabricação.

A figura 1-1 mostra o fluxo de informações no desenvolvimento de um novo veículo, do “Design” à Engenharia de Manufatura.



Figura 1-1 – Fluxo de informações no desenvolvimento de um novo veículo.

A etapa seguinte à definição dos meios de fabricação dos componentes metálicos estampados é a Engenharia de Ferramental. A Engenharia de Ferramental recebe os arquivos CAD de cada componente automobilístico estampado para projetar a ferramenta de estampagem. O arquivo CAD contém apenas as informações dos componentes estampados finais. Assim, numa primeira etapa são desenvolvidos todos os complementos do componente, isto é, todas as formas de cada operação de estampagem que resultará na obtenção do componente final. A figura 1-2 mostra um componente final (pára-lamas) e os complementos desenvolvidos para sua estampabilidade.

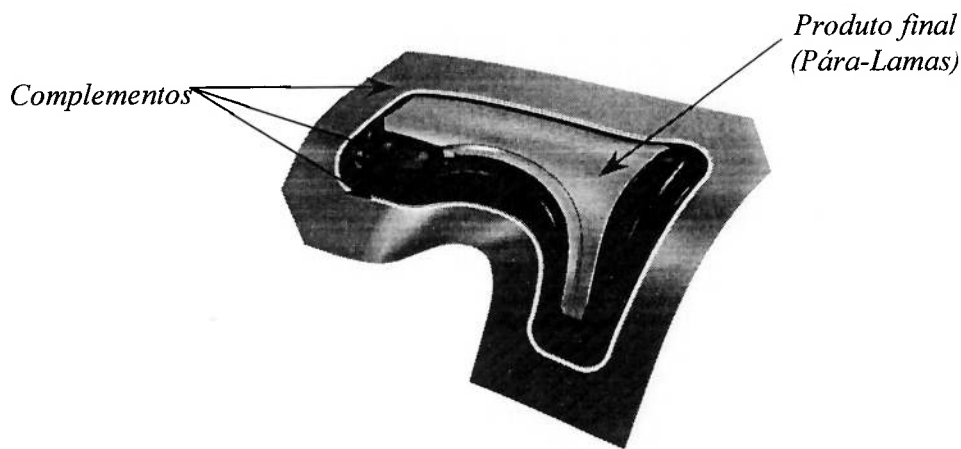


Figura 1-2 – Complementos de estampabilidade.

No desenvolvimento de complementos, faz-se o caminho reverso num processo de estampagem normal. Desta forma, inicia-se o desenvolvimento pela última operação até a primeira. No exemplo da figura 1-3, o desenvolvimento é iniciado pelo recorte e puncionamento, seguido pelo flangeamento e forma, recalque e flangeamento, recorte, repuxo e finalmente o blanque liso inicial.

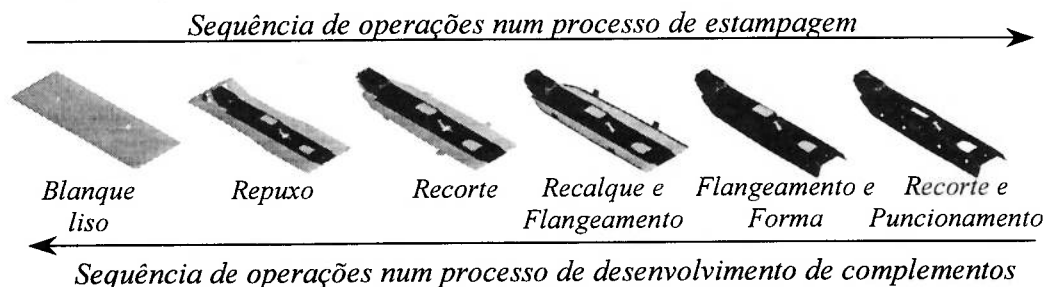


Figura 1-3 – Seqüência de desenvolvimento de complementos.

Após a criação das linhas e superfícies de cada operação de estampagem, é feita a simulação de estampagem dos componentes por elementos finitos a fim de verificar quaisquer defeitos de estampagem, como ruptura, enrugamento e afinamento da chapa. Caso haja necessidade, novos complementos são desenvolvidos a fim de se obter nesta etapa componentes estampados isentos de defeitos. A figura 1-4 mostra um resultado não satisfatório de simulação de estampagem de um componente.

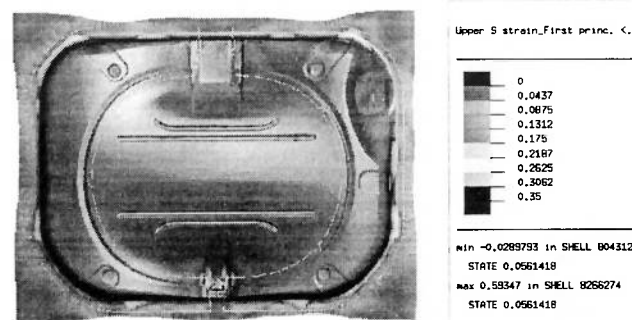


Figura 1-4 – Resultado da análise de estampagem por elementos finitos de um componente.

A partir de resultados satisfatórios na simulação de estampagem e das informações de processo e produto previamente definidas, além das normas de projeto, é executado o projeto da ferramenta de estampagem no sistema CAD. Algumas etapas do projeto de uma ferramenta são mostradas na figura 1-5. Observa-se que o projeto da ferramenta é feito seqüencialmente de dentro para fora, ou seja, a partir das superfícies da peça do automóvel para os componentes da ferramenta.

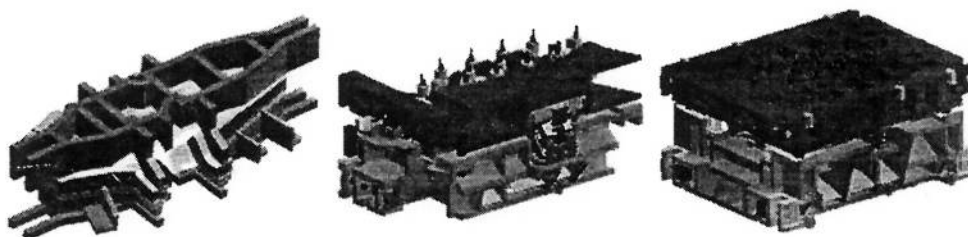


Figura 1-5 – Etapas no projeto de ferramenta de estampagem.

Existe, então, a necessidade de criação de um modelo de isopor para que as ferramentas possam ser feitas de ferro fundido. Assim, finalizado o projeto da ferramenta, um modelo de isopor é criado por máquinas de usinagem CAM. Toda a

informação contida no arquivo CAD do projeto da ferramenta é transmitida via sistema à máquina de usinagem. A figura 1-6 mostra a ferramenta tomando forma física através da máquina que usina o modelo de isopor.

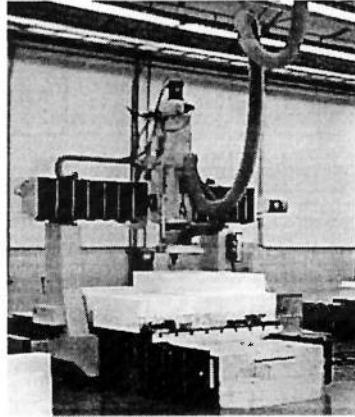


Figura 1-6 – Usinagem de modelo de isopor.

Na fundição, são preparadas caixas de areia sobre o modelo de isopor para que o ferro fundido incandescente tome a forma exata do isopor. Cada componente da ferramenta — como base, sujeitador, faca, punção e matriz — é fundido separadamente. A figura 1-7 mostra a preparação para a fundição de uma ferramenta.



Figura 1-7 – Preparação para a fundição de ferramenta.

Depois de fundida, a ferramenta física passará por usinagens finais, retirando-se o metal bruto e atribuindo-se a ela qualidade de superfície final de estampagem através de máquinas de usinagem, que trabalham com as informações

definidas pelo projeto de ferramentas. A próxima etapa constitui a montagem de cada componente na ferramenta. Finalmente, para que a ferramenta comece a estampar, ajustes finais são feitos na ferramenta. Todos os ajustes são realizados manualmente e baseados na experiência de cada ferramenteiro. Alguns complementos dos componentes fazem parte destes ajustes na ferramenta a fim de que se obtenha um produto estampável. A figura 1-8 mostra um ferramenteiro modificando os complementos de um componente.

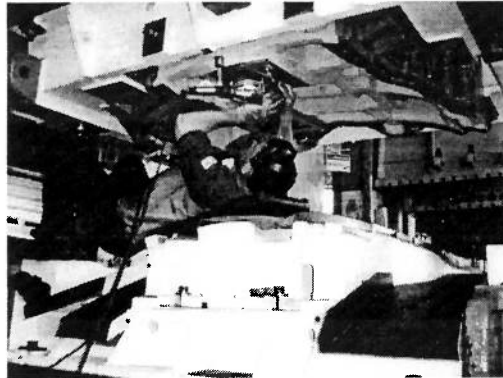


Figura 1-8 – Modificações dos complementos de um componente.

Após as modificações, os componentes não passam por medição dos complementos. Desta forma, o componente matemático especificado pelo projeto (modelo digital) não é atualizado, permitindo que o componente em fase de ajustes (modelo físico) tenha complementos diferentes do modelo digital, como esquematiza a figura 1-9.

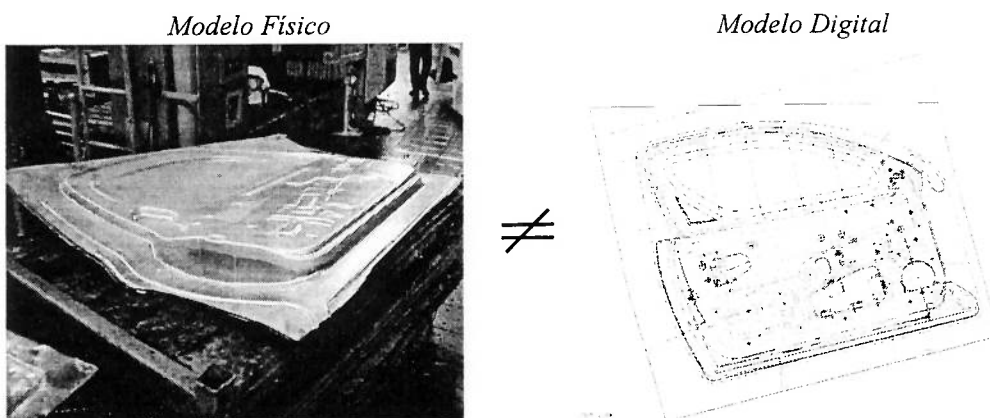


Figura 1-9 – Modelo físico e digital de um mesmo componente.

Dentro da grande variedade de componentes estampados de um automóvel, existem aqueles que são semelhantes em automóveis de diferentes modelos. A semelhança entre os componentes de diferentes modelos torna possível a prática de modificações também semelhantes a cada novo automóvel nesta fase de ajustes finais.

Nesta fase, faz-se necessário o uso da recuperação de forma. A função do sistema de recuperação de forma é a de obter as modificações práticas nos complementos dos componentes. As informações das modificações serão consideradas no projeto de novos produtos, impedindo que os complementos dos componentes sofram modificações na fase de ajustes finais.

1.2 Métodos de Recuperação de Forma

Com o objetivo de se determinar um método adequado para a recuperação de forma de peças estampadas, foram analisados os métodos disponíveis.

Nesta pesquisa, verificou-se que a recuperação de formas é uma área extremamente ampla, visto os diferentes campos de interesse, tais como indústria, tecnologia, lazer e medicina. Portanto, é natural que existam diferentes métodos adequados a cada campo de interesse.

Os métodos de recuperação de forma analisados foram subdivididos em dois grupos: métodos contactantes e ópticos, conforme mostra a figura 1-10.

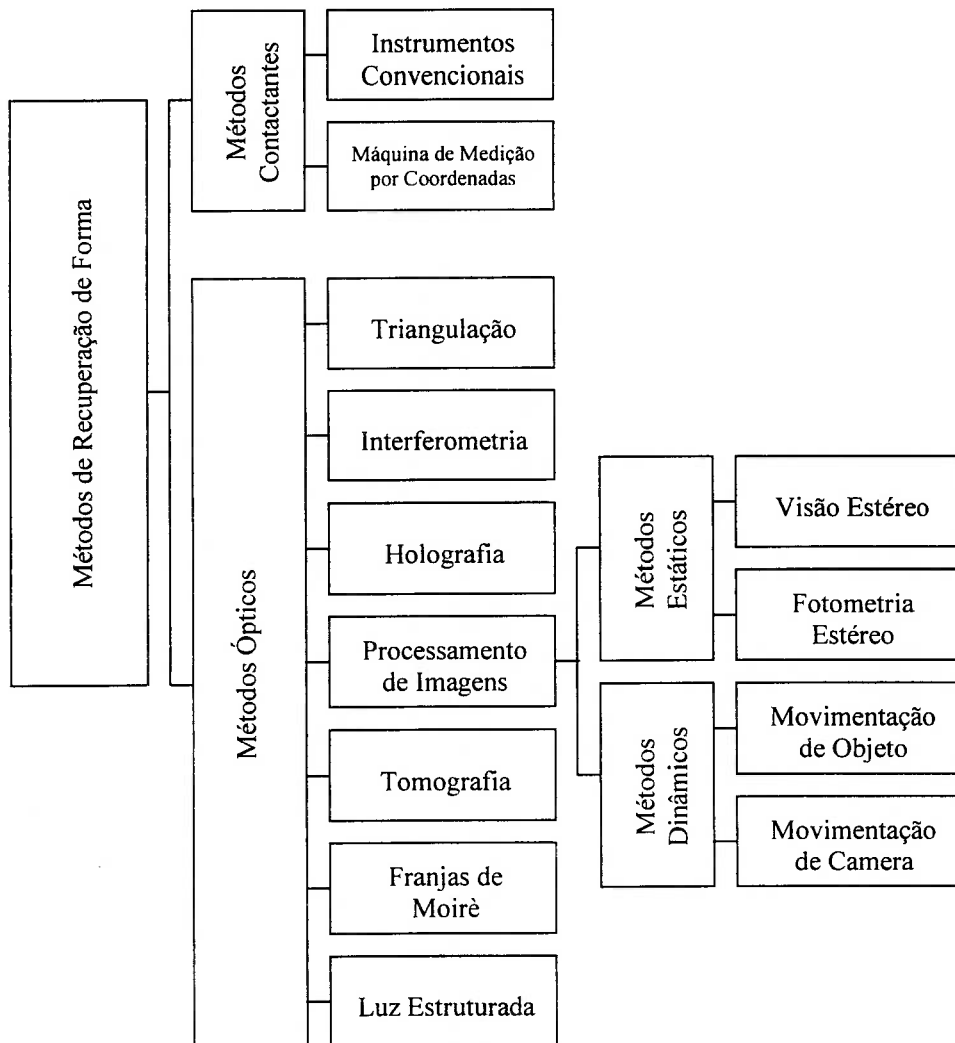


Figura 1-10 – Métodos de recuperação de forma.

1.2.1 Métodos Contactantes

A tendência histórica de tocar o objeto a fim de medi-lo tornou este método mais desenvolvido ao longo do tempo, permitindo que os equipamentos de

medição por contacto estivessem bastante acessíveis no mercado. Subdividiu-se este método em dois subgrupos: os instrumentos convencionais e as máquinas de medição por coordenadas.

1.2.1.1 Instrumentos Convencionais

Neste subgrupo, classificam-se os instrumentos de metrologia mais conhecidos como trenas, paquímetros, micrômetros e gabaritos de raios, entre outros.

São equipamentos de grande facilidade de trabalho, não exigindo treinamentos específicos de operação. Devido às dimensões reduzidas, permitem bom manuseio e mobilidade no local de medição, apresentando também grande flexibilidade de aplicação.

No entanto, algumas limitações de medição caracterizam estes equipamentos, causando erros de leitura. Um exemplo disso é a medição por comparação visualizada na figura 1-11 com um gabarito de raios.

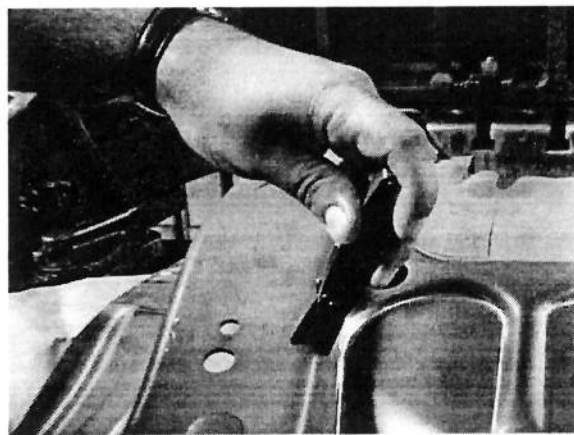


Figura 1-11 – Medição com gabarito de raios.

1.2.1.2 Máquina de Medição por Coordenadas (MMC)

A Máquina de Medição de Coordenadas (MMC) é um equipamento que localiza as coordenadas de pontos de uma peça em estruturas tridimensionais. A medição numa MMC é executada com o uso de três braços móveis e perpendiculares da máquina. Os deslocamentos destes braços são gravados por transdutores ou dispositivos de medição numa referência fixa à medida que um sensor toca a peça.

Graças à evolução dos computadores por volta de 1960, as máquinas de medição de coordenadas tornaram-se um poderoso recurso de medição nas indústrias. Desde então, as MMCs passaram a fazer parte do processo de inspeção de produtos. A figura 1-12 mostra uma MMC utilizada na inspeção de um componente automobilístico estampado.

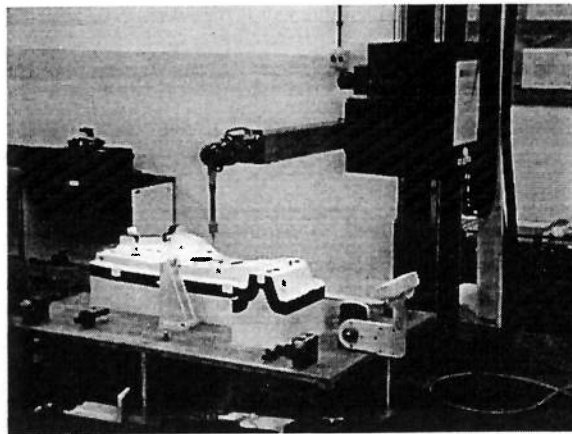


Figura 1-12 – Medição de peça com MMC.

De todos métodos contactantes, este é o que apresenta maior precisão. Segundo Bush; Harlom; Thompson [6], a precisão de uma MMC é 10,16 μm . Por sua vez, Becker; Heynacher [7] apresentaram uma MMC com precisão de nanômetros.

Sabe-se que, para a medição da peça visualizada na figura 1-12 com 1200 pontos, são necessárias aproximadamente 3 horas, conforme informações da Sala de Medições de Ferramental da General Motors do Brasil.

Apesar das MMCs tradicionais não serem móveis, ocupando espaços físicos relativamente grandes (3000 x 2500 x 1750 mm), já existem braços de medição portáteis de menores dimensão e capacidade, mas com o mesmo princípio de funcionamento das MMCs tradicionais. É necessário controlar as condições de temperatura, umidade e vibração das MMCs no ambiente de trabalho.

A recuperação de forma de peças estampadas com a MMC demanda muito tempo, trabalha-se com alta precisão, necessita-se de um ambiente especial para a medição e a automação da medição é limitada. Além disso, são necessários treinamentos específicos de operação, impedindo que operadores não-especialistas utilizem o equipamento.

1.2.2 Métodos Ópticos

Os métodos contactantes apresentam limitações quanto à precisão, no caso dos instrumentos convencionais, ou quanto à velocidade de medição e ao ambiente especial de trabalho, no caso da MMC. Foram verificadas alternativas de medição através dos métodos ópticos que superam as limitações dos métodos contactantes. Jarvis [8] apresenta uma visão geral dos métodos ópticos e suas aplicações. A seguir, será apresentada uma descrição mais detalhada dos métodos ópticos.

1.2.2.1 Triangulação

O princípio deste método está na emissão de um feixe de luz sobre a superfície de um objeto e observação por um sistema óptico. A partir da posição observada no sistema óptico, da distância entre a fonte do feixe luminoso e o sistema

óptico, e da direção do feixe, calcula-se a posição na superfície do objeto, obtendo-se a forma total do mesmo.

A figura 1-13 mostra o esquema da técnica de triangulação.

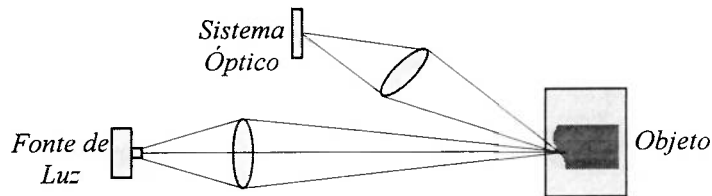


Figura 1-13 – Representação esquemática da técnica de triangulação.

Existem várias aplicações industriais deste método, algumas das quais são apresentadas por Chugui et al. [9] e Folsom; Bondurant [10].

As vantagens da triangulação citadas por Fan; Tsai [11] são a facilidade de operação e a rapidez de cálculo.

1.2.2.2 Interferometria

A recuperação de forma por interferometria ocorre através da propagação de ondas por uma fonte luminosa. Após passarem num divisor de feixe, num espelho ou em lentes, as ondas ficam com comprimentos e fases diferentes até o anteparo de reflexão. A junção dos feixes de diferentes fases gera as franjas de interferência que carregam as informações de forma do objeto.

Este método apresenta restrição com relação ao tipo de superfície do objeto, já que a medição é possível quando a reflexão da luz no objeto estiver numa mesma direção. Este tipo de reflexão é conhecido como especular e é característico de superfícies polidas, como os espelhos. As peças estampadas, no entanto, apresentam superfície microscopicamente irregular, refletindo raios de luz em várias direções na chamada reflexão difusa e impedindo a medição por interferometria.

A figura 1-14 esquematiza os tipos de reflexão de luz.

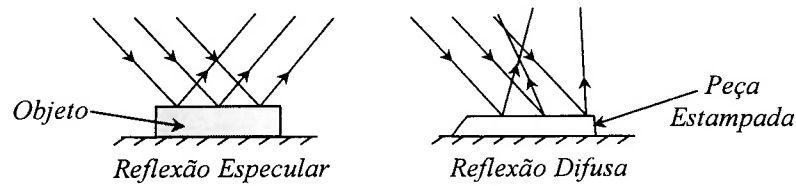


Figura 1-14 – Tipos de reflexão de luz.

A interferometria possui vastas áreas de aplicações. Conforme Tome; Stahl [12], as áreas de maior expansão são: topologia de micro-superfície, testes não destrutivos e contorno de macro-superfícies.

1.2.2.3 Holografia

Em holografia, grava-se num filme fotográfico a estampa de interferência produzida a partir de um feixe de referência e da onda de luz emitida pelo objeto. A intensidade em qualquer ponto na estampa de interferência depende da fase e da amplitude de onda do objeto, registrados na gravação. A iluminação do holograma com a referência de onda original permite que um observador posicionado atrás do holograma enxergue uma imagem 3D do objeto. A figura 1-15 mostra as fases de gravação e reconstrução por holografia.

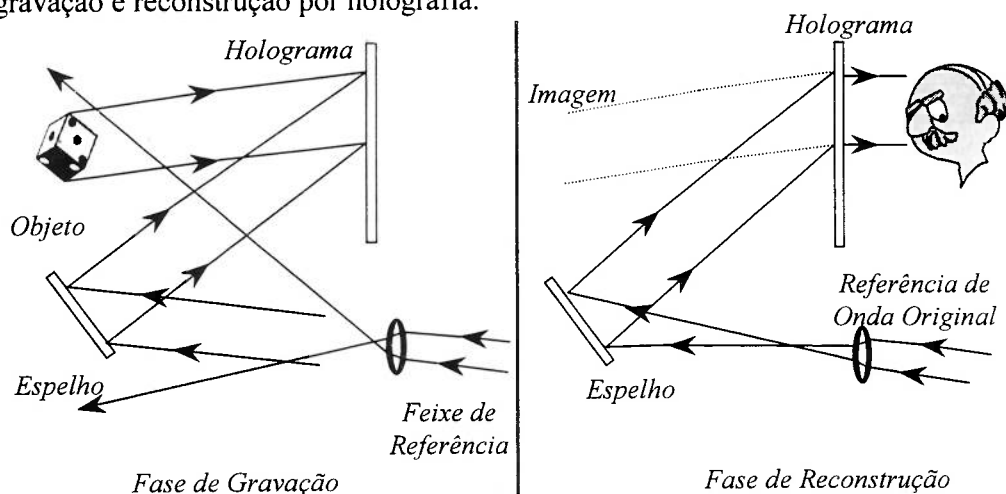


Figura 1-15 – Representação das fases de gravação e reconstrução em holografia.

A holografia também é uma importante técnica utilizada na medição das variações de grandezas físicas tais como deslocamentos, tensões, densidades e vibração, como apresentado por Grant; Brown [13] na análise de pneus da indústria automotiva. Além de verificar os possíveis defeitos do pneu, o método permite medir as propriedades do material, como tensão e deformação.

O uso de um sistema holográfico exige excessivo tempo de montagem dos equipamentos, além de conhecimentos específicos da técnica para a recuperação de forma.

1.2.2.4 Processamento de Imagens

Através da obtenção de imagens de um determinado objeto a partir da utilização de uma câmera de vídeo CCD, por exemplo, sob condições de posição conhecidas, bem como a análise destas imagens, seguida da extração de informação da posição do objeto, faz-se a recuperação de forma por processamento de imagens.

Pode-se subdividir este método em dois grupos distintos: métodos estáticos e métodos dinâmicos.

1.2.2.4.1 Métodos Estáticos

Este subgrupo do método de processamento de imagens difere do método dinâmico devido à grandeza tempo. Neste método, não são necessárias informações de tempo para a obtenção de forma.

A fotometria estéreo é um exemplo de recuperação de forma por processamento de imagens estático. A partir de diferentes imagens de um objeto, com diferentes direções de iluminação e direção de visualização constante, é possível obter a forma do objeto. As propriedades de orientação e reflexão de superfície e a

distribuição da luz são determinantes neste método. Um método de fotometria estéreo é utilizado com sucesso por Kim; Hong; Choi [14].

Outra maneira de recuperação de forma estática de processamento de imagens é a visão estéreo. Neste caso, múltiplas câmeras em posições diferentes obtêm distintas imagens do mesmo objeto. A partir destas imagens e da posição das câmeras, calcula-se as posições dos pontos na superfície dos objetos. A visão estéreo é utilizada por Kosmopoulos; Varvarigou [15] na inspeção de qualidade das folgas entre as partes de automóveis.

1.2.2.4.2 Métodos Dinâmicos

Diferentemente do método estático, no método dinâmico são necessárias informações temporais para a recuperação de forma do objeto. A partir destas informações, obtêm-se maiores detalhes da superfície do objeto do que em métodos estáticos, como concavidades ou convexidades.

A rotação do objeto é um tipo de método dinâmico, em que se utilizam uma câmera fixa para a captura de imagens do objeto e uma mesa giratória na qual o objeto estará localizado. A partir do controle da rotação do objeto e da análise da trajetória dos pontos ao longo do tempo nas imagens do objeto, é feita a reconstrução do objeto. A figura 1-16 mostra o esquema da obtenção de forma por rotação do objeto.

Este método é empregado por Zheng; Fukagawa; Abe [16] na recuperação de forma de objetos com regiões côncavas.

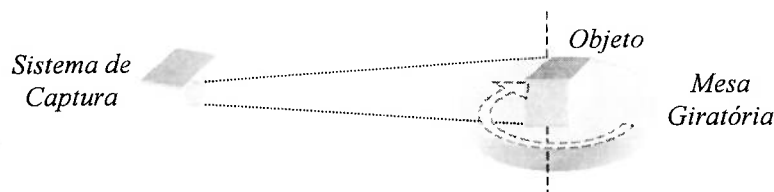


Figura 1-16 – Representação esquemática do método de rotação do objeto.

O mesmo princípio acontece com a movimentação da câmera. Aqui o objeto é posicionado num referencial fixo, deslocando-se a câmera lateralmente. Desta vez, tem-se o controle do deslocamento da câmera para a análise da trajetória dos pontos nas imagens do objeto na reconstrução. Fox [17] mostra uma alternativa para a movimentação lateral de câmera com o uso de espelhos giratórios.

Os métodos de processamento de imagens não são portáteis. Os equipamentos demandam tempo de montagem e exigem conhecimentos específicos de operação. Além disso, existem restrições de tamanho ou forma dos objetos que sofrerão a recuperação de forma.

1.2.2.5 Tomografia

A tomografia é caracterizada pela passagem de raios-X num objeto e a gravação da intensidade destes raios num detector. Como resultado, é obtida, por meio de uma imagem, a representação do nível de absorção de cada porção do feixe de raios-X ao longo de seu caminho no objeto. Esta imagem equivale a uma seção no objeto. Recupera-se a forma em tomografia através aplicação de raios-X em diferentes ângulos de projeção, e assim tem-se a formação de várias seções ao longo do objeto. A figura 1-17 esquematiza o princípio da técnica de tomografia.

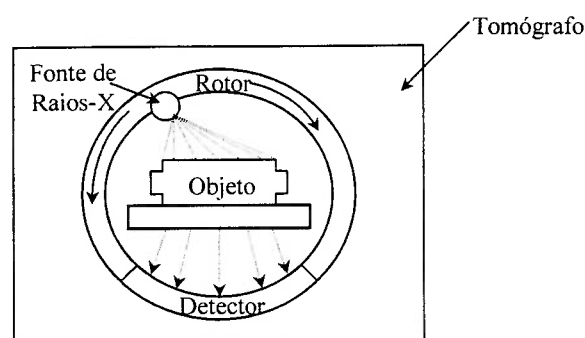


Figura 1-17 – Esquema do princípio da técnica de tomografia.

Apesar da maior aplicação de tomografia ocorrer na área médica, existe um crescente uso desta técnica no setor industrial. O maior uso desta técnica está na

medição das partes internas de um determinado componente. A recuperação de forma de um coletor de exaustão do motor de um automóvel por tomografia é feita por Dropps [18]. A tomografia é utilizada por Gupta; Isaacson [19] para a verificação de falhas na fundição de um bloco de motor e em um pneu de automóvel.

Este método apresenta precisão típica de 0,025 mm. O tomógrafo, equipamento utilizado para a recuperação de forma em tomografia, possui limites para os tamanhos de objetos, não é portátil e exige treinamento específico de operação. A necessidade de projeção e gravação de raios-X em vários ângulos para a reconstrução do objeto demanda tempo excessivo. O tempo de medição de um coletor de exaustão do motor de um automóvel é de 6 a 9 horas [18]. Além disso, o custo de um tomógrafo não pode ser desconsiderado, pois situa-se em torno de 5 a 10 milhões de dólares.

1.2.2.6 Franjas de Moiré

Neste método, uma fonte de luz cria sombras na superfície de um objeto após atravessar uma grade, e um sistema óptico capta a imagem do objeto através da mesma grade. A imagem coletada representa as curvas de nível da superfície do objeto.

A figura 1-18 representa de maneira esquemática o método das franjas de Moiré.

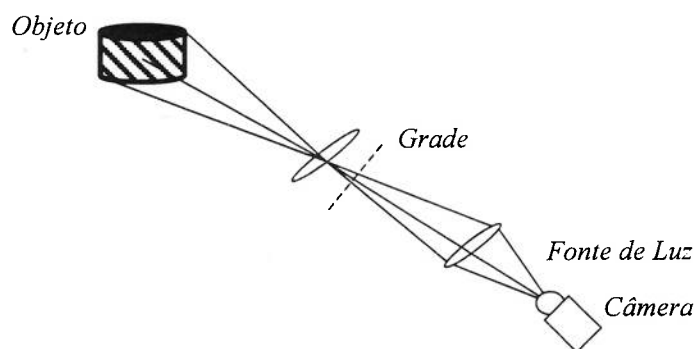


Figura 1-18 – Representação esquemática do método das Franjas de Moiré.

Segundo Whitehouse [20], a palavra Moiré vem da indústria têxtil francesa e seu significado está associado à onda ou aparência de água. As franjas de Moiré têm sido usadas na indústria de “silk” para controle de qualidade.

Este método tem sido bastante utilizado na medição e controle de superfícies nas indústrias, com os seguintes tipos de aplicação:

- Análise de tensões para a determinação de deformações da superfície;
- Determinação do contorno do objeto ou a sua comparação com o estado inicial;
- Medição de distorção fora de plano de uma superfície em relação ao seu estado inicial.

As vantagens deste método identificadas por Neto [21] são: a praticidade na aquisição de dados, a flexibilidade de medição, a possibilidade de uso estático ou dinâmico, a medição de superfícies planas ou curvas, a ausência da necessidade de proteção ou de ambiente especial para o equipamento, a ausência de necessidade do uso de sensores e a inexistência de contato com a superfície de medição.

Wykes; Morshedizadeh [22] verificam que quanto menor e mais plano for o objeto, maior será a precisão do sistema com a utilização deste método. Uma das razões pela renovação do interesse nas técnicas de Moiré citadas por Gäsvisk [23] é o aumento da capacidade e a diminuição nos custos de processamento digital de imagem. Kim; Choi; Oh [24] apresentam um sistema de recuperação baseado na interferometria de Moiré.

Apesar das vantagens deste método identificadas acima, um dos equipamentos utilizados na recuperação de forma por Moiré torna a implementação inviável. Este equipamento, que permite a projeção e ao mesmo tempo a captura de imagens com a projeção sobre um objeto, é complexo e de difícil acesso no mercado.

1.2.2.7 Luz Estruturada

O princípio básico do método luz estruturada consiste na projeção de um feixe com geometria conhecida numa superfície de trabalho. Esta geometria de projeção pode ser de pontos, listras ou grades. A entidade projetada é então modulada pela profundidade da superfície do objeto. Contendo informações das características do objeto, a projeção é capturada por uma câmera (CCD ou fotográfica) e processada em seguida. A correspondência entre os valores de intensidade ou cor, localização do ponto capturado (coordenadas de pixel) e a modulação sofrida pela projeção no objeto são responsáveis pela localização das coordenadas num sistema 3D.

Como mostra a figura 1-19, o objetivo do método por luz estruturada é obter as características 3D do objeto a partir da projeção 2D distorcida.

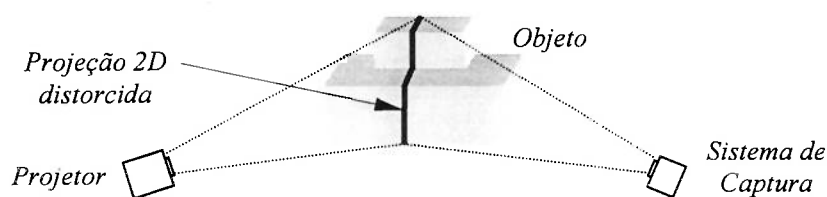


Figura 1-19 - Representação esquemática do método de luz estruturada.

As diferentes maneiras de projeção com luz estruturada surgiram na busca da aprimoração do método. Assim, as primeiras formas de projeção se davam através de único ponto no objeto por raio laser. A captura de várias cenas com um único ponto de projeção levam, no entanto, considerável tempo de processamento das informações.

Surgiu então o padrão de planos verticais de luz, que permite ganhar tempo de processamento [25]. Estes planos, gerados por laser, criam listras verticais de luz no objeto, permitindo que o rastreamento vertical não seja necessário.

Apesar da evolução com os planos de luz, a captura e o processamento de dados eram feitos individualmente para cada plano numa única cena. Para se ganhar mais tempo, as soluções encontradas foram as projeções de múltiplas listras ou

planos de luz e o processamento conjunto das informações. Entretanto, um ponto qualquer na imagem da câmera corresponderia a inúmeras listras ou planos de luz projetados.

O sistema composto de dois projetores e uma câmera procurou eliminar o problema da correspondência e ganhar tempo de processamento, a partir da determinação de um conjunto de soluções possíveis e de um volume de trabalho do sistema [26].

Pesquisas mais recentes com o uso da luz estruturada buscam eliminar o problema da correspondência através da codificação das estampas de projeção. As principais codificações encontradas na literatura para as projeções são: binária, temporal, intensidade, cor e estampa randômica.

A codificação binária trabalha com a projeção de geometrias específicas marcadas por uma informação de intensidade dupla: claro ou escuro. A partir de posições arbitrárias no plano de projeção é possível transmitir uma única listra à câmera [27]. A figura 1-20 mostra um exemplo deste tipo de projeção.

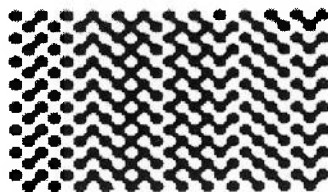


Figura 1-20 – Estampa de projeção com codificação binária.

Na codificação temporal, o espaço a ser medido é subdividido em outros subespaços. Atribui-se a cada subespaço uma característica de intensidade de luz diferente que varia com o decorrer do tempo. Deste modo, o número de estampas está relacionado com o tempo de medição. Valkenburg; McIvor [28] obtêm bons resultados com o uso desta técnica e a consideração de distorção nas lentes e projetor. A codificação temporal também permite o recurso da estampa auto-referenciada [29]. A cada estampa diferente, é possível criar “n!” subespaços, através da mudança na seqüência das estampas ao longo do tempo. Desta forma, um número maior de códigos é possível de ser gerado a partir de uma menor quantidade de estampas.

As estampas de projeção com a variação de intensidade luminosa são utilizadas na codificação por intensidade. A fonte de luz poderá ser laser ou comum. Neste caso, as propriedades de reflexão da superfície de trabalho estão diretamente ligadas à precisão de medição. Miyasaka; Araki [30] apresenta o uso desta técnica com o uso de luz monocromática, visto que não há proporcionalidade entre a intensidade de reflexão e a intensidade incidente da luz para o uso com níveis de cinza. A figura 1-21 ilustra um tipo de estampa de projeção com codificação por intensidade luminosa.



Figura 1-21 – Estampa de projeção com codificação por intensidade luminosa.

A codificação por cor associa um determinado padrão de cor às características 3D do objeto. O uso de seis tipos diferentes de cores no sistema HSI para faixas verticais e horizontais é utilizado por Salvi; Batlle; Mouaddib [31] para a codificação.

A utilização de estampa randômica na codificação segue a mesma idéia, alternando-se, porém as estampas para a criação dos códigos [32]. Maruyama; Abe [33] empregam a projeção de listras cortadas randomicamente para permitir a facilidade de “matching” na reconstrução e a flexibilidade nos posicionamentos de câmera e projetor.

A busca de tempos menores na aquisição por luz estruturada tem conduzido a automação deste método. Lee; Park [34] apresentam a geração automática de caminhos de projeção de listras por laser e aquisição por um único dispositivo, diminuindo o tempo de aquisição de dados.

O uso de luz estruturada em componentes manufaturados havia sido verificado como tema de trabalhos anteriores. Destaca-se a ampla utilização na identificação de defeitos em linhas de produção. Como não são necessários a aquisição e o processamento de grande volume de informações, restringindo-se a

região de aquisição numa área suscetível ao defeito, a identificação fora do padrão torna-se direta.

Chen; Zheng [35] utilizam a luz estruturada num sistema com duas câmeras presas a robôs e um projetor. Várias imagens são obtidas para recuperar a forma de uma placa metálica lisa de 11,65 cm x 7,35 cm e uma leve inclinação de 10°. Foram obtidos resultados com erros de profundidade de 2 mm.

Liu; Sawada; Sakamoto [36] verificam pequenas deformações em pequenas peças metálicas estampadas através da luz estruturada. Resultados expressivos foram obtidos com o uso da projeção de listras, codificação por níveis de cinza e aquisições de imagens antes e após a deformação de peças de pequenas dimensões e de geometria simples como: cilindros e quadrados.

Por sua vez, Niel; Kölp; Burgstaller [37] empregam o método em pauta para a recuperação de forma para peças metálicas forjadas. O sistema é composto de um robô, responsável pela fixação do projetor programável, e duas câmeras fotográficas para a captura. Várias imagens são obtidas pelas câmeras a partir de posições diferentes.

Entretanto, não foram verificados trabalhos disponíveis que objetivam a recuperação de forma em peças estampadas de geometria complexa com o uso da luz estruturada.

O método de luz estruturada é um caso particular da técnica de triangulação, apresentada acima. Como já colocado, a rapidez e a facilidade de operação são as grandes vantagens desta técnica. A medição através da distorção do padrão projetado sobre o objeto torna o método flexível, permitindo que ele seja aplicável aos diferentes tamanhos e formas de um componente estampado. Os recursos para a implementação deste método permitem automatizar o sistema, com ganhos cada vez maiores de tempo e praticidade de operação. Além disso, o uso de apenas um sistema de projeção e outro de aquisição de dados pode garantir a simplicidade de equipamentos, permitindo praticar a portabilidade do sistema.

1.3 Particularidades dos Componentes Estampados

Existem algumas características típicas dos componentes estampados que devem ser consideradas na recuperação de forma. Estas características podem contribuir na escolha do método para os componentes estampados.

Um componente estampado pode ter dimensões variadas. Existem componentes grandes, como os painéis laterais que têm o comprimento e a altura aproximados de um automóvel (3250 x 1200 mm), assim como componentes pequenos, como a tampa de um tanque de combustível (150 mm de diâmetro). No que se refere à profundidade de estampagem, trabalha-se também com variações. Citam-se os exemplos do repuxo profundo de um assoalho traseiro, região de colocação do “step” do veículo, com profundidade de 350 mm e as pequenas formas de até 5 mm de profundidade de alguns componentes estruturais. A espessura de um componente estampado varia entre 0,5 e 1,75 mm. No entanto, estas características não conferem peso elevado aos componentes, permitindo o manuseio sem a necessidade de equipamentos especiais. O uso da recuperação de forma deve ser aplicado a esta diferente gama de componentes.

Um ponto a ser considerado na medição de componentes estampados é a simetria de informações. A medição de apenas uma das faces do componente estampado é suficiente para a medição, dado que a face oposta do componente possui a mesma geometria. Caso a simetria não fosse verificada, a necessidade de medição na face oposta do componente seria iminente. Este ponto é relevante quando o fator tempo é considerado. A figura 1-22 mostra as duas faces de um componente estampado, ilustrando a simetria de informações.

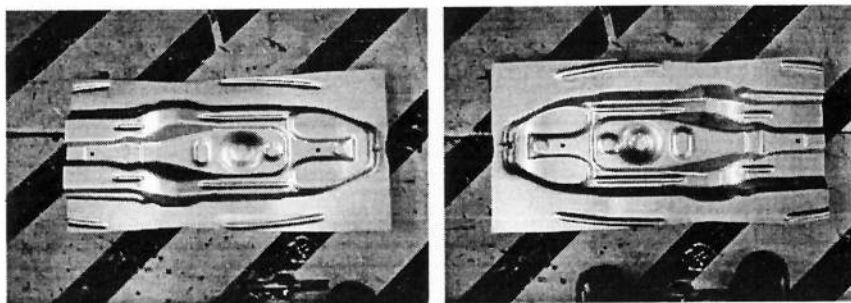


Figura 1-22 – Simetria de informações em componente estampado.

Quanto à forma, um componente estampado pode apresentar superfície rica em forma ou suave. A definição da forma está ligada a função do componente, estrutural ou estética. Os componentes de função estrutural apresentam nervuras e diferentes formas, em geral compõe a parte interna e não são visíveis no automóvel. Como exemplo de componentes ricos em forma do automóvel, podem ser citadas as longarinas, colunas, assoalho e reforços. Já os componentes de função estética têm a superfície suave e relevo bem combinado. Eles formam a chamada casca, transmitindo o estilo do automóvel e sendo, portanto, totalmente visíveis. As portas e laterais externas, pára-lamas e teto são exemplos de componentes de forma suave do automóvel.

A figura 1-23 mostra as formas de um componente estrutural e um estético.

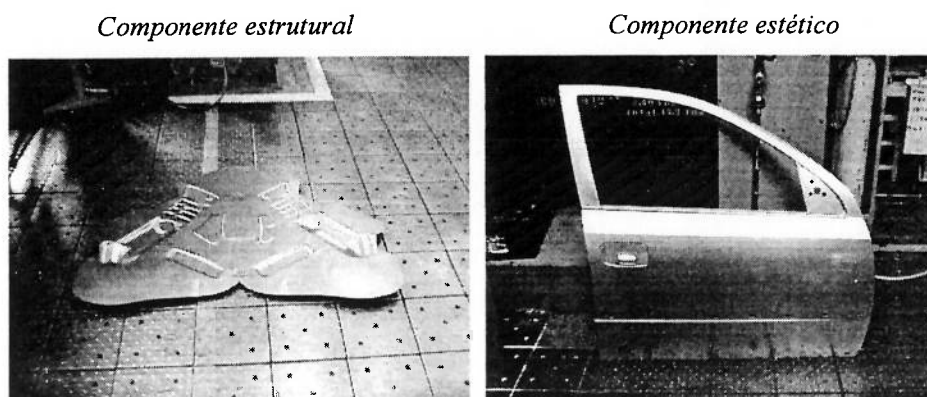


Figura 1-23 – Formas de componentes estrutural e estético.

No que diz respeito às modificações nos complementos, sabe-se que a qualidade visual exigida de um componente estético é muito maior do que a de um componente estrutural. A razão disso é a maior exposição do componente no veículo. Desta forma, é compreensível que haja maiores modificações nos complementos dos componentes estéticos do que dos estruturais.

A composição em aço dos componentes estampados dá a liberdade a alguns eventos importantes. O manuseio inadequado pode levar a defeitos nos componentes, como amassado ou ruptura. Isto sugere que o risco de defeitos num componente é maior tanto quanto é maior o manuseio ou o transporte à medição.

Como consequência do processo de estampagem e também da característica do material estampável, os componentes com perfis na forma de “U” estão sujeitos ao fenômeno do “springback” ou retorno elástico. Neste fenômeno o componente adquire uma forma não adequada, caracterizada pela abertura das abas do componente, o que compromete a qualidade do produto. Este tem sido um grande desafio para as ferramentarias do mundo todo, já que não há uma solução, mas apenas maneiras de minimizar o fenômeno. Uma delas é a medição no componente após a estampagem, da abertura da aba. Determinada a abertura, é feita a compensação numa operação de estampagem posterior, levando o componente à sua forma esperada. Esta medição é feita atualmente com o uso de uma MMC e, como já citado, é um processo lento. A figura 1-24 mostra um componente estampado isento de defeitos ao lado de outro sob o efeito do retorno elástico.

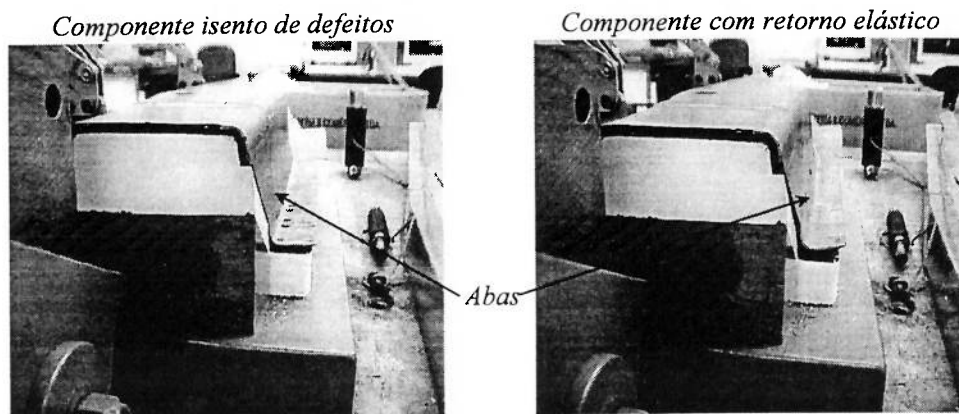


Figura 1-24 – Fenômeno do retorno elástico de um componente.

Ao mesmo tempo as características magnéticas do aço permitem a localização do componente com o uso de imã. Isto pode garantir o posicionamento do componente para a medição num referencial fixo.

Os componentes estampados podem ser considerados meios opacos e homogêneos, já que não permitem a propagação de luz através deles e possuem as mesmas propriedades físicas (densidade, pressão e temperatura) em toda a sua extensão. Devido às irregularidades de superfície (não polimento), a luz sofre reflexão difusa sobre um componente estampado. Estas características devem ser consideradas no uso de métodos ópticos para a recuperação de forma.

Os atuais sistemas de qualidade vigentes (ISO14000), bem como as normas internas de segurança, impedem o uso de óleos lubrificantes no processo de estampagem na General Motors do Brasil. Devido às condições de limpeza de ferramental e dos blanques de estampagem, o componente estampado não precisará passar por lavagens para sofrer a recuperação de forma.

1.4 Síntese Inicial

Com o intuito único de posicionar o leitor quanto às informações colocadas até este momento, apresenta-se nesta seção um resumo inicial.

Identificou-se a necessidade da recuperação de forma em peças estampadas a partir do processo de fabricação destas peças na General Motors do Brasil. A medição de uma peça estampada física e sua comparação com o modelo digital inicial tornam possível a fabricação de novas peças estampadas similares de maneira mais eficiente e com melhor qualidade.

A seguir, foram investigados os possíveis métodos de recuperação de forma. Tomou-se o cuidado de analisar a adequação dos métodos para a aplicação em peças estampadas.

Foram levantadas algumas particularidades relevantes quanto às peças estampadas. Estas particularidades, em conjunto com as análises dos possíveis métodos investigados e das condições de medição, irão definir o método para a recuperação de forma de peças estampadas.

Desta forma, a escolha do método para a recuperação de forma de peças estampadas deve considerar:

- O tipo de informações a serem obtidas. Conforme já citado em 1.1, a necessidade da recuperação de forma de peças estampadas está exclusivamente na obtenção da geometria da peça. As grandezas dos fenômenos dinâmicos do processo de estampagem não são relevantes aqui;

- A velocidade de medição. Espera-se recuperar a forma da peça em cerca de 15 minutos;
- O ambiente de trabalho. O sistema deverá recuperar a geometria numa sala de 5 x 12 metros com controle de luminosidade e sem controle de temperatura;
- A acessibilidade dos equipamentos do sistema de recuperação de forma. Os equipamentos do sistema devem ser de fácil acesso no mercado;
- A praticidade de uso. A praticidade de uso do sistema também é relevante, pois operadores não-especialistas deverão poder operá-lo;
- A flexibilidade do sistema. O método adotado também terá de ser flexível, permitindo a recuperação de forma de peças estampadas de variadas formas e passível de ajustes;
- A possibilidade de automatização do sistema. Esta exigência permite executar medições em menos tempo e com maior praticidade;
- Não há a necessidade de excelentes níveis de precisão. Podem ser considerados aceitáveis erros da ordem de milímetros;
- Os resultados disponíveis e o campo de trabalho. Deve-se observar também o desenvolvimento previsto para a realização deste trabalho e, ao mesmo tempo, o campo de pesquisa disponível.

1.5 Objetivo

Diante do que foi discutido, o presente trabalho terá como objetivo: desenvolver um sistema que permita a recuperação de forma de peças estampadas utilizando o método de luz estruturada.

A obtenção das modificações terá de ser eficiente. Isto significa recuperar a forma dos componentes de maneira rápida, em cerca de 15 minutos. A explicação para esta necessidade está na alta relação entre tempo dos ajustes de ferramental e

tempo de desenvolvimento de um novo veículo. O sistema de recuperação de forma precisará poupar o tempo de ajuste de ferramental, não demandando, portanto, mais tempo no desenvolvimento de um novo veículo.

A medição deverá ser adequada ao ambiente de trabalho. A relevância do quesito tempo, aliada à segurança das informações de medição, exige que a aplicação do método seja feita no mesmo local aonde são praticadas as modificações dos complementos, na ferramentaria. Assim, logo após a estampagem, a medição será feita próxima à prensa. Neste local não estão disponíveis controles de temperatura, mas sim os de luminosidade.

Levando-se em conta que os espaços físicos estão cada vez mais ocupados, revela-se também aqui a necessidade de o sistema ser portátil. Esta característica permite que seja praticada a recuperação de forma em diferentes linhas de prensas ou até mesmo em diferentes plantas de fabricação.

A execução da medição não será feita por um especialista, o que exige um sistema de medição prático e ágil. Ao mesmo tempo, o sistema deve prever a automação do processo de medição, buscando cada vez menos tempo de medição e mais praticidade de execução.

As diferentes formas e tamanhos de um componente estampado requerem um sistema flexível. Os componentes estampados podem ter dimensões próximas ao comprimento de um automóvel, como é o caso dos painéis laterais, ou dimensões menores, como, por exemplo, a maioria dos componentes estruturais do automóvel. Quanto à forma, existem componentes de superfície suave que caracterizam a maior parte dos painéis visíveis do automóvel, ou componentes com superfície rica em forma, o que distingue os painéis internos do automóvel. A recuperação de forma precisa ser aplicada nestes diferentes tipos de componentes da mesma maneira.

No que se refere à precisão da medição, erros da ordem de milímetros são aceitáveis. Isto se deve à ordem de grandeza das modificações dos complementos, que ficam entre 5 e 50 mm em altura, não alcançando os décimos ou centésimos de milímetro. Assim não se faz necessário um sistema de alta precisão.

Existe atualmente um equipamento disponível capaz de obter as modificações dos complementos na General Motors do Brasil. Este equipamento é a máquina de medição por coordenadas (MMC), caracterizada na seção 1.2.1.2 deste trabalho. Entretanto, o equipamento não atende aos requisitos acima expostos.

Foi verificado também um equipamento que efetua medições em três dimensões no Instituto de Tecnologia Brasil-Alemanha (ITBA) na Escola Politécnica da Universidade de São Paulo. Apesar dos resultados das medições serem praticamente instantâneos, este equipamento não é portátil e apresenta restrições quanto ao tamanho dos objetos em análise. Não foi identificada a possibilidade de análise visual a partir de uma reconstrução em três dimensões de um determinado objeto. Além disso, alguns conhecimentos do idioma alemão se fazem necessários para a operação do aplicativo de medição. Estas características inviabilizaram a utilização deste equipamento no presente trabalho, já que não se encaixam no objetivo descrito acima.

1.6 Metodologia

Este trabalho foi dividido em cinco capítulos a fim de atingir o objetivo anteriormente proposto.

O capítulo 1 mostra a necessidade da recuperação de forma, os métodos de recuperação de forma pesquisados, a decisão por um método para a aplicação em peças estampadas, o objetivo e as inovações deste trabalho.

No capítulo 2 será apresentada a modelagem do sistema. Da mesma maneira que Nishida [38], o equacionamento do sistema utilizará um plano de projeção fixo em relação ao projetor e a simplificação da triangulação baseada em transformação geométrica de perspectiva, levando-se em conta alguns elementos do contexto de processamento de imagens.

O método de recuperação de forma será inicialmente implementado no capítulo 3 em objetos de isopor de dimensões conhecidas a fim de verificar o

comportamento do sistema. Neste capítulo, será apresentada também a análise dos resultados obtidos. Será empregado o recurso da simulação digital com a finalidade de separar o erro do modelo do erro do sistema. Como limitações do sistema, serão identificados: 1. uma tendência de falta de precisão na metade da imagem capturada do lado oposto ao da câmera; 2. o surgimento das áreas de oclusão devido ao aumento de altura do objeto e 3. a diminuição do ângulo em relação ao plano de projeção. Estas limitações levarão a elaboração de uma nova abordagem para o método de luz estruturada.

Esta nova abordagem será apresentada no capítulo 4. Ela consistirá em adquirir informações do objeto a partir de um posicionamento de câmera oposto em relação ao projetor, selecionar as informações mais precisas e combiná-las com as obtidas pelo posicionamento preliminar de câmera. Um processo numérico com conceitos de algoritmo genético será descrito ao final do capítulo como método de calibração do sistema.

A recuperação de forma de peças estampadas será ensaiada no capítulo 5 com a aplicação de todo o desenvolvimento. Os resultados serão analisados quanto à reconstrução dos pontos através da comparação com a medição feita por uma máquina de medição por coordenadas e, no que se refere à forma, através de comparação visual.

1.7 Inovações

Esta dissertação inova na aplicação do método porque não foram encontrados trabalhos anteriores com o objetivo de recuperar a forma de peças estampadas de geometria complexa com luz estruturada. Existem algumas iniciativas disponíveis, como a apresentada por Chen; Zheng [35] com câmeras fixadas em um robô e um projetor para recuperar a forma de uma peça metálica lisa e de pequena dimensão (11,65 x 7,35 cm). Liu; Sawada; Sakamoto [36] verificam as pequenas deformações de peças estampadas de geometria simples e de pequenas dimensões antes e após a estampagem. Já Niel; Kölp; Burgstaller [37] trabalham com peças

metálicas forjadas, um projetor programável fixado a um robô e imagens capturadas de vários pontos.

Simultaneamente, a nova abordagem desenvolvida neste trabalho também é uma inovação. A solução adotada de capturar imagens do objeto a partir de outro posicionamento de câmera, selecionar as informações mais precisas e combiná-las com as previamente obtidas introduz uma novidade no método de recuperação de forma adotado.

CAPÍTULO II – DESCRIÇÃO DO MÉTODO DE RECUPERAÇÃO DE FORMA EMPREGADO

Este capítulo tem por objetivo descrever o método adotado para a recuperação de forma. Na modelagem do sistema, optou-se pela abordagem utilizada por Nishida [38], devido aos bons resultados por ele obtidos.

2.1 Composição do Sistema

O sistema de recuperação de forma por luz estruturada é composto fisicamente por: projetor de luz, o dispositivo de captura de imagens e o equipamento de processamento e armazenamento das informações.

2.2 Modelagem do Sistema

A fim de modelar matematicamente o sistema, foram assumidos um posicionamento genérico para os equipamentos e um modelo de câmera. A seguir foram equacionados os pontos necessários para a triangulação.

2.2.1 Posicionamento Genérico dos Equipamentos

Adotou-se um posicionamento genérico para os equipamentos. Este posicionamento foi definido a partir de condições de contorno do sistema que visam simplificar o equacionamento. Estas condições de contorno são:

- 1- Posicionamento de projetor e câmera no mesmo plano horizontal;
- 2- Localização perpendicular de eixo central do projetor com relação ao plano de projeção.

O posicionamento genérico dos equipamentos para a recuperação de forma por luz estruturada é mostrado em vista de topo na figura 2-1.

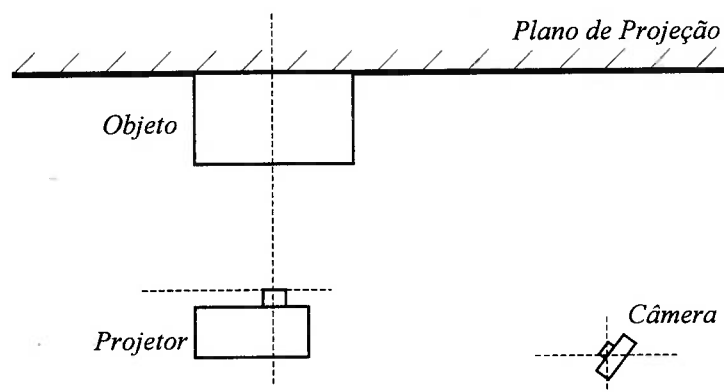


Figura 2-1 – Posicionamento genérico dos equipamentos.

Satisfeitas as condições de contorno, é possível ajustar a posição exata dos equipamentos, como será visto nos capítulos 3, 4 e 5 de acordo com o ensaio praticado.

2.2.2 Modelo de Câmera

O modelo de câmera adotado foi o de “pinhole”, devido à simplicidade e boa correlação com as câmeras reais. Neste modelo, a projeção está definida pela reta

que, a partir da abertura infinitesimal do diafragma, passa pelo ponto do plano da imagem. Todos os pontos desta reta serão projetados no mesmo ponto da imagem.

A figura 2-2 mostra a representação esquemática do modelo de “pinhole”.

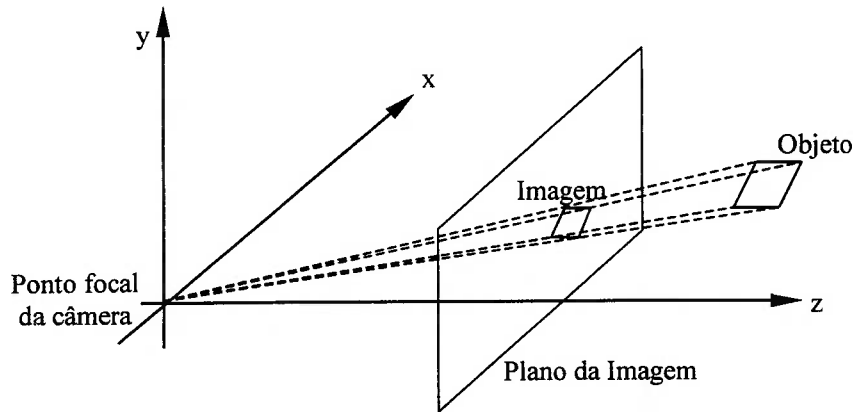


Figura 2-2 – Representação do modelo de câmera.

2.2.3 Equacionamento do Sistema

A partir das definições do modelo de câmera e das condições de contorno no posicionamento genérico dos equipamentos, segue-se o equacionamento do sistema. A figura 2-3 mostra o esquema óptico geométrico do sistema. O ponto focal do projetor é representado pela letra F , e da câmera pela letra G . O plano de referência é representado pelo plano π e o plano de formação da imagem é representado pelo plano π' . O ponto P é um ponto qualquer do objeto.

Para equacionar o sistema, foi utilizada a geometria analítica. As equações literais foram resolvidas manualmente e checadas com o uso do aplicativo Mathcad. O desenvolvimento das equações pode ser obtido no Anexo B. O objetivo final é encontrar as coordenadas do ponto P em função das coordenadas do projetor e da câmera.

Definiu-se por nomenclatura a representação das coordenadas cartesianas dos pontos pelas letras x , y e z . O subscrito destas letras indica o ponto ao qual a coordenada se refere. Da mesma maneira equacionou-se retas e planos com a representação de x , y e z com os subscritos de geometria correspondente. Todas as coordenadas estão referenciadas à origem do sistema, definida como sendo o canto superior esquerdo do padrão projetado.

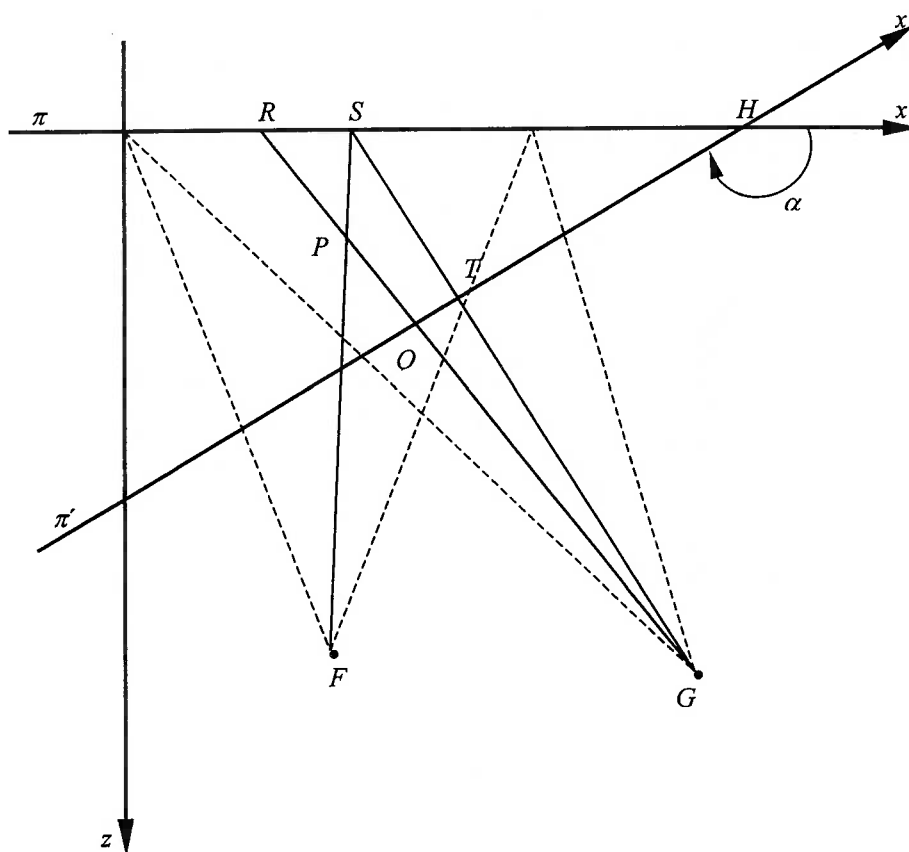


Figura 2-3 – Esquema óptico geométrico do sistema.

Pressupõe-se para o equacionamento o conhecimento do ponto focal do projetor F , o ponto focal da câmera G e o plano π' definido pelo ponto H e o ângulo α . Na prática, o conhecimento destas posições leva ao conhecimento da posição relativa do projetor e da câmera em relação ao plano de referência. Estes valores podem ser obtidos através de medições no local do ensaio. As informações sobre o plano π' são para determinar a direção em que a câmera será apontada. A direção do projetor é conhecida devido à condição de contorno apresentada em 2.2.1.

Inicia-se o equacionamento pela determinação do ponto S . Este ponto pertence ao plano de referência e receberá a projeção do mesmo padrão do ponto P . A diferença vista pela câmera entre as distâncias horizontais entre pontos P e S será responsável pela reconstrução do objeto em profundidade. O ponto S é determinado pela interseção da reta definida pelos pontos F e P e o plano π . As equações da reta FP e do plano π são:

$$\begin{cases} x_{FP} = x_F + (x_P - x_F) \cdot \lambda \\ y_{FP} = y_F + (y_P - y_F) \cdot \lambda \\ z_{FP} = z_F + (z_P - z_F) \cdot \lambda \end{cases} \quad (2.1)$$

$$z_{\pi} = 0 \quad (2.2)$$

Igualando z_{FP} a z_{π} , calculando λ e substituindo em 2.1, tem-se o ponto S com a seguinte equação:

$$\begin{cases} x_S = \frac{x_F \cdot z_P - z_F \cdot x_P}{z_P - z_F} \\ y_S = \frac{y_F \cdot z_P - z_F \cdot y_P}{z_P - z_F} \\ z_S = 0 \end{cases} \quad (2.3)$$

Segue-se então o cálculo do ponto Q , projeção do ponto P sobre o plano π' . O conjunto de todos os pontos Q forma a imagem capturada pela câmera. A interseção da reta formada pelos pontos G e P com o plano π' define o ponto Q .

As equações da reta GP e do plano π' são:

$$\begin{cases} x_{GP} = x_G + (x_P - x_G) \cdot \lambda \\ y_{GP} = y_G + (y_P - y_G) \cdot \lambda \\ z_{GP} = z_G + (z_P - z_G) \cdot \lambda \end{cases} \quad (2.4)$$

$$x_{\pi'} \cdot \text{sen } \alpha - z_{\pi'} \cdot \cos \alpha - x_H \cdot \text{sen } \alpha = 0 \quad (2.5)$$

Determina-se o ponto Q igualando-se x_{GP} com x_π e calculando-se λ . As equações do ponto Q são então:

(2.6)

$$\begin{cases} x_Q = \frac{x_G \cdot x_H \cdot \text{sen}\alpha - x_P \cdot x_H \cdot \text{sen}\alpha + x_G \cdot z_P \cdot \text{cos}\alpha - z_G \cdot x_P \cdot \text{cos}\alpha}{z_P \cdot \text{cos}\alpha - z_G \cdot \text{cos}\alpha - x_P \cdot \text{sen}\alpha + x_G \cdot \text{sen}\alpha} \\ y_Q = \frac{y_G \cdot z_P \cdot \text{cos}\alpha + y_P \cdot x_G \cdot \text{sen}\alpha - y_P \cdot z_G \cdot \text{cos}\alpha + y_G \cdot x_H \cdot \text{sen}\alpha - y_G \cdot x_P \cdot \text{sen}\alpha - y_P \cdot x_H \cdot \text{sen}\alpha}{z_P \cdot \text{cos}\alpha - z_G \cdot \text{cos}\alpha - x_P \cdot \text{sen}\alpha + x_G \cdot \text{sen}\alpha} \\ z_Q = \frac{x_G \cdot z_P \cdot \text{sen}\alpha - z_P \cdot x_H \cdot \text{sen}\alpha - z_G \cdot x_P \cdot \text{sen}\alpha + z_G \cdot x_H \cdot \text{sen}\alpha}{z_P \cdot \text{cos}\alpha - z_G \cdot \text{cos}\alpha - x_P \cdot \text{sen}\alpha + x_G \cdot \text{sen}\alpha} \end{cases}$$

O ponto T , projeção do ponto S sobre o plano π' , é calculado da mesma maneira. O conjunto de todos os pontos T representa a imagem do plano de referência capturada pela câmera. Determina-se T pela interseção da reta GS com o plano π' . A reta GS tem como equação:

$$\begin{cases} x_{GS} = x_G + \left[\frac{x_F \cdot z_P - z_F \cdot x_P}{z_P - z_F} - x_G \right] \cdot \lambda \\ y_{GS} = y_G + \left[\frac{y_F \cdot z_P - z_F \cdot y_P}{z_P - z_F} - y_G \right] \cdot \lambda \\ z_{GS} = z_G - z_G \cdot \lambda \end{cases} \quad (2.7)$$

Utilizando-se as equações 2.5 e 2.7, obtém-se a equação do ponto T :

(2.8)

$$\begin{aligned} x_T &= \frac{z_F \cdot x_P \cdot x_H \cdot \text{sen}\alpha - z_F \cdot x_G \cdot x_H \cdot \text{sen}\alpha + z_F \cdot x_P \cdot z_G \cdot \text{cos}\alpha - z_P \cdot x_F \cdot z_G \cdot \text{cos}\alpha}{z_P \cdot x_G \cdot \text{sen}\alpha - x_F \cdot z_P \cdot \text{sen}\alpha - z_F \cdot x_G \cdot \text{sen}\alpha + z_F \cdot x_P \cdot \text{sen}\alpha + z_F \cdot z_G \cdot \text{cos}\alpha - z_P \cdot z_G \cdot \text{cos}\alpha} + \\ &\quad \frac{z_P \cdot x_G \cdot x_H \cdot \text{sen}\alpha - z_P \cdot x_F \cdot x_H \cdot \text{sen}\alpha}{z_P \cdot x_G \cdot \text{sen}\alpha - x_F \cdot z_P \cdot \text{sen}\alpha - z_F \cdot x_G \cdot \text{sen}\alpha + z_F \cdot x_P \cdot \text{sen}\alpha + z_F \cdot z_G \cdot \text{cos}\alpha - z_P \cdot z_G \cdot \text{cos}\alpha} \\ y_T &= \frac{z_F \cdot x_P \cdot y_G \cdot \text{sen}\alpha - x_F \cdot z_P \cdot y_G \cdot \text{sen}\alpha + z_F \cdot y_P \cdot z_G \cdot \text{cos}\alpha - z_F \cdot y_G \cdot x_H \cdot \text{sen}\alpha +}{z_P \cdot x_G \cdot \text{sen}\alpha - x_F \cdot z_P \cdot \text{sen}\alpha - z_F \cdot x_G \cdot \text{sen}\alpha + z_F \cdot x_P \cdot \text{sen}\alpha + z_F \cdot z_G \cdot \text{cos}\alpha - z_P \cdot z_G \cdot \text{cos}\alpha} + \\ &\quad \frac{z_P \cdot y_G \cdot x_H \cdot \text{sen}\alpha + z_F \cdot y_P \cdot x_H \cdot \text{sen}\alpha - z_P \cdot y_F \cdot x_H \cdot \text{sen}\alpha - z_F \cdot y_P \cdot x_G \cdot \text{sen}\alpha}{z_P \cdot x_G \cdot \text{sen}\alpha - x_F \cdot z_P \cdot \text{sen}\alpha - z_F \cdot x_G \cdot \text{sen}\alpha + z_F \cdot x_P \cdot \text{sen}\alpha + z_F \cdot z_G \cdot \text{cos}\alpha - z_P \cdot z_G \cdot \text{cos}\alpha} + \\ &\quad \frac{z_P \cdot y_F \cdot x_G \cdot \text{sen}\alpha - z_P \cdot y_F \cdot z_G \cdot \text{cos}\alpha}{z_P \cdot x_G \cdot \text{sen}\alpha - x_F \cdot z_P \cdot \text{sen}\alpha - z_F \cdot x_G \cdot \text{sen}\alpha + z_F \cdot x_P \cdot \text{sen}\alpha + z_F \cdot z_G \cdot \text{cos}\alpha - z_P \cdot z_G \cdot \text{cos}\alpha} \\ z_T &= \frac{z_P \cdot z_G \cdot x_H \cdot \text{sen}\alpha - x_F \cdot z_P \cdot z_G \cdot \text{sen}\alpha + z_F \cdot x_P \cdot z_G \cdot \text{sen}\alpha - z_F \cdot z_G \cdot x_H \cdot \text{sen}\alpha}{z_P \cdot x_G \cdot \text{sen}\alpha - x_F \cdot z_P \cdot \text{sen}\alpha - z_F \cdot x_G \cdot \text{sen}\alpha + z_F \cdot x_P \cdot \text{sen}\alpha + z_F \cdot z_G \cdot \text{cos}\alpha - z_P \cdot z_G \cdot \text{cos}\alpha} \end{aligned}$$

As equações dos pontos Q e T foram obtidas em relação ao sistema de coordenadas $Oxyz$ da figura 2.3. Sabendo-se que a imagem capturada pela câmera é formada no plano π' , é necessário expressar as equações (2.6) e (2.8) no sistema de coordenadas $Hx'y'z'$. Utiliza-se assim a matriz de conversão:

$$\begin{bmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix}$$

seguida da translação:

$$\begin{bmatrix} -x_H \cdot \cos \alpha \\ 0 \\ x_H \cdot \sin \alpha \end{bmatrix}$$

Após a conversão, chega-se às equações (2.9) e (2.10), referentes aos pontos Q e T no sistema $Hx'y'z'$.

(2.9)

$$\begin{cases} x'_Q = \frac{z_G \cdot x_H - z_G \cdot x_P + z_P \cdot x_G - z_P \cdot x_H}{z_P \cdot \cos \alpha - z_G \cdot \cos \alpha - x_P \cdot \sin \alpha + x_G \cdot \sin \alpha} \\ y'_Q = \frac{y_G \cdot x_H \cdot \sin \alpha - y_P \cdot x_H \cdot \sin \alpha + y_G \cdot z_P \cdot \cos \alpha - y_P \cdot z_G \cdot \cos \alpha + y_P \cdot x_G \cdot \sin \alpha - y_G \cdot x_P \cdot \sin \alpha}{z_P \cdot \cos \alpha - z_G \cdot \cos \alpha - x_P \cdot \sin \alpha + x_G \cdot \sin \alpha} \end{cases}$$

(2.10)

$$\begin{cases} x'_T = \frac{z_F \cdot x_P \cdot z_G - x_F \cdot z_P \cdot z_G + z_P \cdot z_G \cdot x_H - z_F \cdot z_G \cdot x_H}{z_F \cdot z_G \cdot \cos \alpha - z_P \cdot z_G \cdot \cos \alpha + z_F \cdot x_P \cdot \sin \alpha - x_F \cdot z_P \cdot \sin \alpha + z_P \cdot x_G \cdot \sin \alpha - z_F \cdot x_G \cdot \sin \alpha} \\ y'_T = \frac{z_P \cdot y_F \cdot x_G \cdot \sin \alpha - x_F \cdot z_P \cdot y_G \cdot \sin \alpha - z_P \cdot y_F \cdot x_H \cdot \sin \alpha - z_F \cdot y_G \cdot x_H \cdot \sin \alpha}{z_F \cdot z_G \cdot \cos \alpha - z_P \cdot z_G \cdot \cos \alpha + z_F \cdot x_P \cdot \sin \alpha - x_F \cdot z_P \cdot \sin \alpha + z_P \cdot x_G \cdot \sin \alpha - z_F \cdot x_G \cdot \sin \alpha} + \\ \frac{z_F \cdot x_P \cdot y_G \cdot \sin \alpha - z_F \cdot y_P \cdot x_G \cdot \sin \alpha + z_F \cdot y_P \cdot z_G \cdot \cos \alpha + z_F \cdot y_P \cdot x_H \cdot \sin \alpha}{z_F \cdot z_G \cdot \cos \alpha - z_P \cdot z_G \cdot \cos \alpha + z_F \cdot x_P \cdot \sin \alpha - x_F \cdot z_P \cdot \sin \alpha + z_P \cdot x_G \cdot \sin \alpha - z_F \cdot x_G \cdot \sin \alpha} + \\ \frac{z_P \cdot y_G \cdot x_H \cdot \sin \alpha - z_P \cdot y_F \cdot z_G \cdot \cos \alpha}{z_F \cdot z_G \cdot \cos \alpha - z_P \cdot z_G \cdot \cos \alpha + z_F \cdot x_P \cdot \sin \alpha - x_F \cdot z_P \cdot \sin \alpha + z_P \cdot x_G \cdot \sin \alpha - z_F \cdot x_G \cdot \sin \alpha} \end{cases}$$

Foram obtidas, assim as equações necessárias para a triangulação. Através da diferença entre a cena obtida do plano de referência e da cena com o

objeto em relação à câmera, $(x'_Q - x'_T)$, torna-se possível determinar a profundidade do objeto (z_P) pela inversão apropriada da equação. Deste modo, cria-se aqui uma nova variável Δx para representar a diferença entre a projeção do plano de referência e a projeção com o objeto, como mostra a figura 2-4.

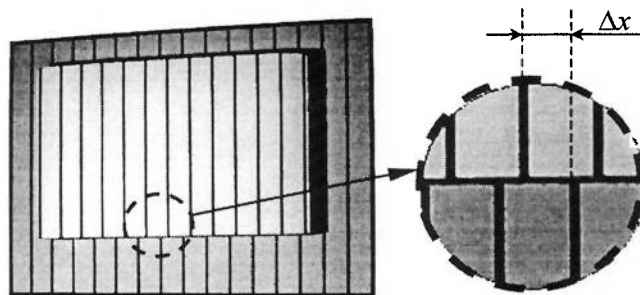


Figura 2-4 – Representação do termo Δx .

Apesar do completo equacionamento do sistema, as equações (2.9) e (2.10) são complexas e possuem a agravante de se referirem a imagens como a mostrada na figura 2-5. Esta imagem revela a presença do efeito de perspectiva, gerando um padrão trapezoidal nas imagens. A dificuldade verificada aqui está no processamento, já que a leitura deveria ser feita ao longo de linhas inclinadas, desfavoráveis à consideração matricial de imagens digitais.

A figura 2-5 ilustra a dificuldade de processamento ao longo de linhas inclinadas, como as mostradas em tracejado.

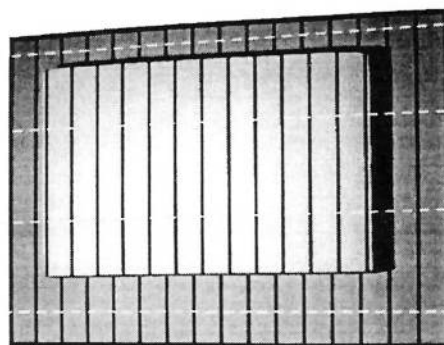


Figura 2-5 – Imagem com aspecto trapezoidal.

Visando superar esta dificuldade, desenvolveu-se uma simplificação da triangulação através de uma transformação de perspectiva. O desenvolvimento completo desta simplificação é apresentado na próxima seção.

Na simplificação, os pontos Q e T serão convertidos nos pontos R e S . O ponto S já foi determinado, e assim a transformação fará apenas o caminho inverso da equação (2.10). Calcula-se R pela interseção entre a reta GP e o plano π . As equações da reta GP e do plano π foram definidas em (2.4) e (2.2), respectivamente.

Igualando-se z_{GP} a z_{π} calculando-se λ e substituindo-se em (2.4), o ponto R fica definido por:

$$\begin{cases} x_R = \frac{z_P \cdot x_G - z_G \cdot x_P}{z_P - z_G} \\ y_R = \frac{z_P \cdot y_G - z_G \cdot y_P}{z_P - z_G} \\ z_R = 0 \end{cases} \quad (2.11)$$

Verifica-se que as equações de R e S são muito mais simples do que as equações de Q e T . Além disso, nas equações (2.11) e (2.3) não se fazem presentes os termos referentes ao plano π' , permitindo que as imagens a serem processadas fiquem no tamanho retangular.

É feito então o cálculo do termo Δx , correspondente à diferença entre x_R e x_S :

$$\Delta x = \frac{z_G \cdot x_R \cdot z_P - x_F \cdot z_G \cdot z_P - z_F \cdot x_R \cdot z_P + z_F \cdot x_G \cdot z_P}{z_P \cdot z_G - z_F \cdot z_G} \quad (2.12)$$

A fim de se determinar as equações do ponto P , as equações (2.11) e (2.12) são trabalhadas, obtendo-se as equações (2.13), que serão utilizadas para a reconstrução 3D a partir das imagens obtidas pela câmera.

$$\begin{cases} z_P = \frac{\Delta x \cdot z_F \cdot z_G}{z_G \cdot x_F - x_G \cdot z_F + \Delta x \cdot z_G + z_F \cdot x_R - z_G \cdot x_R} \\ x_P = \frac{z_G \cdot x_R - z_P \cdot x_R + z_P \cdot x_G}{z_G} \\ y_P = \frac{z_G \cdot y_R - z_P \cdot y_R + z_P \cdot y_G}{z_G} \end{cases} \quad (2.13)$$

2.2.4 Simplificação da Triangulação

A fim de se obter a simplificação das equações da triangulação para evitar tempo excessivo de processamento, foi utilizada uma transformação de perspectiva. Nesta transformação, a projeção no plano π' é levada para o plano π , a partir do ponto focal G da câmera.

Conforme as condições de contorno da seção 2.2.1, posiciona-se o projetor a fim de se obter uma projeção retangular e de que a câmera obtenha uma imagem trapezoidal. As figuras 2-6 e 2-7 esquematizam, respectivamente, a cena capturada pela câmera no plano π' e a mesma cena com projeção no plano π . Através da correspondência entre os pontos destes esquemas, a transformação de perspectiva é elaborada.

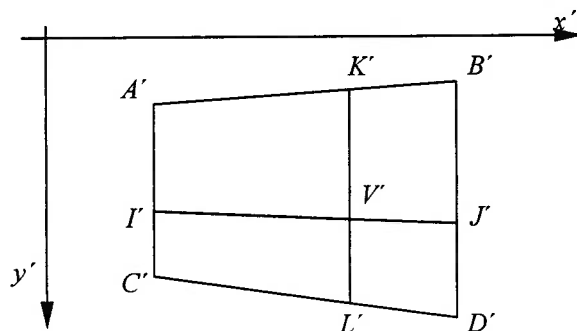


Figura 2-6 – Projeção das bordas da imagem no plano π' .

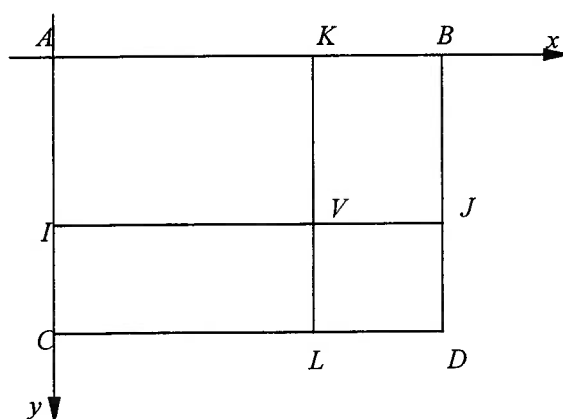


Figura 2-7 – Projeção das bordas da imagem transformada no plano π .

Os pontos A , B , C e D da figura 2-7 são obtidos através da medição da altura e largura do padrão projetado no posicionamento do projetor. O ponto V é um ponto qualquer na projeção, cuja posição na figura 2-6 será identificada. I , J , K e L são pontos auxiliares, correspondentes às projeções de V sobre os lados do retângulo.

A tabela 2-1 mostra as coordenadas dos pontos da figura 2-7. A própria geometria do retângulo facilita a determinação das coordenadas, baseando-se apenas nos pontos A , D e V .

	A	B	C	D	I	J	K	L	V
X	x_A	x_D	x_A	x_D	x_A	x_D	x_V	x_V	x_V
Y	y_A	y_A	y_D	y_D	y_V	y_V	y_A	y_D	y_V

Tabela 2-1 – Coordenadas geométricas dos pontos da figura 2-7.

O próximo passo é encontrar as relações entre estes pontos e os pontos do plano π' . Para isso, utilizou-se a equação (2.9), substituindo-se os valores das coordenadas de cada ponto da figura 2-7. Vale lembrar que não está sendo considerada aqui a profundidade da cena. Desta forma, obtém-se:

$$\begin{cases} x'_A = \frac{z_G \cdot x_A - z_G \cdot x_H}{z_G \cdot \cos \alpha + x_A \cdot \sin \alpha - x_G \cdot \sin \alpha} \\ y'_A = \frac{y_A \cdot x_H \cdot \sin \alpha - y_G \cdot x_H \cdot \sin \alpha + y_A \cdot z_G \cdot \cos \alpha - y_A \cdot x_G \cdot \sin \alpha + y_G \cdot x_A \cdot \sin \alpha}{z_G \cdot \cos \alpha + x_A \cdot \sin \alpha - x_G \cdot \sin \alpha} \end{cases}$$

$$\begin{cases} x'_B = \frac{z_G \cdot x_D - z_G \cdot x_H}{z_G \cdot \cos \alpha + x_D \cdot \sin \alpha - x_G \cdot \sin \alpha} \\ y'_B = \frac{y_A \cdot x_H \cdot \sin \alpha - y_G \cdot x_H \cdot \sin \alpha + y_A \cdot z_G \cdot \cos \alpha - y_A \cdot x_G \cdot \sin \alpha + y_G \cdot x_D \cdot \sin \alpha}{z_G \cdot \cos \alpha + x_D \cdot \sin \alpha - x_G \cdot \sin \alpha} \end{cases}$$

$$\begin{cases} x'_C = \frac{z_G \cdot x_A - z_G \cdot x_H}{z_G \cdot \cos \alpha + x_A \cdot \sin \alpha - x_G \cdot \sin \alpha} \\ y'_C = \frac{y_D \cdot x_H \cdot \sin \alpha - y_G \cdot x_H \cdot \sin \alpha + y_D \cdot z_G \cdot \cos \alpha - y_D \cdot x_G \cdot \sin \alpha + y_G \cdot x_A \cdot \sin \alpha}{z_G \cdot \cos \alpha + x_A \cdot \sin \alpha - x_G \cdot \sin \alpha} \end{cases}$$

$$\begin{cases} x'_D = \frac{z_G \cdot x_D - z_G \cdot x_H}{z_G \cdot \cos \alpha + x_D \cdot \sin \alpha - x_G \cdot \sin \alpha} \\ y'_D = \frac{y_D \cdot x_H \cdot \sin \alpha - y_G \cdot x_H \cdot \sin \alpha + y_D \cdot z_G \cdot \cos \alpha - y_D \cdot x_G \cdot \sin \alpha + y_G \cdot x_D \cdot \sin \alpha}{z_G \cdot \cos \alpha + x_D \cdot \sin \alpha - x_G \cdot \sin \alpha} \end{cases}$$

$$\begin{cases} x'_I = \frac{z_G \cdot x_A - z_G \cdot x_H}{z_G \cdot \cos \alpha + x_A \cdot \sin \alpha - x_G \cdot \sin \alpha} \\ y'_I = \frac{y_V \cdot x_H \cdot \sin \alpha - y_G \cdot x_H \cdot \sin \alpha + y_V \cdot z_G \cdot \cos \alpha - y_V \cdot x_G \cdot \sin \alpha + y_G \cdot x_A \cdot \sin \alpha}{z_G \cdot \cos \alpha + x_A \cdot \sin \alpha - x_G \cdot \sin \alpha} \end{cases}$$

$$\begin{cases} x'_J = \frac{z_G \cdot x_D - z_G \cdot x_H}{z_G \cdot \cos \alpha + x_D \cdot \sin \alpha - x_G \cdot \sin \alpha} \\ y'_J = \frac{y_V \cdot x_H \cdot \sin \alpha - y_G \cdot x_H \cdot \sin \alpha + y_V \cdot z_G \cdot \cos \alpha - y_V \cdot x_G \cdot \sin \alpha + y_G \cdot x_D \cdot \sin \alpha}{z_G \cdot \cos \alpha + x_D \cdot \sin \alpha - x_G \cdot \sin \alpha} \end{cases}$$

$$\begin{cases} x'_K = \frac{z_G \cdot x_V - z_G \cdot x_H}{z_G \cdot \cos \alpha + x_V \cdot \sin \alpha - x_G \cdot \sin \alpha} \\ y'_K = \frac{y_A \cdot x_H \cdot \sin \alpha - y_G \cdot x_H \cdot \sin \alpha + y_A \cdot z_G \cdot \cos \alpha - y_A \cdot x_G \cdot \sin \alpha + y_G \cdot x_V \cdot \sin \alpha}{z_G \cdot \cos \alpha + x_V \cdot \sin \alpha - x_G \cdot \sin \alpha} \end{cases}$$

$$\begin{cases} x'_L = \frac{z_G \cdot x_V - z_G \cdot x_H}{z_G \cdot \cos \alpha + x_V \cdot \sin \alpha - x_G \cdot \sin \alpha} \\ y'_L = \frac{y_D \cdot x_H \cdot \sin \alpha - y_G \cdot x_H \cdot \sin \alpha + y_D \cdot z_G \cdot \cos \alpha - y_D \cdot x_G \cdot \sin \alpha + y_G \cdot x_V \cdot \sin \alpha}{z_G \cdot \cos \alpha + x_V \cdot \sin \alpha - x_G \cdot \sin \alpha} \end{cases}$$

$$\begin{cases} x'_V = \frac{z_G \cdot x_V - z_G \cdot x_H}{z_G \cdot \cos \alpha + x_V \cdot \sin \alpha - x_G \cdot \sin \alpha} \\ y'_V = \frac{y_V \cdot x_H \cdot \sin \alpha - y_G \cdot x_H \cdot \sin \alpha + y_V \cdot z_G \cdot \cos \alpha - y_V \cdot x_G \cdot \sin \alpha + y_G \cdot x_V \cdot \sin \alpha}{z_G \cdot \cos \alpha + x_V \cdot \sin \alpha - x_G \cdot \sin \alpha} \end{cases}$$

A partir das coordenadas dos pontos, é possível agora determinar as distâncias entre os pontos no plano π' . Utilizou-se a equação (2.14) para o cálculo de distância entre pontos.

$$|P_1P_2| = \sqrt{(x_{P1} - x_{P2})^2 + (y_{P1} - y_{P2})^2} \quad (2.14)$$

Devido à insuficiência de espaço, foram omitidos aqui os resultados do cálculo da distância entre os pontos. Os cálculos foram feitos por meio do aplicativo Mathcad.

Observando-se a figura 2-6, é possível estabelecer algumas relações de distância por semelhança de triângulos, já que se tem $A'C'$, $B'D'$ e $K'L'$ paralelos.

Assim:

$$\frac{|A'I'|}{|I'C'|} = \frac{|B'J'|}{|J'D'|} = \frac{|K'V'|}{|V'L'|} = \frac{y_V - y_A}{y_D - y_V} \quad (2.15)$$

No entanto, foi possível verificar também que existe um fator de correção dado pelos lados desiguais da figura 2-6. Desta forma, tem-se:

$$\frac{|C'L'|}{|L'D'|} \cdot \frac{|B'D'|}{|A'C'|} = \frac{|I'V'|}{|V'J'|} \cdot \frac{|B'D'|}{|A'C'|} = \frac{|A'K'|}{|K'B'|} \cdot \frac{|B'D'|}{|A'C'|} = \frac{x_V - x_A}{x_D - x_V} \quad (2.16)$$

Definem-se as razões r_h e r_v a fim de simplificar o equacionamento:

$$r_h = \frac{x_V - x_A}{x_D - x_A} \quad (2.17)$$

$$r_v = \frac{y_V - y_A}{y_D - y_A} \quad (2.18)$$

No plano π' , as razões r_h e r_v são obtidas através das relações (2.15) e (2.16):

$$r_h' = \frac{r_h \cdot |A'C'|}{r_h \cdot |A'C'| - r_h \cdot |B'D'| - |B'D'|}$$

$$r_v' = r_v$$

As coordenadas de I' e J' são calculadas pelas equações que seguem para se obter o ponto V' :

$$\begin{cases} x_I' = x_A' + x_C' \cdot r_v - x_A' \cdot r_v \\ y_I' = y_A' + y_C' \cdot r_v - y_A' \cdot r_v \end{cases} \quad (2.19)$$

$$\begin{cases} x_J' = x_B' + x_D' \cdot r_v - x_B' \cdot r_v \\ y_J' = y_B' + y_D' \cdot r_v - y_B' \cdot r_v \end{cases} \quad (2.20)$$

Finalmente, obtém-se V' pelas equações:

$$\begin{cases} x_V' = x_I' + x_J' \cdot r_h - x_I' \cdot r_h \\ y_V' = y_I' + y_J' \cdot r_h - y_I' \cdot r_h \end{cases} \quad (2.21)$$

Enfim, o procedimento adotado que simplifica as equações da triangulação e transforma a imagem do plano π' para o plano π é:

1- Determinar as coordenadas dos pontos A' , B' , C' e D' da imagem capturada pela câmera (aspecto trapezoidal semelhante à figura 2-6);

2- Definir imagem de mesmo tamanho (largura x altura) do padrão projetado no plano de referência ou em escala;

3- Calcular r_h , r_v , r'_h e r'_v ;

4- Encontrar as coordenadas de I , J e finalmente V .

CAPÍTULO III – IMPLEMENTAÇÃO DO MÉTODO DE RECUPERAÇÃO DE FORMA ATRAVÉS DE ENSAIOS PRELIMINARES

Neste capítulo, o método de recuperação de forma será inicialmente implementado com a finalidade de verificar o comportamento do sistema. Serão aqui descritos os equipamentos utilizados no sistema de recuperação de forma, os objetos a ensaiar, a captura das imagens e cada etapa de tratamento destas imagens para obter os resultados preliminares. A partir da análise destes resultados, serão identificadas as limitações do sistema ao final do capítulo.

3.1 Equipamentos do Sistema

Algumas características esperadas do sistema de recuperação de forma, como portabilidade, praticidade, adequação ao ambiente de trabalho, eficiência e precisão, dependem dos equipamentos utilizados. Dentre estas características, observa-se que as três primeiras são qualitativas, classificando ou não um sistema portátil, prático e adequado. Já as duas últimas são quantitativas, qualificando um sistema como mais ou menos eficiente e preciso, dependendo do equipamento em questão. Assim, um projetor de luz e um dispositivo de captura de imagens de baixa resolução - 176 x 144 pixels - conduziriam a resultados de menor precisão, conforme identificado por Nishida [38]. Da mesma forma, um equipamento de processamento e armazenamento de informações com processador de baixa velocidade, cerca de 100 MHz, demandaria mais tempo na medição. Verificou-se nos equipamentos disponíveis na General Motors do Brasil as características qualitativas esperadas e bons níveis para as características quantitativas citadas acima. Desta forma, foram adotados os equipamentos caracterizados a seguir.

O projetor de luz estruturada é um projetor de multimídia da marca SONY, modelo vpl-cx6 com resolução de 800 x 600 pixels.

A câmera fotográfica digital da marca SONY, modelo DSC-P10, com comprimento focal de 7,9 a 23,7 mm e resolução de 5.000.000 pixels, é o dispositivo de captura de imagens.

O equipamento de processamento e armazenamento das informações é um micro-computador PC com processador Athlon AMD de 2.0 GHz, com 512 Mbytes de memória RAM. A linguagem de programação utilizada para o desenvolvimento do software foi C++ .

Os equipamentos adotados para o sistema de recuperação de forma são visualizados na figura 3-1.

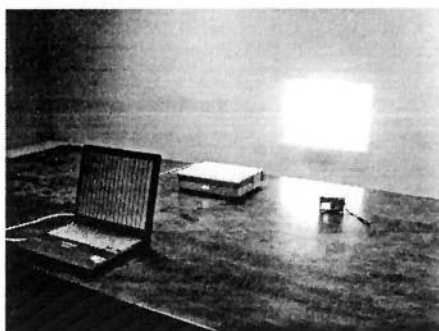


Figura 3-1 – Equipamentos do sistema de recuperação de forma.

3.2 Objetos a Ensaiai

Os objetos adotados para sofrer recuperação de forma inicial foram um paralelepípedo e uma meia esfera. Esta assunção levou em conta a simplicidade de forma destes objetos, visto que a finalidade dos ensaios é uma análise preliminar do sistema.

Os objetos a ensaiar são feitos de isopor devido à disponibilidade destes modelos no mercado e a boa resposta à reflexão de luz verificada.

Quanto às dimensões, decidiu-se ensaiar objetos de tamanhos aproximados a um componente estampado de médio porte. Isto significa um tamanho intermediário entre peças estampadas pequenas, como a tampa de um tanque de combustível, e grandes, como painel lateral de um automóvel.

A figura 3-2 mostra os objetos a ensaiar adotados e suas dimensões em milímetros.

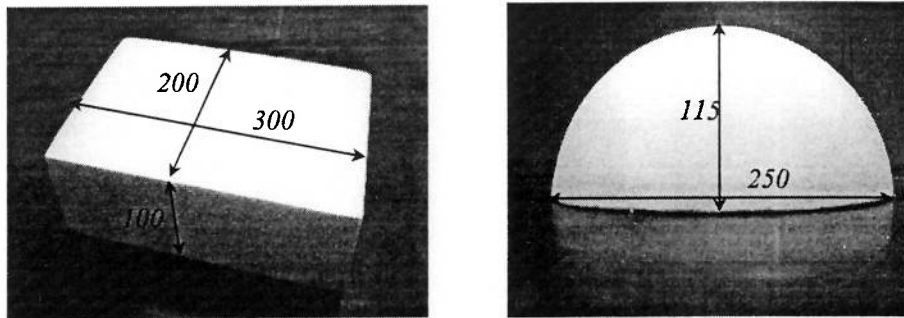


Figura 3-2 – Objetos utilizados para recuperação de forma com dimensões em milímetros.

3.3 Captura das Imagens

Definidos os objetos a ensaiar e os equipamentos do sistema de recuperação de forma, a etapa seguinte de implementação do método é a captura das imagens. Para isso, os equipamentos foram posicionados levando-se em consideração:

- As condições de contorno apresentadas em 2.2.1. Estas condições devem ser respeitadas uma vez que todo o equacionamento do sistema havia sido fundamentado nelas;
- Os objetos a ensaiar. Para a recuperação de forma total é necessário que o padrão de projeção esteja não somente sobre o paralelepípedo e a meia esfera por completo, mas também sobre os arredores destes objetos no plano de projeção π para o cálculo do termo Δx , como mostra a figura 2-4. Isso exige um afastamento adequado do projetor em relação ao plano de projeção;

- A área utilizada para a montagem dos equipamentos.

O posicionamento dos equipamentos para a recuperação de forma por luz estruturada do paralelepípedo e da meia esfera é mostrado em vista de topo na figura 3-3.

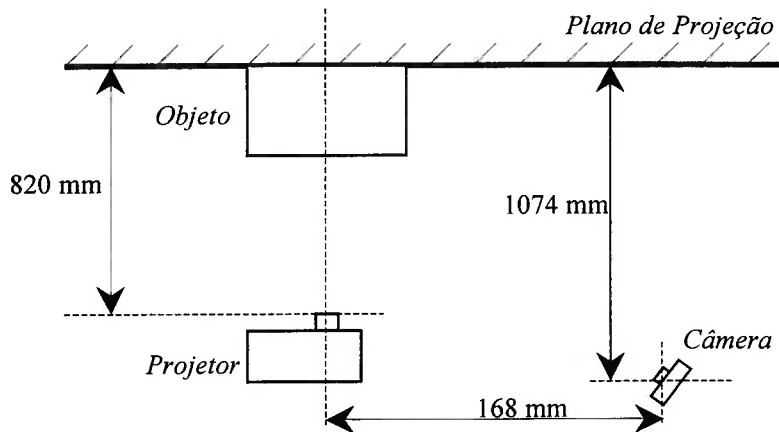


Figura 3-3 – Posicionamento dos equipamentos.

As medidas mostradas na figura 3-3 estão relacionadas aos pontos focais de câmera e projetor, e têm seus cálculos apresentados no Anexo A.

A figura 3-4 mostra as imagens capturadas pela câmera sob o mesmo ângulo (α) de 176° em relação ao plano de projeção para os objetos em questão.

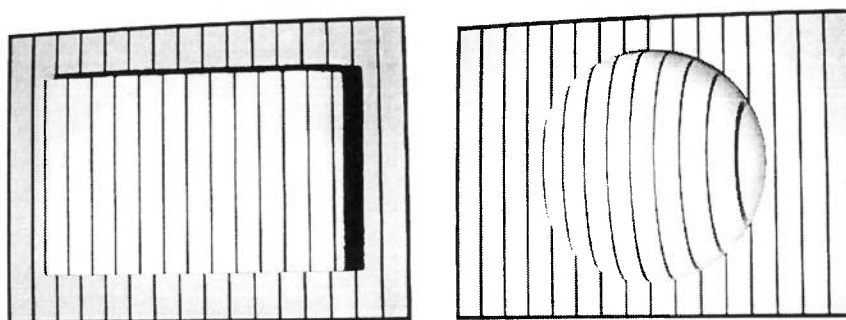


Figura 3-4 – Imagens capturadas pela câmera dos objetos.

3.4 Tratamento das Imagens

A etapa seguinte à captura das imagens na implementação do sistema de recuperação de forma é o tratamento das imagens. Dividiu-se esta etapa seqüencialmente em pré-processamento e processamento das imagens capturadas. A descrição das etapas, bem como os resultados iniciais obtidos, são apresentados abaixo.

3.4.1 Pré-processamento

Esta etapa tem por objetivo preparar as imagens capturadas para o processamento. Isso se faz necessário em razão de haverem sido utilizados uma transformação de perspectiva e um padrão de projeção binário (preto e branco) para aplicar a triangulação no processamento.

A rotina do pré-processamento está esquematizada na figura 3-5.

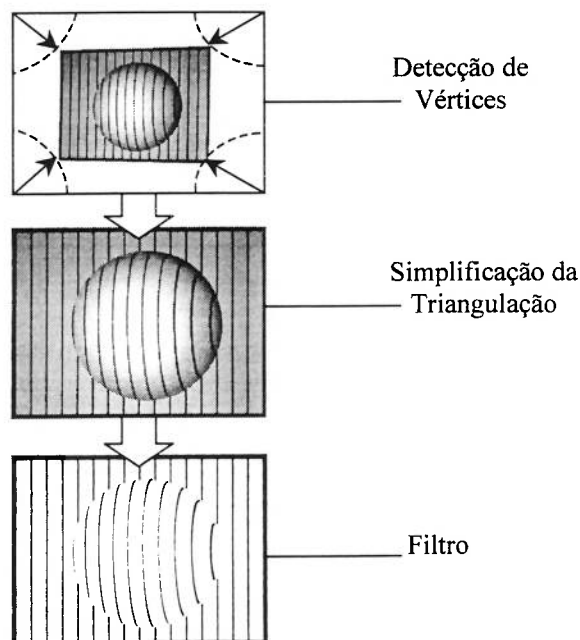


Figura 3-5 – Rotina do pré-processamento.

O uso da margem no padrão de projeção tem como objetivo demarcar a região de trabalho. Esta demarcação é fundamental para a detecção dos vértices, pois atribui aos pontos da cena em análise valores de intensidade conhecidos. Assim, na detecção faz-se a procura pelo pixel de menor intensidade, mais escuro, na menor distância dos vértices da imagem capturada pela câmera.

A partir da determinação das coordenadas dos vértices, torna-se possível aplicar a transformação de perspectiva. A transformação é aplicada na imagem de vértices detectados pela utilização do procedimento explicado na seção 2.2.4.

A figura 3-6 mostra as imagens obtidas após a transformação de perspectiva das imagens capturadas pela câmera.

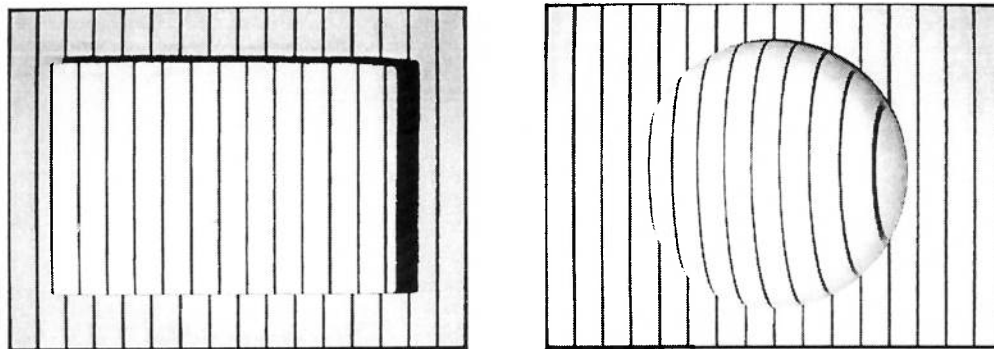


Figura 3-6 – Imagens após a simplificação da triangulação.

A fim de prosseguirmos com o tratamento das imagens, cabe aqui um comentário com relação às sombras dos modelos em análise. As sombras são regiões de baixa intensidade (escuras) localizadas no plano de referência e com forma definida pelo objeto em análise. As sombras são indesejáveis aqui porque podem esconder informações importantes no processamento. Desta maneira, tomou-se o cuidado de remover as sombras no pré-processamento do sistema.

A etapa seguinte é a aplicação do filtro. Até aqui as imagens foram trabalhadas em escala de cinza. O uso do filtro transforma as imagens em escala de cinza para imagens em escala binária (preto e branco), permitindo que a identificação dos pontos na imagem para a triangulação seja feita de maneira única.

A figura 3-7 mostra as imagens dos objetos após a aplicação do filtro.

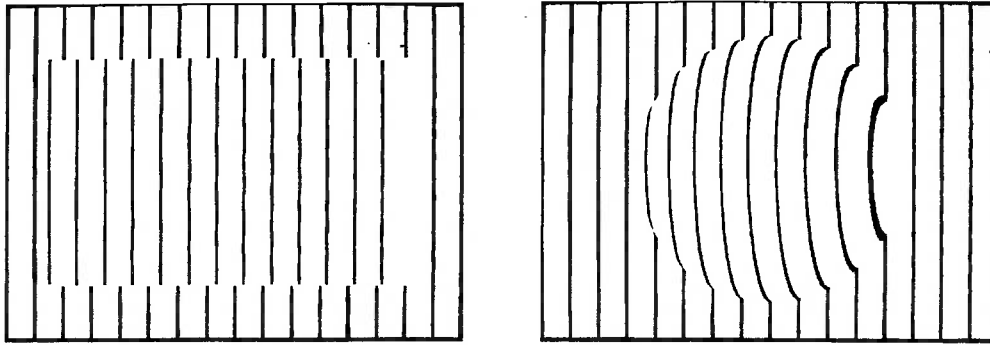


Figura 3-7 – Imagens após aplicação de filtro.

3.4.2 Processamento

Nesta etapa, o sistema transformará as cenas obtidas após o pré-processamento em pontos tridimensionais. Isso é operado através da aplicação das equações da triangulação (2.13), apresentadas aqui novamente para o processamento.

$$\left\{ \begin{array}{l} z_p = \frac{\Delta x \cdot z_F \cdot z_G}{z_G \cdot x_F - x_G \cdot z_F + \Delta x \cdot z_G + z_F \cdot x_R - z_G \cdot x_R} \\ x_p = \frac{z_G \cdot x_R - z_p \cdot x_R + z_p \cdot x_G}{z_G} \\ y_p = \frac{z_G \cdot y_R - z_p \cdot y_R + z_p \cdot y_G}{z_G} \end{array} \right. \quad (2.13)$$

O processamento é pontual, ou seja, para cada ponto lido da cena obtida após o pré-processamento, é gerada uma posição no sistema de três dimensões. Considerando-se que as cenas obtidas são na realidade matrizes de pontos, e que para cada ponto existe uma intensidade (branco ou preto), o processamento é aplicado analisando-se a posição e a intensidade dos pontos.

Deste modo, todos os termos das equações (2.13) são conhecidos neste momento para o processamento. Os termos z_F , x_G e z_G , obtidos por medição no local do ensaio, têm seus valores em pixel mostrados na tabela 3-1.

z_F	x_G	z_G
1490,9	705,4	1952,7

Tabela 3-1 – Valores obtidos para os posicionamentos do projetor e da câmera.

O termo Δx é calculado pela análise da posição dos pontos no plano de referência e na cena que contém o objeto, como mostrado na figura 2.4. Por sua vez, os termos x_R e y_R são obtidos através do próprio sistema de procura de pontos nas cenas em análise. É possível verificar isso por meio de uma análise da figura 2-3 e das cenas que contêm os objetos na figura 3-6. O cálculo de x_F e y_G é feito com base nas dimensões das cenas em análise, devido às condições de contorno assumidas em 2.2.1. Assim, x_F corresponde à metade da largura da cena em análise e y_G corresponde à metade da altura da cena em análise.

A reconstrução dos objetos mostrados na figura 3.2 é vista na figura 3.8.



Figura 3.8 – Reconstrução de paralelepípedo e meia esfera.

3.5 Análise de Resultados

Analisando-se o paralelepípedo reconstruído da figura 3.8, percebe-se a representação da altura, comprimento e largura do paralelepípedo através das faixas em elevação. A mesma afirmação é feita para a meia esfera reconstruída.

Criando-se um corte ao longo do comprimento do paralelepípedo e na posição correspondente à metade de sua largura, obtêm-se as distribuições de alturas mostradas na figura 3-9. É apresentado em linha vermelha, nesta mesma figura, o paralelepípedo real correspondente, adotando-se a escala de 1,82 pixels para cada 1 milímetro.

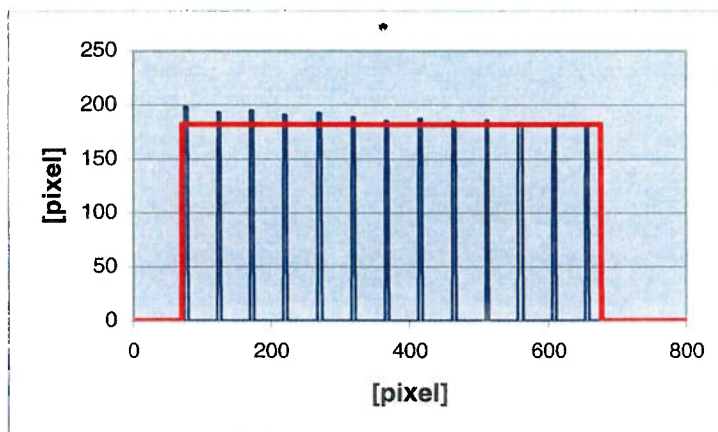


Figura 3-9 – Corte aplicado ao longo do comprimento do paralelepípedo.

A tabela 3-2 mostra o erro médio percentual de altura e os valores em pixels de média de altura e dimensões de base obtidos na seção ao longo do comprimento do paralelepípedo.

<i>Média de Altura</i>	<i>Erro Médio</i>	<i>Dimensões de Base</i>
187,2	2,9 %	525 x 363

Tabela 3-2 – Resultados numéricos obtidos para a seção do paralelepípedo.

3.5.1 Análise de Erro

O gráfico da figura 3-9 representa o conjunto de cada ponto z_P com o x_P correspondente. Dispõe-se de informações para a reconstrução somente na região das listras projetadas sobre o objeto. Assim, as informações de comprimento do paralelepípedo não poderão ser obtidas com precisão pois não há faixas cobrindo os limites de comprimento do paralelepípedo. Quanto às informações de largura, estas apresentaram erro de apenas 1 pixel.

Ainda no corte mostrado na figura 3-9, as mesmas faixas em azul mostram uma tendência de precisão ao longo do paralelepípedo. Lembrando que origem do sistema está localizada no canto superior esquerdo da imagem capturada e a câmera está do lado direito do projetor, verifica-se um aumento gradual na precisão para os pontos capturados próximos à câmera. A figura 3-10 esquematiza esta verificação. Atribui-se esta diferença de precisão como resultado da simplificação da triangulação. Na simplificação, há uma maior migração dos pontos localizados no lado menor do que no lado maior da imagem com aspecto trapezoidal capturada pela câmera. A seção 4 propõe uma solução para esta análise.

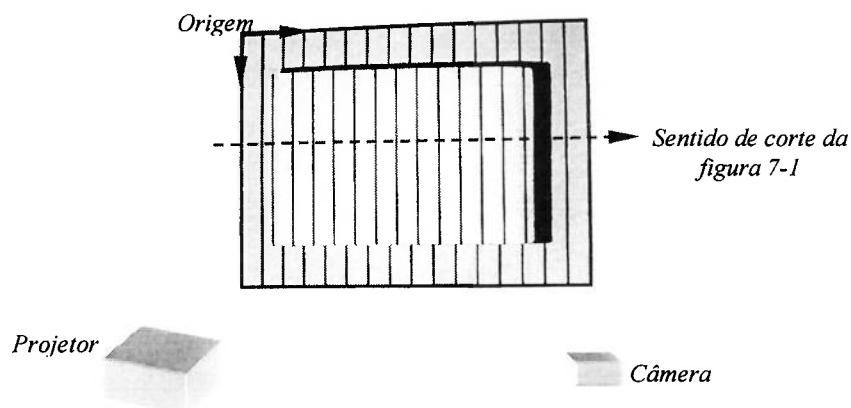


Figura 3-10 – Aumento de precisão para as regiões próximas à câmera.

Observa-se agora o erro médio obtido para a reconstrução em altura. Apesar do sistema seguir as condições de contorno impostas em 2.2.1, das imagens serem capturadas em sistema digital e tratadas através de modelos computacionais, existem parâmetros de entrada que dependem de medições sujeitas a erro. Estes

parâmetros são os termos z_F , x_G e z_G que foram obtidos por medição no local do ensaio. Analisando-se a equação 2.13, é possível verificar o efeito destes termos no cálculo da informação de altura z_P .

3.5.1.1 Simulação

A fim de investigar os erros obtidos nas reconstruções mostradas na figura 3.8, será empregado o recurso da simulação, separando-se o erro devido aos equipamentos do sistema. A grande vantagem deste recurso está na criação de modelos digitais a partir de determinados valores de posicionamento de projetor câmera, assim como o uso destes mesmos valores na reconstrução. Desta forma, um paralelepípedo de dimensões 300x200 mm foi criado na mesma escala utilizada na seção 3.4. A figura 3-11 mostra a imagem do paralelepípedo simulado.

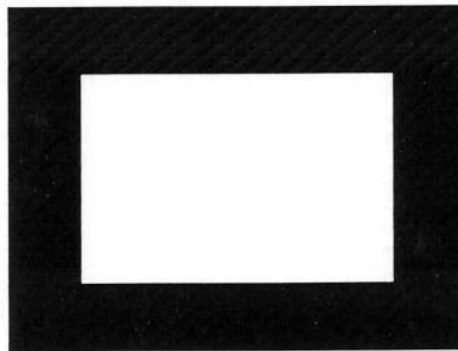


Figura 3-11 – Imagem de paralelepípedo criado por simulação.

A seguir, o padrão de projeção adotado é aplicado sobre o paralelepípedo, simulando o projetor. Isso é feito com o uso das equações (2.3) que relacionam cada ponto do sistema de três dimensões, ponto P , com os respectivos pontos do plano de referência, ponto S , no esquema da figura 2.3. A distância focal do projetor e o posicionamento da câmera adotados neste caso, incluindo o ângulo com o plano de projeção, são os mesmos da seção 3.3.

A figura 3-12 mostra o resultado da projeção do padrão adotado sobre o paralelepípedo já criado.

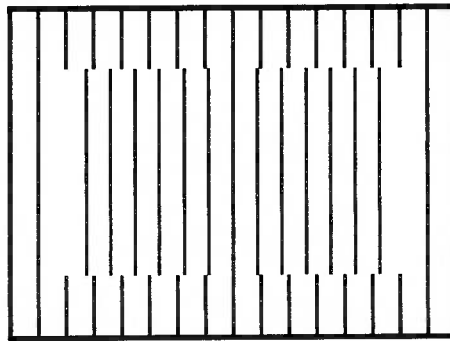


Figura 3-12 – Imagem de simulação de projeção sobre o paralelepípedo.

A simulação de captura da imagem pela câmera é possível através da aplicação das equações (2.9). Assim como a simulação de projeção, calcula-se a localização de cada ponto no plano de formação de imagens da câmera, ponto Q , a partir de cada ponto do sistema de três dimensões, ponto P . A figura 3-13 mostra o resultado da simulação de captura da imagem pela câmera.

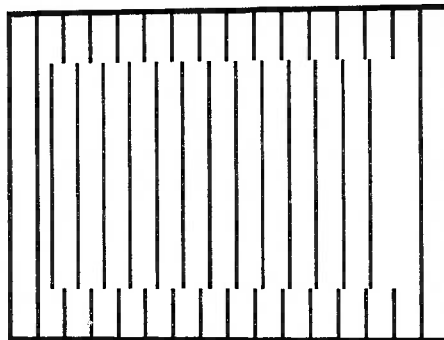


Figura 3-13 – Imagem de simulação de captura da câmera.

Assim como as imagens da figura 3-4, a imagem da figura 3-13 é submetida ao pré-processamento e ao processamento. Neste caso, o erro verificado na reconstrução foi de 0,45%. Considerando-se que na seção 3.5 foi obtido um erro de 2,9% para as mesmas condições, pode-se afirmar que parte do erro deve-se justamente à imprecisão nas medições de localização de projetor e câmera.

3.5.2 Áreas de Oclusão

É sensato observar que existem limites de altura para os objetos que proporcionem reconstrução com os mesmos níveis de precisão obtidos acima. À medida que objetos de maiores alturas são utilizados para as mesmas posições de projetor e câmera, há uma fuga dos pontos projetados sobre o objeto em relação ao plano de referência, permitindo que haja o surgimento de áreas que ficarão escondidas atrás do objeto. Não há como o sistema ler e reconstruir os pontos destas áreas. Devido à ausência de visibilidade, pode-se denominar estas áreas como áreas de oclusão.

A figura 3-14 mostra o aumento de altura de um objeto e o surgimento da área de oclusão demarcada pela linha tracejada.

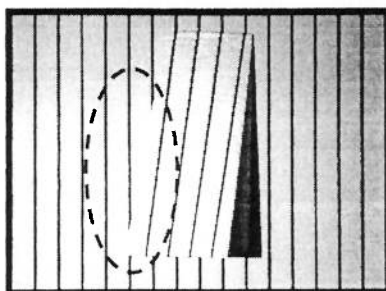


Figura 3-14 – Área de oclusão com aumento de altura do objeto.

Acrescenta-se aqui a relevância deste evento no presente trabalho, dado o fato de que um componente estampado possui formas e alturas variadas.

3.5.3 O Efeito do Ângulo da Câmera

Dentro das condições de contorno assumidas em 2.2.1, pode-se capturar as imagens com variações no ângulo (α) da câmera em relação ao plano de projeção. Ao mesmo tempo, os objetos de trabalho não possuem um padrão de forma, apresentando superfícies com faces de diferentes inclinações.

A captura de imagens num determinado ângulo aliada ao comportamento da superfície do objeto em medição possibilita resultados diferentes de precisão. A captura de imagens com o ângulo α normal à face do objeto de trabalho conduz a resultados de maior precisão. Desta forma, diminui-se o ângulo α para se obter maior precisão em objetos com superfícies inclinadas em relação ao plano de projeção.

No entanto, existe uma relação inversamente proporcional entre o ângulo α e as áreas de oclusão. À medida que se diminui o ângulo α , acontece o aumento das áreas de oclusão. Como já citado acima, o sistema não pode ler estas áreas. O surgimento da área de oclusão devido à diminuição do ângulo α é visualizado na área demarcada pela linha tracejada da figura 3-15, pela comparação da captura a 175° e a 150° da mesma meia esfera mostrada na seção 3.2.

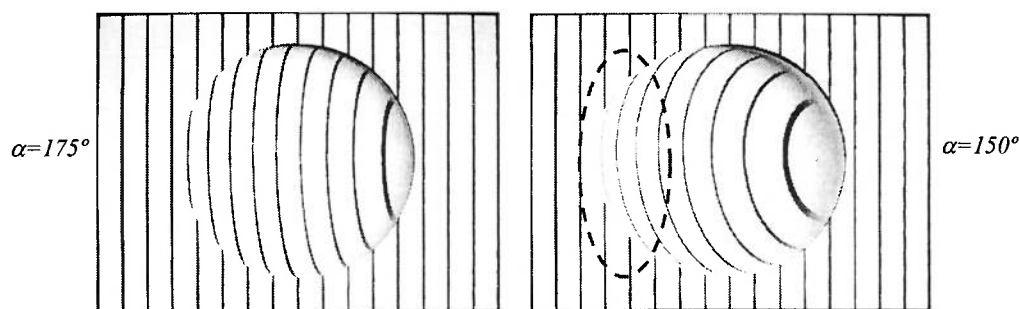


Figura 3-15 – Área de oclusão devido à variação do ângulo α .

3.6 Limitações do Sistema

Conforme exposto acima, foram identificadas as seguintes limitações neste sistema:

- Falta de precisão no lado da imagem capturada oposto ao da câmera;
- Surgimento de áreas de oclusão devido ao aumento de altura do objeto;
- Surgimento de áreas de oclusão devido à variação no ângulo α .

A fim de superar estas limitações, foi elaborada uma nova abordagem para este método.

CAPÍTULO IV – NOVA ABORDAGEM

4.1 Descrição da Nova Abordagem

Esta nova abordagem trata da obtenção de informações sobre o objeto a partir de um outro ponto de aquisição, da seleção das informações mais precisas e da combinação destas informações com aquelas já obtidas pelo método utilizado nos ensaios acima expostos.

Desta forma, devido a melhor precisão obtida no lado da imagem capturado pela câmera e o surgimento de áreas de oclusão no lado oposto ao da câmera, mostrados na análise de resultados, decidiu-se colher dados do objeto com a câmera posicionada também do lado esquerdo do projetor.

Considerando-se as mesmas restrições impostas em 2.2.1 para o posicionamento dos equipamentos, bem como o posicionamento de câmera do lado esquerdo do projetor, foi montado o esquema óptico geométrico do sistema mostrado na figura 4-1 analogamente ao esquema da figura 2-3.

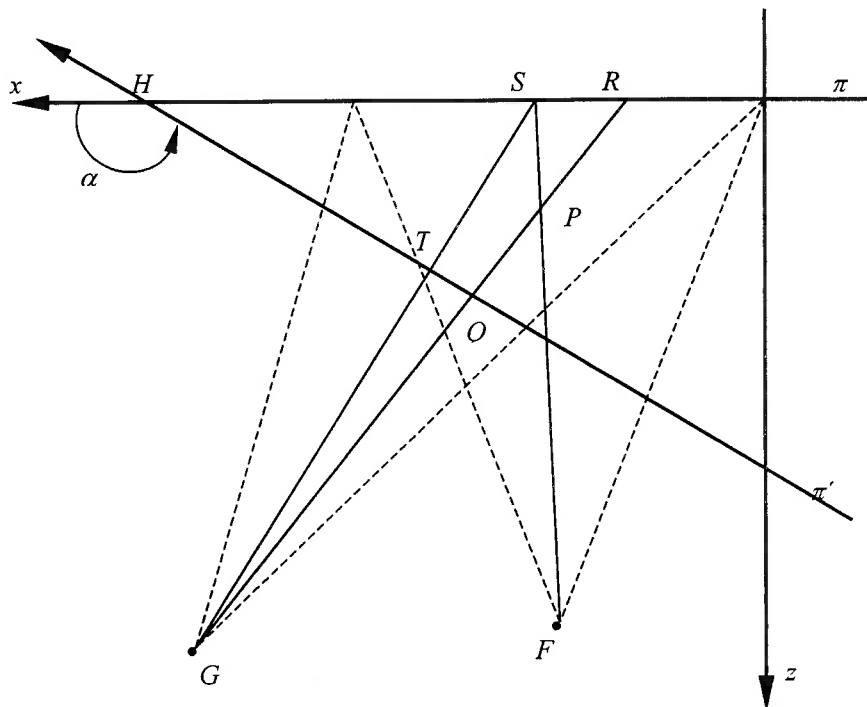


Figura 4-1 – Esquema óptico geométrico do sistema na nova abordagem.

Utilizou-se a mesma nomenclatura para os pontos, planos e ângulo com a finalidade de estabelecer uma analogia com o modelo existente. Apesar do posicionamento simétrico de câmera, o sistema comporta-se de maneira semelhante. Os pontos avançados no sentido do eixo x em relação aos outros com o posicionamento de câmera à direita continuam aqui avançados devido à mudança de sentido do eixo x . Desta maneira, a mudança de posicionamento do ponto S em relação ao ponto R por exemplo, significa, fisicamente, a mudança no sentido de formação da faixa na cena que contém o objeto.

Isto posto, pode-se considerar para os pontos da figura 4-1 o mesmo equacionamento montado para a figura 2-3, com alteração apenas da origem do sistema no tratamento das imagens.

Analisando-se agora a simplificação da triangulação, segue-se o mesmo princípio observado para o equacionamento da triangulação na nova abordagem. Foi verificada na nova abordagem a mesma formação trapezoidal da imagem, porém de maneira simétrica, como mostra a figura 4-2.

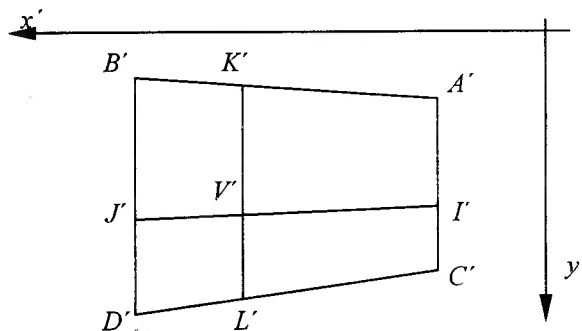


Figura 4-2 – Formação trapezoidal da imagem na nova abordagem.

A alteração no posicionamento dos pontos em relação ao eixo y' não modifica a simplificação da triangulação por causa da alteração combinada do eixo x' . Deste modo, o procedimento estabelecido para a simplificação da triangulação exposto em 2.2.4 será aqui utilizado da mesma maneira.

4.2 Calibração do Sistema

Uma vez definida a abordagem para a recuperação de forma, procede-se a calibração do sistema. O objetivo da calibração é determinar os valores dos termos z_F , x_G e z_G presentes nas equações (2.13). Estes termos localizam as posições do projetor e da câmera no sistema.

A calibração de sistemas baseados em luz estruturada é tema de vários trabalhos disponíveis na literatura. Zhang [44] utiliza um conjunto de equações que calculam os parâmetros procurados em imagens capturadas em pelo menos duas posições diferentes de câmera. Verificou-se também a proposta de translação do objeto de calibração e a medição nas diferentes posições, estabelecendo-se um posicionamento corrigido para o objeto a partir de um referencial único [45].

Como método de calibração, foi adotado o processo numérico desenvolvido por Nishida [38] com conceitos de algoritmo genético. Este processo parte de um conjunto de valores iniciais para z_F , x_G e z_G , reconstrói o objeto utilizando as equações da triangulação 2.13 e determina as coordenadas (x_P , y_P e z_P) dos pontos encontrados. Estas coordenadas apresentarão erros quando comparadas às dimensões reais do objeto. A minimização destes erros leva ao conjunto de valores z_F , x_G e z_G procurado.

O conceito de algoritmo genético está baseado na evolução dos seres biológicos, considerando a existência de genótipos, fenótipos, seleção natural e mutação na geração de filhos a partir de um pai. Os genótipos são os termos z_F , x_G e z_G procurados no sistema, atuando como fenótipos através da reconstrução em três dimensões dos objetos em análise. A partir de um conjunto de valores pais, novos conjuntos de valores nascem como filhos com mutações, pequenas diferenças em relação aos pais. A seleção natural é aplicada através de testes no sistema e da escolha de um resultado ideal. Entende-se por resultado ideal o conjunto de valores z_F , x_G e z_G que mais aproximam a reconstrução em altura do objeto, z_P , do conhecido para os objetos em análise. A geração de pais e filhos prossegue até que a informação de altura z_P calculada seja muito próxima ao z_P do objeto conhecido.

A figura 4-3 esquematiza a calibração com conceito de algoritmo genético.

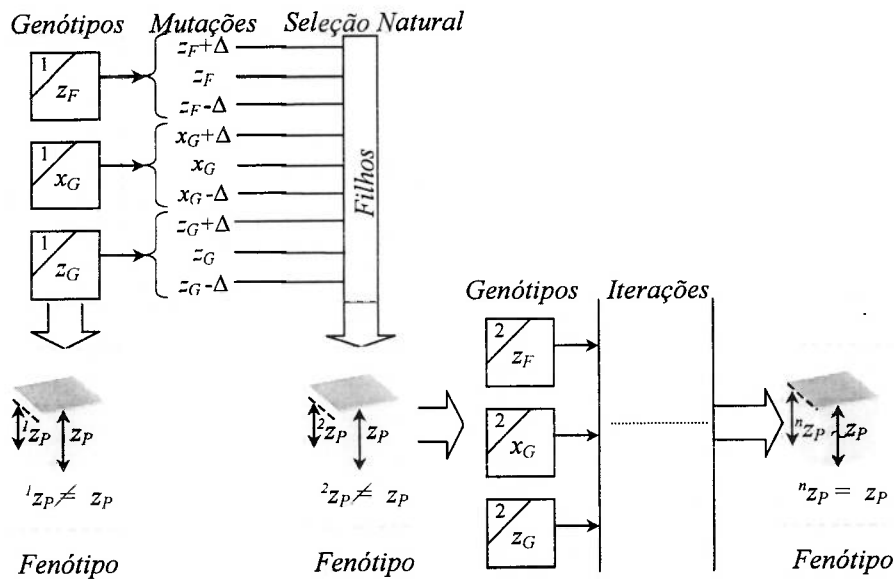


Figura 4-3 – Esquema de calibração por algoritmo genético.

Inicia-se a calibração, assumindo-se para z_F , x_G e z_G iniciais os valores equivalentes à medição no local do ensaio. Este primeiro conjunto de valores é então testado, utilizando-se as equações da triangulação (2.13) e comparando-se o resultado com a altura de objeto esperada. Os conjuntos de valores testados a seguir não são apenas os resultados de variações somadas ou subtraídas em cada um dos parâmetros, mas também as combinações possíveis dos parâmetros com as variações. O valor correspondente ao comprimento focal máximo da câmera (23,7 mm) foi assumido como valor inicial da variação. Assim, a partir de 3 parâmetros (z_F , x_G e z_G) e 3 situações de variação (soma, subtração e o valor inalterado), tem-se 27 possibilidades de teste. Da mesma maneira que o conjunto de valores iniciais, as 27 possibilidades são testadas. Havendo impossibilidade de atingir conjuntos de valores melhores que o inicial, reduz-se a variação de cada parâmetro à metade.

A procura pelos conjuntos de valores com combinações que proporcionem um valor de altura calculado próximo ao do esperado prossegue até que a variação atinja valores menores que 1 pixel.

O objeto utilizado para a calibração do sistema foi o paralelepípedo da seção 3.5.1.1.

A calibração do sistema é feita numa etapa anterior ao processamento das imagens e logo após o pré-processamento e a seleção das informações mais precisas das imagens capturadas a partir dos dois posicionamentos diferentes de câmera.

Em razão da nova abordagem, o sistema foi calibrado a partir das imagens capturadas nos dois diferentes posicionamentos de câmera (à esquerda e à direita do projetor). Isto foi aplicado nas imagens de maneira individual, obtendo-se dois conjuntos de valores z_F , x_G e z_G , um para cada imagem capturada.

O motivo de a calibração ser efetuada nesta etapa deve-se ao uso da nova abordagem. A característica da calibração de procura pelo conjunto de valores que reconstrua o objeto com maior precisão depende da uniformidade de informações na imagem a ser processada. No caso da abordagem preliminar, verificou-se através da figura 3.9 que existe uma não-uniformidade de informações visualizada através da diferença considerável de altura obtida entre os lados esquerdo e direito da imagem. O uso da técnica de calibração aqui apresentada na abordagem preliminar leva a um conjunto de valores z_F , x_G e z_G que diverge totalmente, gerando maiores erros de reconstrução. A diminuição desta não-uniformidade através do uso da nova abordagem permitiu o uso desta técnica de calibração com sucesso.

4.3 Ensaio da Nova Abordagem

Com o propósito de aplicar a nova abordagem e a calibração do sistema, ensaiou-se a mesma meia esfera apresentada na seção 3.3. Os posicionamentos de câmera e projetor medidos no local do ensaio são apresentados na tabela 4-1.

z_F	x_G	z_G
1490,9	1041,8	2000

Tabela 4-1 – Valores medidos para os posicionamentos do projetor e da câmera.

Utilizou-se o mesmo posicionamento de câmera para os lados esquerdo e direito do projetor. Como já citado acima, a diferença de cada lado está na origem considerada no tratamento das imagens.

A figura 4-4 mostra as imagens capturadas pela câmera posicionada conforme a tabela 4-1.

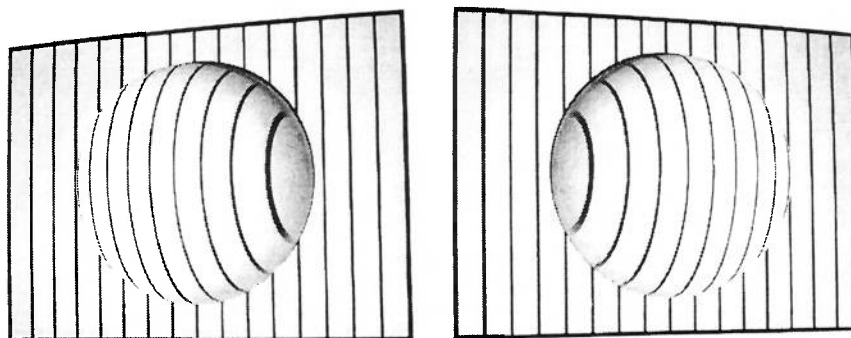


Figura 4-4 – Imagens capturadas pela câmera posicionada à direita e à esquerda do projetor.

O pré-processamento é aplicado individualmente para cada imagem capturada. Em seguida, descartou-se a porção correspondente à metade da imagem no lado oposto ao da câmera. Isto foi feito baseando-se nas análises da seção 3.5.

As imagens são, então, submetidas à calibração. Conforme exposto em 4.2, utilizou-se o paralelepípedo da seção 3.5.1.1, capturando-se as imagens nos posicionamentos de câmera e projetor apresentados na tabela 4-1. A calibração determinou os valores de z_F , x_G e z_G apresentados na tabela 4-2.

<i>Posição de câmera em relação ao projetor</i>	z_F	x_G	z_G
Direita	1366,6	625,4	1604,5
Esquerda	1369,7	621,3	1597,4

Tabela 4-2 – Valores obtidos por calibração para os posicionamentos do projetor e da câmera.

Observam-se diferenças expressivas entre os valores dos parâmetros z_F , x_G e z_G nas tabelas 4-1 e 4-2. Estas diferenças podem ser explicadas devido à característica de ponderação da calibração, otimizando os parâmetros de maneira

iterativa através apenas dos resultados das reconstruções. Isto significa que não há nenhuma restrição quanto às combinações e aos valores adotados para os posicionamentos de câmera e projetor no sistema de calibração aqui adotado.

Apesar das diferenças observadas entre os valores dos parâmetros, obteve-se na calibração um erro médio total de 0,26% na reconstrução do paralelepípedo.

A etapa seguinte é o processamento. Nele, o sistema faz a leitura das informações e os cálculos da triangulação para cada imagem, efetuando a representação em conjunto.

A figura 4-5 mostra a reconstrução da meia esfera ensaiada na nova abordagem.

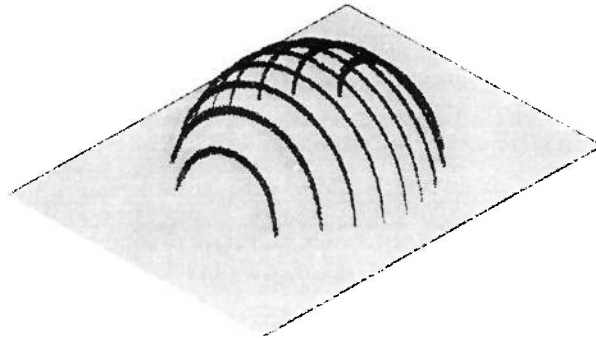


Figura 4-5 – Reconstrução da meia esfera ensaiada na nova abordagem.

CAPÍTULO V – RECUPERAÇÃO DE FORMA DE PEÇAS ESTAMPADAS

5.1 Peças Estampadas Utilizadas

Neste momento, aplica-se todo o desenvolvimento feito até aqui para a recuperação de forma dos componentes estampados. Os nomes dos componentes adotados com as principais dimensões em milímetros, obtidas por trena são mostrados na figura 5-1.

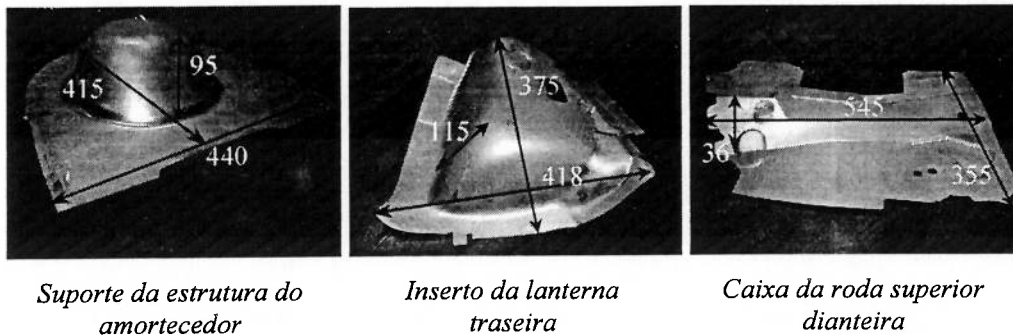


Figura 5-1 – Componentes estampados adotados para a reconstrução.

5.2 Preparação dos Ensaios

As peças estampadas foram submetidas à recuperação de forma logo após o processo de estampagem. Não houve a necessidade de limpeza das peças ou tratamento de superfície, já que as peças atenderam de maneira satisfatória a reflexão do padrão projetado.

A preocupação quanto à fixação das peças no plano de projeção foi a de ter a certeza que a parte mais baixa em altura da peça estaria totalmente sobre o plano de projeção. Esta preocupação visa evitar erros de medição. Foram utilizados

imãs para a fixação das peças estampadas, aproveitando-se as propriedades magnéticas do aço.

Adotou-se a escala de 1 milímetro para cada 1,25 pixel. Os três componentes foram capturados com o mesmo posicionamento de câmera e projetor.

Os valores calibrados para os parâmetros z_F , x_G e z_G são apresentados na tabela 5-1.

<i>Posição de câmera em relação ao projetor</i>	z_F	x_G	z_G
Direita	1500	912,5	1775,6
Esquerda	1402	992,5	1935,6

Tabela 5-1 – Valores obtidos por calibração no posicionamento do projetor e da câmera para os componentes estampados.

5.3 Resultados de Recuperação de Forma

A figura 5-2 mostra as imagens capturadas pela câmera para o suporte da estrutura do amortecedor.

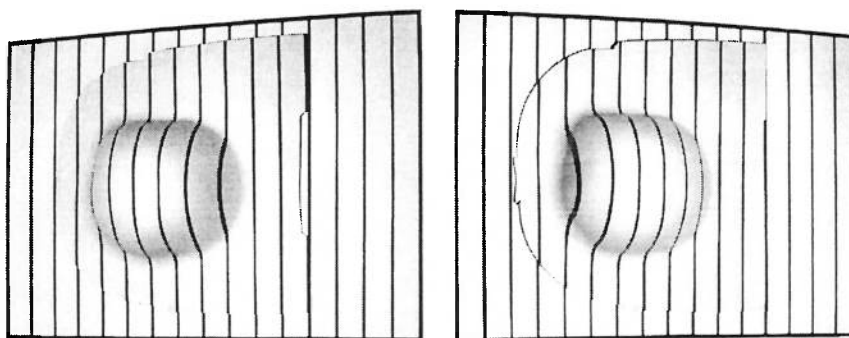


Figura 5-2 – Imagens capturadas pela câmera do suporte da estrutura do amortecedor.

A figura 5-3 mostra a reconstrução do suporte da estrutura do amortecedor.

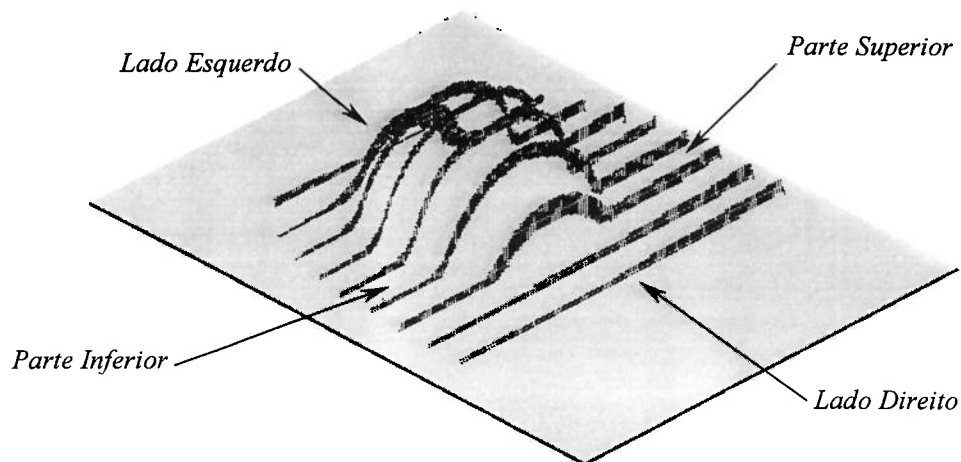


Figura 5-3 – Reconstrução do suporte da estrutura do amortecedor.

Observa-se que o sistema reproduziu a forma do suporte da estrutura do amortecedor ao longo das listras. A suavidade da superfície deste componente permitiu uma boa reconstrução. Destacam-se as diferenças de altura entre as partes superior e inferior do componente.

5.3.1 Comparação de Medição

Com o intuito de validar o sistema em estudo, comparou-se a medição apresentada em 5.3 com aquela feita por uma máquina de medição por coordenadas (MMC). Por causa da excessiva demanda de tempo necessário para a recuperação de forma por uma MMC, já citada no capítulo 1 deste trabalho, decidiu-se realizar uma medição parcial do componente estampado para a comparação.

Assim sendo, o suporte da estrutura do amortecedor foi submetido à medição na MMC, posicionando-se a referência para a medição na metade do recorte no lado direito e normal à superfície da peça. A partir desta referência, percorreu-se com o apalpador da MMC todo o comprimento da peça, do lado direito para o

esquerdo, sob a mesma coordenada correspondente à largura, criando-se uma seção. A referência adotada e a seção gerada a partir da medição são esquematizados na figura 5-4.

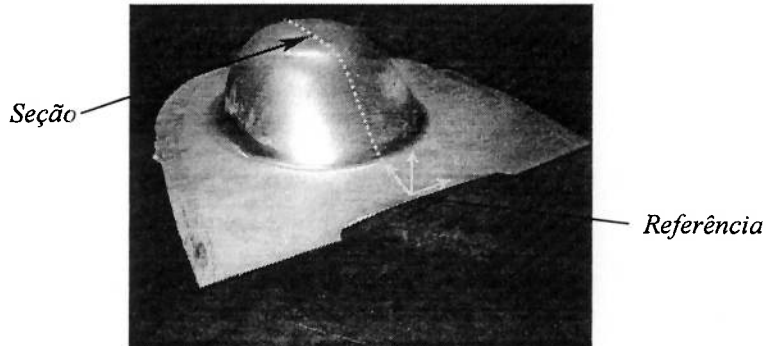


Figura 5-4 – Referência adotada e seção na peça na medição pela MMC.

Foram adquiridos 114 pontos na medição pela MMC. A visualização dos pontos foi possível através de um software CAD. Neste software, foi capturada a imagem dos pontos adquiridos mostrados em escala na figura 5-5.

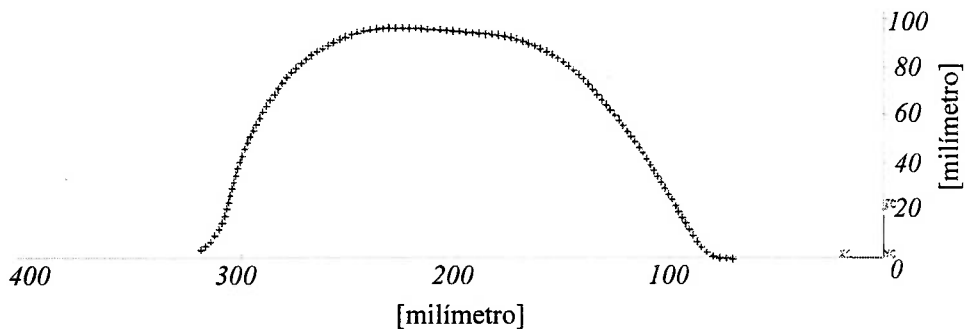


Figura 5-5 – Imagem de captura dos pontos adquiridos pela MMC num software CAD.

Em seguida, migrou-se a referência adotada para a medição na MMC para aquela definida para o sistema de recuperação de forma por luz estruturada. Os pontos obtidos por luz estruturada foram convertidos de pixel para milímetros, obedecendo a escala adotada - 1 milímetro para cada 1,25 pixel.

Coletaram-se os pontos obtidos na medição com luz estruturada, criando-se um corte na peça em estudo na mesma posição e direção da seção criada na medição com a MMC.

A figura 5-6 esquematiza a direção da seção criada na medição do suporte da estrutura do amortecedor com a MMC.

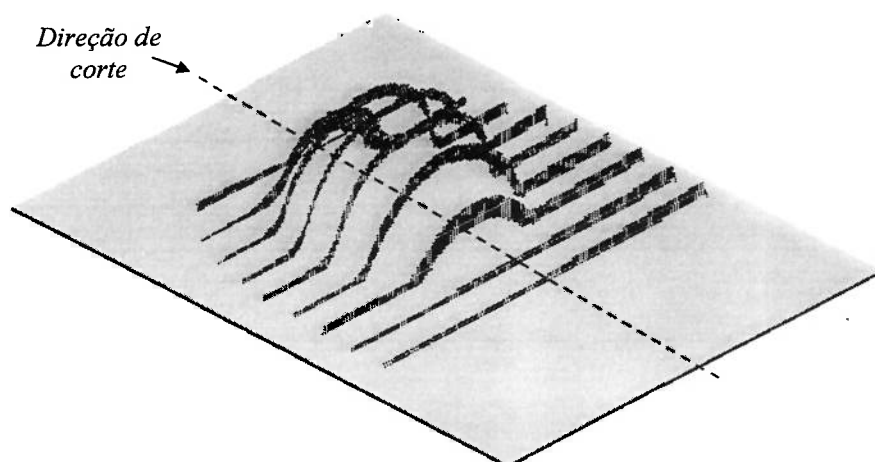


Figura 5-6 – Esquema de corte na medição por luz estruturada na peça.

Finalmente, os pontos medidos pela MMC e por luz estruturada foram reunidos para a comparação. A figura 5-7 mostra a comparação da medição em milímetros entre a MMC e a luz estruturada numa seção do suporte da estrutura do amortecedor.

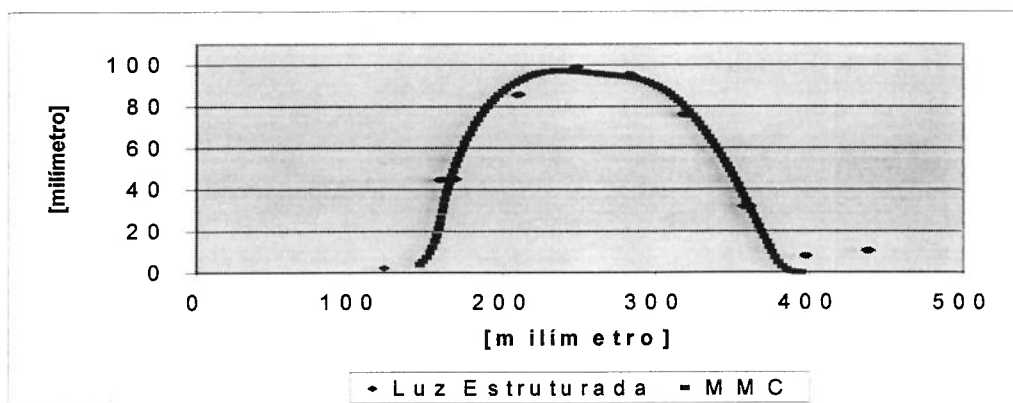


Figura 5-7 – Comparação de medição.

Devido a alta precisão das MMCs, da ordem de microns, e à precisão esperada deste sistema, 1 pixel, é logicamente admissível que a medição feita pela MMC está mais próxima da realidade do que a por luz estruturada.

Analisando-se a figura 5-7, percebe-se que a maioria dos pontos obtidos por luz estruturada está próxima da medição feita pela MMC, mostrando a capacidade deste sistema na medição de formas tridimensionais. A rigor, pode-se comparar os pontos obtidos por luz estruturada e pela MMC que possuem as mesmas coordenadas horizontais. Assim, a máxima diferença comparável obtida entre a altura do sistema aqui proposto e a MMC foi de 2,1 mm na coordenada horizontal 362,4 mm.

É importante salientar a eficiência do sistema aqui proposto, pois a recuperação de forma do suporte da estrutura do amortecedor levou cerca de 15 minutos para ser realizada.

Apesar dos valores diferentes de calibração para os parâmetros z_F , x_G e z_G determinados para as imagens capturadas com diferentes posicionamentos, foram obtidas representações com proximidades similares em relação à medição pela MMC.

Verifica-se também, através da figura 5-7, que os pontos obtidos por luz estruturada de coordenadas horizontais entre 400 e 450 milímetros não puderam ser comparados com a medição feita pela MMC, uma vez que não foram lidos pontos na zona correspondente pela MMC.

Deve-se observar que esta comparação também está sujeita a erros devido à migração de referência. Isto se deve às diferentes orientações adotadas para as medições pela MMC e pelo método aqui exposto.

5.3.2 Comparação de Forma

As figuras 5-8 e 5-9 a seguir apresentam a reconstrução dos outros dois componentes estampados mostrados na figura 5-1 e os reais componentes ao lado, para que seja feita uma comparação visual de forma.

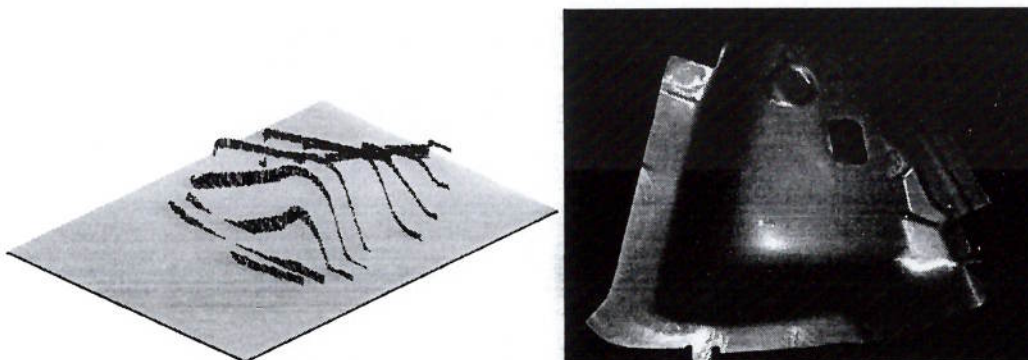


Figura 5-8 – Reconstrução e imagem real do inserto da lanterna traseira.

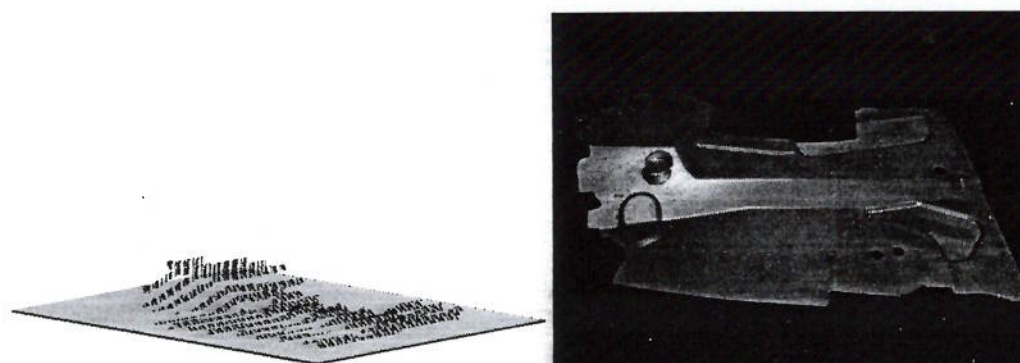


Figura 5-9 – Reconstrução e imagem real da caixa da roda superior dianteira.

As reconstruções acima mostraram a capacidade do sistema em reproduzir a forma dos componentes estampados de diferentes formas e alturas. Entretanto, não foi possível efetuar a reconstrução com os mesmos níveis de detalhe verificados na superfície destes componentes. A causa deste fato está no tipo de padrão de projeção utilizado. Um padrão de projeção com codificação mais densa permitiria obter maior detalhamento na reconstrução dos componentes.

6 Conclusões

Apresentou-se um sistema capaz de recuperar a forma de peças estampadas. Este sistema, já analisado por Nishida [38], foi adaptado a fim de se obter melhor precisão de reconstrução e também eliminar as áreas de oclusão. Isso foi possível através da captura de mais informações do objeto sob posicionamento oposto de câmera em relação ao projetor, da seleção das informações mais precisas e da combinação de resultados na nova abordagem proposta.

As peças estampadas puderam ser medidas sem maiores problemas com relação à visibilidade dos padrões nelas projetados.

Registrou-se rapidez na aquisição e processamento das informações deste sistema. A recuperação de um componente estampado, entre posicionamento dos equipamentos e reconstrução, dura cerca quinze minutos com os equipamentos utilizados aqui. Este fato, além daquele verificado quanto à precisão de reconstrução e da leitura das áreas de oclusão, também justifica a aplicação da nova abordagem.

O sistema desenvolvido é totalmente aplicável ao ambiente de trabalho. Os controles de luminosidade e a disposição de um plano de projeção, bem como a região para o posicionamento de câmera e projetor no ambiente de recuperação de forma, tornam possível o emprego da técnica.

Os usos apenas de um computador, câmera e projetor permitem atribuir a qualidade de portátil ao sistema. Levando-se em conta que os espaços físicos estão cada vez mais ocupados, esta qualidade também é vista de maneira positiva neste contexto. Além disso, os equipamentos podem ser encontrados com facilidade no mercado, atendendo ao requisito de acessibilidade do sistema.

Operadores não-especialistas podem utilizar o sistema devido à praticidade de operação verificada. Isso é importante uma vez que já não haverá especialistas disponíveis para praticar a recuperação de forma na ferramentaria. A ausência de necessidade de profundos conhecimentos da técnica para a operação do sistema dispensa também os treinamentos específicos.

Identificou-se também a flexibilidade do sistema na aplicação em componentes estampados de diferentes formas e alturas. Considerando-se as variadas formas de peças estampadas que compõe a carroceria, aliadas às mudanças de estilo de um automóvel de tempos em tempos, torna-se importante um sistema flexível.

Quanto à precisão, o sistema atendeu às expectativas. Um erro de 2,1 mm foi obtido na medição de uma peça estampada com dimensões aproximadas de 415 x 440 x 95 mm.

Verificou-se a possibilidade de automatização do sistema. A abertura a modificações e a tendência da busca de menos tempo na recuperação de forma podem levar a um maior nível de automatização do sistema.

Sugere-se como possibilidade de trabalhos futuros o incremento na captura de imagens do objeto através de diferentes posicionamentos de câmera. De acordo com o apresentado aqui, a captura de imagens sob outro posicionamento de câmera eliminou as áreas de oclusão na direção do eixo x . O posicionamento de câmera acima e abaixo do projetor poderia eliminar também as áreas de oclusão na direção do eixo y .

Além disso, sugere-se também o uso de padrões com codificação mais densa do que a utilizada aqui, que permitiria obter maior quantidade de dados sobre a superfície do objeto.

ANEXO A – CÁLCULOS DE PONTO FOCAL

Apresentam-se aqui os cálculos dos pontos focais da câmera e do projetor. Estes cálculos foram aplicados da mesma maneira em cada ensaio do sistema. Ressalta-se que os cálculos foram feitos para obter os valores iniciais de z_F , x_G e z_G . Os valores finais destes termos são definidos após a calibração do sistema.

O comprimento focal f da câmera varia entre 7,9 e 23,7 mm [46]. Sabendo-se que a abertura do diafragma e a distância focal são definidas de maneira automática pela câmera no momento da captura da imagem, assumiu-se um comprimento focal de 10 mm para a câmera em todos os ensaios realizados. Os cálculos de x_G e z_G a partir da medição e da consideração do ponto focal seguem abaixo. O índice GM para x e z indica o valor obtido por medição no ensaio.

$$x_G = x_{GM} + 10 \cdot \text{sen } \alpha$$

$$z_G = z_{GM} + 10 \cdot \text{cos } \alpha$$

No caso do projetor, assumiu-se um comprimento focal de 15 mm. O cálculo do z_F considerando-se o comprimento focal é mostrado abaixo. Assim como para a câmera, o índice FM indica o valor obtido por medição no ensaio.

$$z_F = z_{FM} + 15$$

Após calculados, estes valores são convertidos para pixels, dependendo da escala adotada em cada ensaio. Os valores apresentados para z_F , x_G e z_G nos ensaios já incluem este cálculo.

A figura A-1 esquematiza o cálculo para os pontos focais de câmera e projetor.

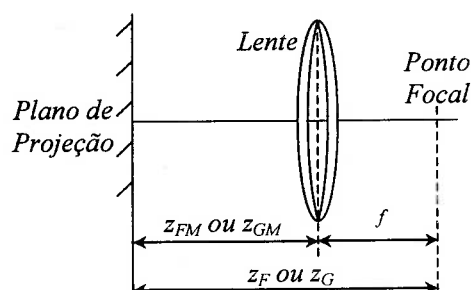


Figura A-1 – Esquema para cálculo do ponto focal.

ANEXO B – EQUACIONAMENTO DO SISTEMA

Será apresentado a seguir o desenvolvimento das equações matemáticas do sistema. A figura 2-3 é adotada como esquema óptico e geométrico no modelamento do sistema.

B.1 – Determinação do Ponto S

$$\text{Equação da reta } FP: \begin{cases} x_{FP} = x_F + (x_P - x_F) \cdot \lambda \\ y_{FP} = y_F + (y_P - y_F) \cdot \lambda \\ z_{FP} = z_F + (z_P - z_F) \cdot \lambda \end{cases} \quad (\text{B.1})$$

$$\text{Equação do plano } \pi: z_{\pi} = 0 \quad (\text{B.2})$$

Para o ponto S: $z_{FP} = z_{\pi}$, daí:

$$0 = z_F + (z_P - z_F) \cdot \lambda$$

$$\lambda = \frac{z_F}{z_F - z_P}$$

Fazendo-se a substituição de λ em (B.1) para encontrar x_S :

$$x_S = x_F + (x_P - x_F) \cdot \left(\frac{z_F}{z_F - z_P} \right)$$

$$x_S = \frac{x_F \cdot z_P - z_F \cdot x_P}{z_P - z_F}$$

Da mesma maneira para em (B.2) encontrar y_S :

$$y_S = y_F + (y_P - y_F) \cdot \left(\frac{z_F}{z_F - z_P} \right)$$

$$y_S = \frac{y_F \cdot z_P - y_P \cdot z_F + y_P \cdot z_F - y_F \cdot z_F}{z_F - z_P}$$

$$y_S = \frac{y_F \cdot z_P - z_F \cdot y_P}{z_P - z_F}$$

Portanto o ponto S fica:

$$\left\{ \begin{array}{l} x_S = \frac{x_F \cdot z_P - z_F \cdot x_P}{z_P - z_F} \end{array} \right. \quad (\text{B.3})$$

$$\left\{ \begin{array}{l} y_S = \frac{y_F \cdot z_P - z_F \cdot y_P}{z_P - z_F} \end{array} \right. \quad (\text{B.4})$$

$$\left\{ \begin{array}{l} z_S = 0 \end{array} \right. \quad (\text{B.5})$$

B.2 – Determinação do Ponto Q

$$\text{Equação da reta } GP: \left\{ \begin{array}{l} x_{GP} = x_G + (x_P - x_G) \cdot \lambda \end{array} \right. \quad (\text{B.6})$$

$$\left\{ \begin{array}{l} y_{GP} = y_G + (y_P - y_G) \cdot \lambda \end{array} \right. \quad (\text{B.7})$$

$$\left\{ \begin{array}{l} z_{GP} = z_G + (z_P - z_G) \cdot \lambda \end{array} \right. \quad (\text{B.8})$$

Utilizou-se o ponto U (cruzamento do plano π' com o eixo z), o ângulo β (suplementar ao ângulo α), o vetor \overrightarrow{HU} (do ponto H ao ponto U), o vetor \vec{v} (coincidente com o eixo y) e o vetor \vec{n} (perpendicular ao plano π') para a determinação da equação do plano π' . O esquema da figura B-1, elaborado a partir da figura 2-3, mostra a localização destes entes geométricos no sistema.

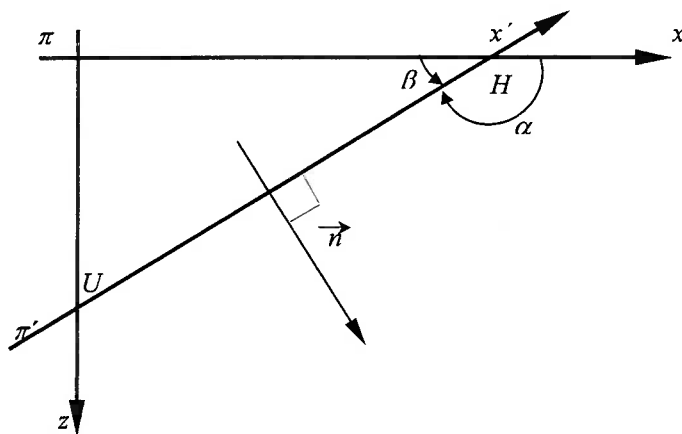


Figura B-1 – Esquema para determinação da equação do plano π' .

Determina-se inicialmente o valor de $|\overrightarrow{HU}|$, através dos pontos H e U .

Assim, sendo:

$$H(x_H, y_H, 0)$$

$$U(0, y_U, z_U)$$

Sabe-se que x_H é conhecido e $y_H = y_U$, deve-se encontrar então z_U . Pelo esquema da figura B-1:

$$\text{sen } \beta = \frac{z_U}{|\overrightarrow{HU}|} \quad \text{e} \quad \cos \beta = \frac{x_H}{|\overrightarrow{HU}|}$$

Sabe-se também que:

$$\text{sen } \beta = \text{sen } \alpha \quad \text{e} \quad \cos \beta = -\cos \alpha$$

Faz-se então:

$$\begin{cases} x_H = -\cos \alpha \cdot |\overrightarrow{HU}| & \text{(B.9)} \\ z_U = \text{sen } \alpha \cdot |\overrightarrow{HU}| & \text{(B.10)} \end{cases}$$

Dividindo-se (B.9) por (B.10), obtém-se z_U :

$$\frac{x_H}{z_U} = \frac{-\cos \alpha}{\text{sen } \alpha}$$

$$z_U = \frac{-x_H \text{ sen } \alpha}{\cos \alpha}$$

Assim,

$$|\overrightarrow{HU}| = \left(-x_H, 0, \frac{-x_H \text{ sen } \alpha}{\cos \alpha} \right)$$

Utilizando-se agora o vetor \vec{v} definido por:

$$\vec{v} = (0, 1, 0)$$

Já que $|\overrightarrow{HU}|$, \vec{v} e \vec{n} são ortogonais, faz-se então o produto vetorial de $|\overrightarrow{HU}|$ por \vec{v} para encontrar \vec{n} :

$$\vec{n} = |\overrightarrow{HU}| \wedge \vec{v} = \left(\frac{x_H \cdot \text{sen } \alpha}{\cos \alpha}, 0, -x_H \right)$$

Desta forma sabe-se que a equação de π' será:

$$\frac{x_H \cdot \text{sen } \alpha}{\cos \alpha} \cdot x_{\pi'} + 0 \cdot y_{\pi'} - x_H \cdot z_{\pi'} + d = 0 \quad (\text{B.11})$$

Para se encontrar d , basta substituir um ponto conhecido (H ou U) na equação (B.11). Substituindo-se o ponto H :

$$\frac{x_H \cdot \text{sen } \alpha}{\cos \alpha} \cdot x_H - x_H \cdot 0 + d = 0$$

$$d = \frac{-x_H^2 \cdot \text{sen } \alpha}{\cos \alpha}$$

Finalmente a equação do plano π' fica:

$$\frac{x_H \cdot \text{sen } \alpha}{\cos \alpha} \cdot x_{\pi'} + 0 \cdot y_{\pi'} - x_H \cdot z_{\pi'} - \frac{-x_H^2 \cdot \text{sen } \alpha}{\cos \alpha} = 0 \quad (\text{B.12})$$

Multiplicando-se toda a equação (B.12) por $\cos \alpha$ e dividindo-se por x_H , chega-se a uma forma visualmente melhor:

$$\text{sen } \alpha \cdot x_{\pi'} - \cos \alpha \cdot z_{\pi'} - x_H \cdot \text{sen } \alpha = 0 \quad (\text{B.13})$$

Obtidas as equações do plano π' (B.13) e da reta GP (B.6), (B.7) e (B.8), seguem-se os cálculos para a obtenção das equações do ponto Q , bastando agora isolar e substituir alguns termos destas equações. Desta forma, isolando-se o λ em (B.6):

$$\lambda = \frac{x_Q - x_G}{x_P - x_G}$$

Substituindo-se este λ em (B.8):

$$z_Q = z_G + (z_P - z_G) \cdot \left(\frac{x_Q - x_G}{x_P - x_G} \right) \quad (\text{B.14})$$

Isolando-se o termo z_Q de (B.13):

$$z_Q = \frac{-x_H \cdot \text{sen } \alpha + x_Q \cdot \text{sen } \alpha}{\cos \alpha} \quad (\text{B.15})$$

Igualando-se (B.14) a (B.15), obtém-se x_Q :

$$z_G + (z_P - z_G) \cdot \left(\frac{x_Q - x_G}{x_P - x_G} \right) = \frac{-x_H \cdot \text{sen } \alpha + x_Q \cdot \text{sen } \alpha}{\text{cos } \alpha}$$

$$x_Q = \frac{x_G \cdot x_H \cdot \text{sen } \alpha - x_P \cdot x_H \cdot \text{sen } \alpha + x_G \cdot z_P \cdot \text{cos } \alpha - z_G \cdot x_P \cdot \text{cos } \alpha}{z_P \cdot \text{cos } \alpha - z_G \cdot \text{cos } \alpha - x_P \cdot \text{sen } \alpha + x_G \cdot \text{sen } \alpha} \quad (\text{B.16})$$

Procede-se da mesma maneira para se determinar z_Q . Isolando-se o λ em (B.8):

$$\lambda = \frac{z_Q - z_G}{z_P - z_G}$$

Substituindo-se este λ em (B.6):

$$x_Q = x_G + (x_P - x_G) \cdot \left(\frac{z_Q - z_G}{z_P - z_G} \right) \quad (\text{B.17})$$

Isolando-se o termo x_Q em (B.13):

$$x_Q = \frac{x_H \cdot \text{sen } \alpha + z_Q \cdot \text{cos } \alpha}{\text{sen } \alpha} \quad (\text{B.18})$$

Igualando-se (B.17) a (B.18), obtém-se z_Q :

$$x_G + (x_P - x_G) \cdot \left(\frac{z_Q - z_G}{z_P - z_G} \right) = \frac{x_H \cdot \text{sen } \alpha + z_Q \cdot \text{cos } \alpha}{\text{sen } \alpha}$$

$$z_Q = \frac{x_G \cdot z_P \cdot \text{sen } \alpha - z_P \cdot x_H \cdot \text{sen } \alpha - z_G \cdot x_P \cdot \text{sen } \alpha + z_G \cdot x_H \cdot \text{sen } \alpha}{z_P \cdot \text{cos } \alpha - z_G \cdot \text{cos } \alpha - x_P \cdot \text{sen } \alpha + x_G \cdot \text{sen } \alpha} \quad (\text{B.19})$$

O próximo passo é encontrar y_Q . Para isso multiplica-se cada termo das equações (B.6) por $\text{sen } \alpha$ e (B.8) por $\text{cos } \alpha$ respectivamente, obtendo-se:

$$x_Q \cdot \text{sen } \alpha = x_G \cdot \text{sen } \alpha + (x_P - x_G) \cdot \lambda \cdot \text{sen } \alpha \quad (\text{B.20})$$

$$z_Q \cdot \text{cos } \alpha = z_G \cdot \text{cos } \alpha + (z_P - z_G) \cdot \lambda \cdot \text{cos } \alpha \quad (\text{B.21})$$

A seguir subtrai-se (B.20) de (B.21), obtendo-se:

(B.22)

$$x_Q \cdot \text{sen } \alpha - z_Q \cdot \text{cos } \alpha = x_G \cdot \text{sen } \alpha - z_G \cdot \text{cos } \alpha + (x_P - x_G) \cdot \lambda \cdot \text{sen } \alpha - (z_P - z_G) \cdot \lambda \cdot \text{cos } \alpha$$

Da equação (B.13), sabe-se que:

$$\text{sen } \alpha \cdot x_Q - \text{cos } \alpha \cdot z_Q = x_H \cdot \text{sen } \alpha \quad (\text{B.23})$$

Assim pode-se igualar o segundo termo da equação (B.23) ao segundo termo da equação (B.22), para se obter λ :

$$x_H \cdot \text{sen } \alpha = x_G \cdot \text{sen } \alpha - z_G \cdot \text{cos } \alpha + (x_P - x_G) \cdot \lambda \cdot \text{sen } \alpha - (z_P - z_G) \cdot \lambda \cdot \text{cos } \alpha$$

$$\lambda = \frac{x_H \cdot \text{sen } \alpha - x_G \cdot \text{sen } \alpha + z_G \cdot \text{cos } \alpha}{\text{sen } \alpha \cdot (x_P - x_G) - \text{cos } \alpha \cdot (z_P - z_G)}$$

y_Q é determinado pela substituição deste λ em (B.7):

$$y_Q = y_G + (y_P - y_G) \cdot \left(\frac{x_H \cdot \text{sen } \alpha - x_G \cdot \text{sen } \alpha + z_G \cdot \text{cos } \alpha}{\text{sen } \alpha \cdot (x_P - x_G) - \text{cos } \alpha \cdot (z_P - z_G)} \right) \quad (\text{B.24})$$

$$y_Q = \frac{y_G \cdot z_P \cdot \text{cos } \alpha + y_P \cdot x_G \cdot \text{sen } \alpha - y_P \cdot z_G \cdot \text{cos } \alpha + y_G \cdot x_H \cdot \text{sen } \alpha - y_G \cdot x_P \cdot \text{sen } \alpha - y_P \cdot x_H \cdot \text{sen } \alpha}{z_P \cdot \text{cos } \alpha - z_G \cdot \text{cos } \alpha - x_P \cdot \text{sen } \alpha + x_G \cdot \text{sen } \alpha}$$

B.3 – Determinação do Ponto T

$$\text{Equação da reta } GS: \begin{cases} x_{GS} = x_G + (x_S - x_G) \cdot \lambda & (\text{B.25}) \\ y_{GS} = y_G + (y_S - y_G) \cdot \lambda & (\text{B.26}) \\ z_{GS} = z_G + (z_S - z_G) \cdot \lambda & (\text{B.27}) \end{cases}$$

Substituindo x_S , y_S e z_S , determinados em (B.3), (B.4) e (B.5) nas equações (B.25), (B.26) e (B.27), tem-se:

$$\left\{ \begin{array}{l} x_T = x_G + \left[\frac{x_F \cdot z_P - z_F \cdot x_P}{z_P - z_F} - x_G \right] \cdot \lambda \\ y_T = y_G + \left[\frac{y_F \cdot z_P - z_F \cdot y_P}{z_P - z_F} - y_G \right] \cdot \lambda \\ z_T = z_G - z_G \cdot \lambda \end{array} \right. \quad (\text{B.28})$$

$$\left\{ \begin{array}{l} x_T = x_G + \left[\frac{x_F \cdot z_P - z_F \cdot x_P}{z_P - z_F} - x_G \right] \cdot \lambda \\ y_T = y_G + \left[\frac{y_F \cdot z_P - z_F \cdot y_P}{z_P - z_F} - y_G \right] \cdot \lambda \end{array} \right. \quad (\text{B.29})$$

$$\left\{ \begin{array}{l} x_T = x_G + \left[\frac{x_F \cdot z_P - z_F \cdot x_P}{z_P - z_F} - x_G \right] \cdot \lambda \\ y_T = y_G + \left[\frac{y_F \cdot z_P - z_F \cdot y_P}{z_P - z_F} - y_G \right] \cdot \lambda \\ z_T = z_G - z_G \cdot \lambda \end{array} \right. \quad (\text{B.30})$$

De (B.13), obtém-se para o ponto T :

$$x_T \cdot \text{sen } \alpha - z_T \cdot \text{cos } \alpha - x_H \cdot \text{sen } \alpha = 0$$

$$x_T = \frac{x_H \cdot \text{sen } \alpha + z_T \cdot \text{cos } \alpha}{\text{sen } \alpha} \quad (\text{B.31})$$

Isolando-se o λ em (B.30), tem-se:

$$\lambda = \frac{z_G - z_T}{z_G}$$

Substituindo-se agora o valor deste λ e o valor de x_T de (B.31) em (B.28), tem-se o valor de z_T :

$$\frac{x_H \cdot \text{sen } \alpha + z_T \cdot \text{cos } \alpha}{\text{sen } \alpha} = x_G + \left[\frac{x_F \cdot z_P - z_F \cdot x_P}{z_P - z_F} - x_G \right] \cdot \left(\frac{z_G - z_T}{z_G} \right)$$

(B.32)

$$z_T = \frac{z_P \cdot z_G \cdot x_H \cdot \text{sen } \alpha - x_F \cdot z_P \cdot z_G \cdot \text{sen } \alpha + z_F \cdot x_P \cdot z_G \cdot \text{sen } \alpha - z_F \cdot z_G \cdot x_H \cdot \text{sen } \alpha}{z_P \cdot x_G \cdot \text{sen } \alpha - x_F \cdot z_P \cdot \text{sen } \alpha - z_F \cdot x_G \cdot \text{sen } \alpha + z_F \cdot x_P \cdot \text{sen } \alpha + z_F \cdot z_G \cdot \text{cos } \alpha - z_P \cdot z_G \cdot \text{cos } \alpha}$$

Da mesma forma determina-se o x_T . Isolando-se o λ em (B.25), tem-se:

$$\lambda = \frac{x_T - x_G}{x_S - x_G}$$

De (B.13), obtém-se para o ponto T :

$$x_T \cdot \text{sen } \alpha - z_T \cdot \text{cos } \alpha - x_H \cdot \text{sen } \alpha = 0$$

$$z_T = \frac{x_T \cdot \text{sen } \alpha - x_H \cdot \text{sen } \alpha}{\text{cos } \alpha} \quad (\text{B.33})$$

Substituindo-se o valor deste λ e valor de z_T de (B.33) em (B.30), tem-se o valor de x_T :

$$\frac{x_T \cdot \text{sen } \alpha - x_H \cdot \text{sen } \alpha}{\text{cos } \alpha} = z_G - z_G \cdot \left(\frac{x_T - x_G}{x_S - x_G} \right)$$

(B.34)

$$x_T = \frac{\frac{z_F \cdot x_P \cdot x_H \cdot \text{sen } \alpha - z_F \cdot x_G \cdot x_H \cdot \text{sen } \alpha + z_F \cdot x_P \cdot z_G \cdot \text{cos } \alpha - z_P \cdot x_F \cdot z_G \cdot \text{cos } \alpha}{z_P \cdot x_G \cdot \text{sen } \alpha - x_F \cdot z_P \cdot \text{sen } \alpha - z_F \cdot x_G \cdot \text{sen } \alpha + z_F \cdot x_P \cdot \text{sen } \alpha + z_F \cdot z_G \cdot \text{cos } \alpha - z_P \cdot z_G \cdot \text{cos } \alpha} + \frac{z_P \cdot x_G \cdot x_H \cdot \text{sen } \alpha - z_P \cdot x_F \cdot x_H \cdot \text{sen } \alpha}{z_P \cdot x_G \cdot \text{sen } \alpha - x_F \cdot z_P \cdot \text{sen } \alpha - z_F \cdot x_G \cdot \text{sen } \alpha + z_F \cdot x_P \cdot \text{sen } \alpha + z_F \cdot z_G \cdot \text{cos } \alpha - z_P \cdot z_G \cdot \text{cos } \alpha}}$$

Para se obter o valor de y_T , multiplica-se (B.25) e (B.27) por $\text{sen } \alpha$ e $\text{cos } \alpha$ respectivamente:

$$x_T \cdot \text{sen } \alpha = x_G \cdot \text{sen } \alpha + (x_S - x_G) \cdot \lambda \cdot \text{sen } \alpha \quad (\text{B.35})$$

$$z_T \cdot \text{cos } \alpha = z_G \cdot \text{cos } \alpha - z_G \cdot \lambda \cdot \text{cos } \alpha \quad (\text{B.36})$$

A seguir subtrai-se (B.35) de (B.36), chegando-se a:

$$x_T \cdot \text{sen } \alpha - z_T \cdot \text{cos } \alpha = x_G \cdot \text{sen } \alpha - z_G \cdot \text{cos } \alpha + (x_S - x_G) \cdot \lambda \cdot \text{sen } \alpha + z_G \cdot \lambda \cdot \text{cos } \alpha \quad (\text{B.37})$$

Sabe-se também de (B.13) que para o ponto T :

$$x_T \cdot \text{sen } \alpha - z_T \cdot \text{cos } \alpha = x_H \cdot \text{sen } \alpha \quad (\text{B.38})$$

Igualando-se o segundo termo da equação (B.37) com o segundo termo da equação (B.38), obtém-se λ :

$$x_G \cdot \text{sen } \alpha - z_G \cdot \text{cos } \alpha + (x_S - x_G) \cdot \lambda \cdot \text{sen } \alpha + z_G \cdot \lambda \cdot \text{cos } \alpha = x_H \cdot \text{sen } \alpha$$

$$\lambda = \frac{x_H \cdot \text{sen } \alpha - x_G \cdot \text{sen } \alpha + z_G \cdot \text{cos } \alpha}{x_S \cdot \text{sen } \alpha - x_G \cdot \text{sen } \alpha + z_G \cdot \text{cos } \alpha}$$

Substituindo-se este valor de λ e os valores de x_S e y_S obtidos em (B.03) e (B.04) na equação (B.29) respectivamente, tem-se o valor de y_T :

$$y_T = y_G + \left[\frac{y_F \cdot z_P - z_F \cdot y_P}{z_P - z_F} - y_G \right] \cdot \left(\frac{x_H \cdot \text{sen } \alpha - x_G \cdot \text{sen } \alpha + z_G \cdot \text{cos } \alpha}{\left(\frac{x_F \cdot z_P - z_F \cdot x_P}{z_P - z_F} \right) \cdot \text{sen } \alpha - x_G \cdot \text{sen } \alpha + z_G \cdot \text{cos } \alpha} \right)$$

(B.39)

$$y_T = \frac{z_F \cdot x_P \cdot y_G \cdot \text{sen } \alpha - x_F \cdot z_P \cdot y_G \cdot \text{sen } \alpha + z_F \cdot y_P \cdot z_G \cdot \text{cos } \alpha - z_F \cdot y_G \cdot x_H \cdot \text{sen } \alpha +}{z_P \cdot x_G \cdot \text{sen } \alpha - x_F \cdot z_P \cdot \text{sen } \alpha - z_F \cdot x_G \cdot \text{sen } \alpha + z_F \cdot x_P \cdot \text{sen } \alpha + z_F \cdot z_G \cdot \text{cos } \alpha - z_P \cdot z_G \cdot \text{cos } \alpha} +$$

$$\frac{z_P \cdot y_G \cdot x_H \cdot \text{sen } \alpha + z_F \cdot y_P \cdot x_H \cdot \text{sen } \alpha - z_P \cdot y_F \cdot x_H \cdot \text{sen } \alpha - z_F \cdot y_P \cdot x_G \cdot \text{sen } \alpha}{z_P \cdot x_G \cdot \text{sen } \alpha - x_F \cdot z_P \cdot \text{sen } \alpha - z_F \cdot x_G \cdot \text{sen } \alpha + z_F \cdot x_P \cdot \text{sen } \alpha + z_F \cdot z_G \cdot \text{cos } \alpha - z_P \cdot z_G \cdot \text{cos } \alpha} +$$

$$\frac{z_P \cdot y_F \cdot x_G \cdot \text{sen } \alpha - z_P \cdot y_F \cdot z_G \cdot \text{cos } \alpha}{z_P \cdot x_G \cdot \text{sen } \alpha - x_F \cdot z_P \cdot \text{sen } \alpha - z_F \cdot x_G \cdot \text{sen } \alpha + z_F \cdot x_P \cdot \text{sen } \alpha + z_F \cdot z_G \cdot \text{cos } \alpha - z_P \cdot z_G \cdot \text{cos } \alpha}$$

B.4 – Determinação do Ponto R

$$\text{Equação da reta } GP: \begin{cases} x_{GP} = x_G + (x_P - x_G) \cdot \lambda \\ y_{GP} = y_G + (y_P - y_G) \cdot \lambda \\ z_{GP} = z_G + (z_P - z_G) \cdot \lambda \end{cases}$$

$$\text{Equação do plano } \pi: z_\pi = 0$$

Para o ponto R: $z_{GP} = z_\pi$ daí:

$$0 = z_G + (z_P - z_G) \cdot \lambda$$

$$\lambda = \frac{z_G}{z_G - z_P}$$

Fazendo-se a substituição de λ para encontrar x_R :

$$x_R = x_G + (x_P - x_G) \cdot \left(\frac{z_G}{z_G - z_P} \right)$$

$$x_R = \frac{x_G \cdot z_P - x_P \cdot z_G}{z_P - z_G}$$

Da mesma maneira para encontrar y_R :

$$y_R = y_G + (y_P - y_G) \cdot \left(\frac{z_G}{z_G - z_P} \right)$$

$$y_R = \frac{z_P \cdot y_G - z_G \cdot y_P}{z_P - z_G}$$

Portanto o ponto R fica:

$$\left\{ \begin{array}{l} x_R = \frac{z_P \cdot x_G - z_G \cdot x_P}{z_P - z_G} \end{array} \right. \quad (\text{B.40})$$

$$\left\{ \begin{array}{l} y_R = \frac{z_P \cdot y_G - z_G \cdot y_P}{z_P - z_G} \end{array} \right. \quad (\text{B.41})$$

$$\left\{ \begin{array}{l} z_R = 0 \end{array} \right.$$

B.5 – Determinação do Termo Δx

Calcula-se Δx pela diferença entre x_R e x_S , obtidos em (B.40) e (B.03) respectivamente:

$$x_R = \frac{x_G \cdot z_P - x_P \cdot z_G}{z_P - z_G} \quad \text{e} \quad x_S = \frac{x_F \cdot z_P - z_F \cdot x_P}{z_P - z_F}$$

$$\Delta x = x_R - x_S$$

$$\Delta x = \frac{z_G \cdot x_R \cdot z_P - x_F \cdot z_G \cdot z_P - z_F \cdot x_R \cdot z_P + z_F \cdot x_G \cdot z_P}{z_P \cdot z_G - z_F \cdot z_G} \quad (\text{B.42})$$

B.6 – Determinação do Ponto P

A partir de x_R em (B.40), tem-se x_P :

$$x_R = \frac{x_G \cdot z_P - x_P \cdot z_G}{z_P - z_G}$$

$$x_P = \frac{z_G \cdot x_R - z_P \cdot x_R + z_P \cdot x_G}{z_G}$$

Da mesma maneira obtém-se y_P , agora através do y_R tido em (B.41):

$$y_R = \frac{z_P \cdot y_G - z_G \cdot y_P}{z_P - z_G}$$

$$y_P = \frac{z_G \cdot y_R - z_P \cdot y_R + z_P \cdot y_G}{z_G}$$

A partir do termo Δx obtido em (B.42), pode-se obter z_P :

$$\Delta x = \frac{z_G \cdot x_R \cdot z_P - x_F \cdot z_G \cdot z_P - z_F \cdot x_R \cdot z_P + z_F \cdot x_G \cdot z_P}{z_P \cdot z_G - z_F \cdot z_G}$$

$$z_P = \frac{\Delta x \cdot z_F \cdot z_G}{z_G \cdot x_F - x_G \cdot z_F + \Delta x \cdot z_G + z_F \cdot x_R - z_G \cdot x_R}$$

B.7 – Conversão de Sistema de Coordenadas

A conversão do sistema $Oxyz$ para $Hx'y'z'$ é desenvolvida abaixo. Estas transformações não são comuns no manuseio de imagens e tem seus resultados finais apresentados por Fu [39].

B.7.1 – Rotação

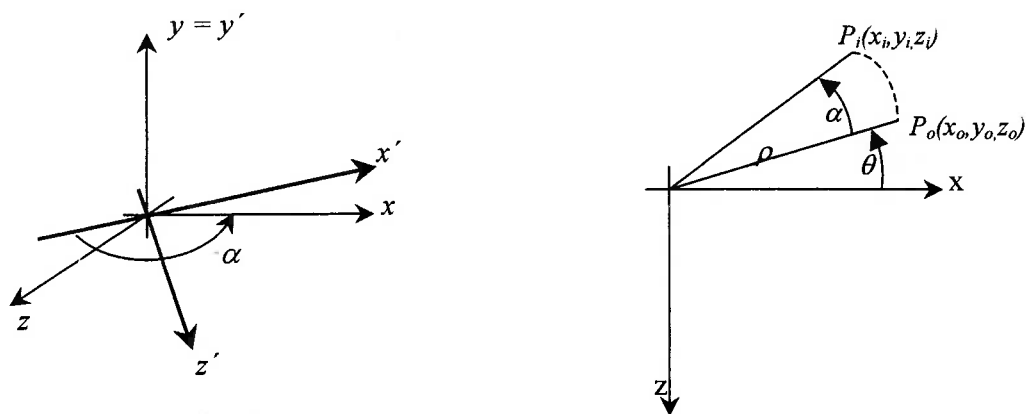


Figura B-2 – Representação da rotação do sistema de coordenadas.

Através da representação mostrada na figura B-2, verifica-se que:

$$x_i = \rho \cdot \cos(\theta + \alpha) = \rho \cdot \cos \theta \cdot \cos \alpha - \rho \cdot \sin \theta \cdot \sin \alpha$$

$$-z_i = \rho \cdot \sin(\theta + \alpha) = \rho \cdot \cos \theta \cdot \sin \alpha + \rho \cdot \sin \theta \cdot \cos \alpha$$

Sendo:

$$x_o = \rho \cdot \cos \theta \quad \text{e} \quad z_o = -\rho \cdot \sin \theta$$

Chega-se em:

$$x_i = x_o \cdot \cos \theta + z_o \cdot \sin \alpha$$

$$y_i = y_o$$

$$z_i = -x_o \cdot \sin \alpha + z_o \cdot \cos \alpha$$

Matricialmente obtém-se:

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} \cos \alpha & 0 & \text{sen } \alpha \\ 0 & 1 & 0 \\ -\text{sen } \alpha & 0 & \cos \alpha \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix}$$

B.7.2 – Translação

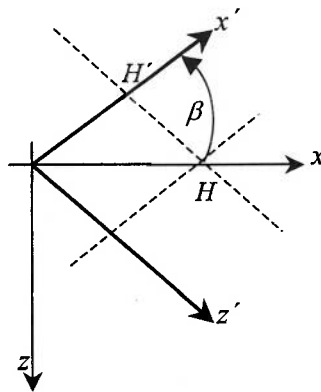


Figura B-3 – Representação da translação do sistema de coordenadas.

A figura B-3 mostra a representação da translação e permite verificar que:

$$z_{H'} = x_H \cdot \text{sen } \beta$$

$$x_{H'} = x_H \cdot \cos \beta$$

Matricialmente obtém-se:

$$\begin{bmatrix} x_H \cdot \cos \beta \\ 0 \\ x_H \cdot \text{sen } \beta \end{bmatrix}$$

Sendo $\beta = 180 - \alpha$, tem-se:

$$\begin{bmatrix} x_H \cdot \cos \beta \\ 0 \\ x_H \cdot \text{sen } \beta \end{bmatrix}$$

A transformação total fica:

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} + \begin{bmatrix} -x_H \cdot \cos \alpha \\ 0 \\ x_H \cdot \sin \alpha \end{bmatrix} \quad (\text{B.43})$$

B.8 – Determinação do Ponto Q'

No sistema $Hx'y'z'$ serão determinados os pontos apenas em relação às coordenadas x' e y' , já que se trabalha com a imagem no plano. Obtém-se o ponto Q' aplicando as equações da transformação (B.43):

$$x_{Q'} = x_Q \cdot \cos \alpha + z_Q \cdot \sin \alpha - x_H \cdot \cos \alpha \quad (\text{B.44})$$

$$y_{Q'} = y_Q \quad (\text{B.45})$$

Substituindo nas equações (B.44) e (B.45) os valores de x_Q , y_Q e z_Q obtidos em (B.16), (B.19) e (B.24) tem-se:

$$\begin{cases} x'_{Q'} = \frac{z_G \cdot x_H - z_G \cdot x_P + z_P \cdot x_G - z_P \cdot x_H}{z_P \cdot \cos \alpha - z_G \cdot \cos \alpha - x_P \cdot \sin \alpha + x_G \cdot \sin \alpha} \\ y'_{Q'} = \frac{y_G \cdot x_H \cdot \sin \alpha - y_P \cdot x_H \cdot \sin \alpha + y_G \cdot z_P \cdot \cos \alpha - y_P \cdot z_G \cdot \cos \alpha + y_P \cdot x_G \cdot \sin \alpha - y_G \cdot x_P \cdot \sin \alpha}{z_P \cdot \cos \alpha - z_G \cdot \cos \alpha - x_P \cdot \sin \alpha + x_G \cdot \sin \alpha} \end{cases}$$

B.9 – Determinação do Ponto T'

De maneira similar ao ponto Q' , o ponto T' é obtido aplicando-se as equações da transformação (B.43):

$$x_{T'} = x_T \cdot \cos \alpha + z_T \cdot \sin \alpha - x_H \cdot \cos \alpha \quad (\text{B.46})$$

$$y_{T'} = y_T \quad (\text{B.47})$$

Substituindo-se nas equações (B.46) e (B.47) os valores de x_T , y_T e z_T obtidos em (B.34), (B.39) e (B.32), chega-se a:

$$\left. \begin{aligned}
 x_T^f &= \frac{z_F \cdot x_P \cdot z_G - x_F \cdot z_P \cdot z_G + z_P \cdot z_G \cdot x_H - z_F \cdot z_G \cdot x_H}{z_F \cdot z_G \cdot \cos\alpha - z_P \cdot z_G \cdot \cos\alpha + z_F \cdot x_P \cdot \text{sen}\alpha - x_F \cdot z_P \cdot \text{sen}\alpha + z_P \cdot x_G \cdot \text{sen}\alpha - z_F \cdot x_G \cdot \text{sen}\alpha} \\
 y_T^f &= \frac{z_P \cdot y_F \cdot x_G \cdot \text{sen}\alpha - x_F \cdot z_P \cdot y_G \cdot \text{sen}\alpha - z_P \cdot y_F \cdot x_H \cdot \text{sen}\alpha - z_F \cdot y_G \cdot x_H \cdot \text{sen}\alpha}{z_F \cdot z_G \cdot \cos\alpha - z_P \cdot z_G \cdot \cos\alpha + z_F \cdot x_P \cdot \text{sen}\alpha - x_F \cdot z_P \cdot \text{sen}\alpha + z_P \cdot x_G \cdot \text{sen}\alpha - z_F \cdot x_G \cdot \text{sen}\alpha} + \\
 &\quad \frac{z_F \cdot x_P \cdot y_G \cdot \text{sen}\alpha - z_F \cdot y_P \cdot x_G \cdot \text{sen}\alpha + z_F \cdot y_P \cdot z_G \cdot \cos\alpha + z_F \cdot y_P \cdot x_H \cdot \text{sen}\alpha}{z_F \cdot z_G \cdot \cos\alpha - z_P \cdot z_G \cdot \cos\alpha + z_F \cdot x_P \cdot \text{sen}\alpha - x_F \cdot z_P \cdot \text{sen}\alpha + z_P \cdot x_G \cdot \text{sen}\alpha - z_F \cdot x_G \cdot \text{sen}\alpha} + \\
 &\quad \frac{z_P \cdot y_G \cdot x_H \cdot \text{sen}\alpha - z_P \cdot y_F \cdot z_G \cdot \cos\alpha}{z_F \cdot z_G \cdot \cos\alpha - z_P \cdot z_G \cdot \cos\alpha + z_F \cdot x_P \cdot \text{sen}\alpha - x_F \cdot z_P \cdot \text{sen}\alpha + z_P \cdot x_G \cdot \text{sen}\alpha - z_F \cdot x_G \cdot \text{sen}\alpha}
 \end{aligned} \right\}$$

ANEXO C – CÓDIGO FONTE

Disponibiliza-se nesta seção o código fonte do software utilizado para o tratamento das imagens. O software é composto de diferentes módulos, um para cada etapa no tratamento das imagens. Os diferentes módulos foram feitos a fim de se verificar o resultado após cada etapa de tratamento das imagens. Os códigos de cada módulo são apresentados abaixo na ordem seqüencial de uso.

C.1 – Detecção de Vértices e Simplificação da Triangulação

```
#include <iostream>
#include <fstream>
#include <string>
#include <memory>
#include "biblioteca.cpp"

using namespace std;

const double intensidade = 0.1;

void erro()
{
    cout << "=====" << endl;
    cout << "  Vertice & Simplificacao  " << endl;
    cout << "=====" << endl << endl;
    cout << "Uso: BORDA <nome do arquivo de opcoes> ou" << endl;
    cout << "      BORDA -c (criar arquivo de opcoes)" << endl;
    exit(1);
}

void criar_arquivo()
{
    fstream ofs("borda.opc", ios::out);

    ofs << "#####" << endl;
    ofs << "#      Opcoes de Borda      #" << endl;
    ofs << "#####" << endl << endl;

    ofs << "# Arquivo de entrada" << endl;
    ofs << "entrada.bmp" << endl << endl;

    ofs << "# Imagem de Saida (Largura & Altura)" << endl;
    ofs << "800" << endl;
    ofs << "600" << endl << endl;

    ofs << "# Arquivo de saida" << endl;
```



```

    ofs << "saida.bmp" << endl << endl;

    exit(0);
}

void opcoes(string& opc, string& ifn, int& w, int& h, string& ofn)
{
    ifstream ifs(opc.c_str(), ios::in);
    if(!ifs) erro();

    string aux;

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) ifn = aux;
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) w = atoi(aux.c_str());
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) h = atoi(aux.c_str());
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) ofn = aux;
    else erro();

    if(w<0) erro();
    if(h<0) erro();
}

double diagonal(imagef& img)
{
    double hipot = (img.w()*img.w()+img.h()*img.h());
    double hipoten = sqrt (hipot);
    return hipoten;
}

double distancia(int x1, int y1, int x2, int y2)
{
    double artif = ((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
    double artific = sqrt (artif);
    return artific;
}

void borda(int& xad, int& yad, int& xbd, int& ybd, int& xcd, int&
ycd, int& xed, int& yed, imagef& img)
{
    double xi = (double)(img.w()-1)/2;

```

```

double yi = (double)(img.h()-1)/2;

double d, dmin = diagonal(img);

for(int i=0; i<=(int)xi; ++i)
    for(int j=0; j<=(int)yi; ++j)
        if(img(i,j)<intensidade)
            {
                d = distancia(0, 0, i,j);

                if(dmin>d)
                    {
                        dmin = d;
                        xad = i;
                        yad = j;
                    }
            }

dmin = diagonal(img);

for(int i=(int)(xi+1); i<img.w(); ++i)
    for(int j=0; j<=(int)yi; ++j)
        if(img(i,j)<intensidade)
            {
                d = distancia((img.w()-1), 0, i,j);

                if(dmin>d)
                    {
                        dmin = d;
                        xbd = i;
                        ybd = j;
                    }
            }

dmin = diagonal(img);

for(int i=0; i<=(int)xi; ++i)
    for(int j=(int)(yi+1); j<img.h(); ++j)
        if(img(i,j)<intensidade)
            {
                d = distancia(0, (img.h()-1), i,j);

                if(dmin>d)
                    {
                        dmin = d;
                        xcd = i;
                        ycd = j;
                    }
            }

dmin = diagonal(img);

for(int i=(int)(xi+1); i<img.w(); ++i)
    for(int j=(int)(yi+1); j<img.h(); ++j)
        if(img(i,j)<intensidade)
            {
                d = distancia((img.w()-1), (img.h()-1),
i,j);

```

```
        if(dmin>d)
        {
            dmin = d;
            xed = i;
            yed = j;
        }
    }

void alinhamento(imagef& iimg, int xad, int yad, int xbd, int ybd,
int xcd, int ycd, int xed, int yed, imagef& oimg)
{
    double riv, rkx, rivd, rkxd;

    double acd = distancia(xad, yad, xcd, ycd);
    double bed = distancia(xbd, ybd, xed, yed);

    double xid, yid, xjd, yjd, xvd, yvd;

    for(int i=0; i<oimg.w(); ++i)
        for(int j=0; j<oimg.h(); ++j)
        {
            riv = (double)i/(oimg.w()-1);
            rkx = (double)j/(oimg.h()-1);

            rivd = -1*riv*acd/(-1*riv*acd+bed*riv-bed);
            rkxd = rkx;

            xid = xad+rkxd*xcd-rkxd*xad;
            yid = yad+rkxd*ycd-rkxd*yad;

            xjd = xbd+rkxd*xed-rkxd*xbd;
            yjd = ybd+rkxd*yed-rkxd*ybd;

            xvd = xid+rivd*xjd-rivd*xid;
            yvd = yid+rivd*yjd-rivd*yid;

            oimg(i,j) = iimg(xvd,yvd);
        }
}

int main(int argc, char* argv[])
{
    if(argc!=2) erro();

    string opc = argv[1];

    if(opc=="-c") criar_arquivo();

    string ifn;
    int w, h;
    string ofn;

    opcoes(opc, ifn, w, h, ofn);

    imagef iimg;

    try
```

```

    {
        loadbmp(ifn, iimg);
    }
    catch(bad_file)
    {
        erro();
    }

    int xad, yad, xbd, ybd, xcd, ycd, xed, yed;

    borda(xad, yad, xbd, ybd, xcd, ycd, xed, yed, iimg);

    imagef oimg(w,h);

    alinhamento(iimg, xad, yad, xbd, ybd, xcd, ycd, xed, yed,
oimg);

    try
    {
        savebmp(oimg, ofn);
    }
    catch(bad_file)
    {
        erro();
    }

    return 0;
}

```

C.2 – Filtro

```

#include <iostream>
#include <fstream>
#include <string>
#include <cstdlib>
#include "biblioteca.cpp"

using namespace std;

void erro()
{
    cout << "=====" << endl;
    cout << "          Filtro          " << endl;
    cout << "=====" << endl << endl;
    cout << "Uso: FILTRO <nome do arquivo de opcoes> ou" << endl;
    cout << "          FILTRO -c (criar arquivo de opcoes)" << endl;
    exit(1);
}

void criar_arquivo()
{
    ofstream ofs("filtro.opc", ios::out);

    ofs << "#####" << endl;
    ofs << "#          Opções de Filtro          #" << endl;
}

```

```
ofs << "#####" << endl << endl;

ofs << "# Arquivo de entrada" << endl;
ofs << "entrada.bmp" << endl << endl;

ofs << "# Imagem de Saida (Largura & Altura)" << endl;
ofs << "800" << endl;
ofs << "600" << endl << endl;

ofs << "# Filtro Limite" << endl;
ofs << "0.25" << endl << endl;

ofs << "# Arquivo de saida" << endl;
ofs << "filtrado.bmp" << endl << endl;

exit(0);
}

void opcoes(string& opc, string& ifn, int& w, int& h, double& e,
string& ofn)
{
    fstream ifs(opc.c_str(), ios::in);
    if(!ifs) erro();

    string aux;

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&!ifs.eof()) getline(ifs,
aux);
    if(!ifs.eof()) ifn = aux;
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&!ifs.eof()) getline(ifs,
aux);
    if(!ifs.eof()) w = atoi(aux.c_str());
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&!ifs.eof()) getline(ifs,
aux);
    if(!ifs.eof()) h = atoi(aux.c_str());
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&!ifs.eof()) getline(ifs,
aux);
    if(!ifs.eof()) e = atof(aux.c_str());
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&!ifs.eof()) getline(ifs,
aux);
    if(!ifs.eof()) ofn = aux;
    else erro();

    if(w<0) erro();
    if(h<0) erro();
    if(e<0) erro();
}
```

```
}  
  
void filtra(imagef& iimg, imagef& oimg, double e)  
{  
    for(int j=0; j<iimg.h(); ++j)  
    {  
        for(int i=0; i<iimg.w(); ++i)  
        {  
            if(iimg(i,j)<e)  
            {  
                oimg(i,j)=0;  
            }  
            else  
                oimg(i,j)=1;  
        }  
    }  
}  
  
int main(int argc, char* argv[])  
{  
    if(argc!=2) erro();  
  
    string opc = argv[1];  
  
    if(opc=="-c") criar_arquivo();  
  
    string ifn;  
    int w, h;  
    double e;  
    string ofn;  
  
    opcoes(opc, ifn, w, h, e, ofn);  
  
    imagef iimg;  
  
    try  
    {  
        loadbmp(ifn, iimg);  
    }  
    catch(bad_file)  
    {  
        erro();  
    }  
  
    imagef oimg(iimg.w(),iimg.h());  
  
    filtra(iimg, oimg, e);  
  
    try  
    {  
        savebmp(oimg, ofn);  
    }  
    catch(bad_file)  
    {  
        erro();  
    }  
  
    return 0;  
}
```

C.3 – Simulação do Paralelepípedo

```

#include <iostream>
#include <fstream>
#include <string>
#include <cstdlib>
#include "biblioteca.cpp"

using namespace std;

void erro()
{
    cout << "=====" << endl;
    cout << " Gerador de Bloco " << endl;
    cout << "=====" << endl << endl;
    cout << "Uso: BLOCO <nome do arquivo de opcoes> ou" << endl;
    cout << "      BLOCO -c (criar arquivo de opcoes)" << endl;
    exit(1);
}

void criar_arquivo()
{
    fstream ofs("bloco.opc", ios::out);

    ofs << "#####" << endl;
    ofs << "#      Opcoes de Bloco      #" << endl;
    ofs << "#####" << endl << endl;

    ofs << "# Tamanho da Imagem" << endl;
    ofs << "800" << endl;
    ofs << "600" << endl << endl;

    ofs << "# Tamanho do Bloco (rel de comprimento)" << endl;
    ofs << "0.5" << endl << endl;

    ofs << "# Tamanho do Bloco (rel de largura)" << endl;
    ofs << "0.5" << endl << endl;

    ofs << "# Arquivo de Saida" << endl;
    ofs << "bloco.bmp" << endl;

    exit(0);
}

void opcoes(string& opc, int& w, int& h, double& wr, double& hr,
string& bfn)
{
    fstream ifs(opc.c_str(), ios::in);
    if(!ifs) erro();

    string aux;

    getline(ifs, aux);
    while((aux[0]!='#')||(aux==""))&&(!ifs.eof()) getline(ifs,
aux);
    if(!ifs.eof()) w = atoi(aux.c_str());
    else erro();
}

```

```

        getline(ifs, aux);
        while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
        if(!ifs.eof()) h = atoi(aux.c_str());
        else erro();

        getline(ifs, aux);
        while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
        if(!ifs.eof()) wr = atof(aux.c_str());
        else erro();

        getline(ifs, aux);
        while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
        if(!ifs.eof()) hr = atof(aux.c_str());
        else erro();

        getline(ifs, aux);
        while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
        if(!ifs.eof()) bfn = aux;
        else erro();

        if(w<0) erro();
        if(h<0) erro();
        if((wr<0)||(wr>1)) erro();
        if((hr<0)||(hr>1)) erro();
    }

void bloco(double wr, double hr, imagef& img)
{
    for(int k=0; k<img.s(); ++k)
        img(k) = 0;

    int w = (int)(img.w()*wr+0.5);
    int h = (int)(img.h()*hr+0.5);

    int ic = (img.w()-w)/2;
    int il = (img.h()-h)/2;

    for(int i=ic; i<(ic+w); ++i)
        for(int j=il; j<(il+h); ++j)
            img(i,j) = 1;
}

int main(int argc, char* argv[])
{
    if(argc!=2) erro();

    string opc = argv[1];

    if(opc=="-c") criar_arquivo();

    int w, h;
    double wr, hr;
    string ffn, bfn;

```



```

opcoes(opc, w, h, wr, hr, bfn);

imagef img(w,h);

bloco(wr, hr, img);

try
{
    savebmp(img, bfn);
}
catch(bad_file)
{
    erro();
}

return 0;
}

```

C.4 – Simulação de Projeção

```

#include <iostream>
#include <fstream>
#include <string>
#include <cstdlib>
#include "biblioteca.cpp"

using namespace std;

void erro()
{
    cout << "=====" << endl;
    cout << "Projeto de Estampa" << endl;
    cout << "=====" << endl << endl;
    cout << "Uso: PROJETA <nome do arquivo de opcoes> or" << endl;
    cout << "      PROJETA -c (cria arquivo de opcoes)" << endl;
    exit(1);
}

void criar_arquivo()
{
    ofstream ofs("projeta.opc", ios::out);

    ofs << "#####" << endl;
    ofs << "# Opcoes de Projecao #" << endl;
    ofs << "#####" << endl << endl;

    ofs << "# Arquivo de relevo" << endl;
    ofs << "bloco.bmp" << endl << endl;

    ofs << "# Arquivo de Estampa" << endl;
    ofs << "grade.bmp" << endl << endl;

    ofs << "# Ponto Focal (Direcao Z)" << endl;
    ofs << "900" << endl << endl;

    ofs << "# Altura do Bloco" << endl;
}

```

```
    ofs << "100" << endl << endl;

    ofs << "# Arquivo de Saida" << endl;
    ofs << "projeta.bmp" << endl;

    exit(0);
}

void opcoes(string& opc, string& rfn, string& pfn, double& zf,
double& h, string& ofn)
{
    fstream ifs(opc.c_str(), ios::in);
    if(!ifs) erro();

    string aux;

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) rfn = aux;
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) pfn = aux;
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) zf = atof(aux.c_str());
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) h = atof(aux.c_str());
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) ofn = aux;
    else erro();

    if(zf<0) erro();
}

void projeta(imagef& ring, imagef& pimg, double zf, double h,
imagef& oimg)
{
    double xf = (double) (pimg.w()-1)/2;
    double yf = (double) (pimg.h()-1)/2;

    double zp;
    int xs, ys;

    for(int xp=(int)xf; xp>=0; --xp)
```

```

    {
        for(int yp=(int)yf; yp>=0; --yp)
        {
            zp = rimg(xp,yp)*h;

            xs = (int)((xf*zp-zf*xp)/(zp-zf)+0.5);
            ys = (int)((yf*zp-zf*yp)/(zp-zf)+0.5);

            oimg(xp,yp) = pimg(xs,ys);
        }

        for(int yp=(int)(yf+1); yp<pimg.h(); ++yp)
        {
            zp = rimg(xp,yp)*h;

            xs = (int)((xf*zp-zf*xp)/(zp-zf)+0.5);
            ys = (int)((yf*zp-zf*yp)/(zp-zf)+0.5);

            oimg(xp,yp) = pimg(xs,ys);
        }
    }

    for(int xp=(int)(xf+1); xp<pimg.w(); ++xp)
    {
        for(int yp=(int)yf; yp>=0; --yp)
        {
            zp = rimg(xp,yp)*h;

            xs = (int)((xf*zp-zf*xp)/(zp-zf)+0.5);
            ys = (int)((yf*zp-zf*yp)/(zp-zf)+0.5);

            oimg(xp,yp) = pimg(xs,ys);
        }

        for(int yp=(int)(yf+1); yp<pimg.h(); ++yp)
        {
            zp = rimg(xp,yp)*h;

            xs = (int)((xf*zp-zf*xp)/(zp-zf)+0.5);
            ys = (int)((yf*zp-zf*yp)/(zp-zf)+0.5);

            oimg(xp,yp) = pimg(xs,ys);
        }
    }
}

int main(int argc, char* argv[])
{
    if(argc!=2) erro();

    string opc = argv[1];

    if(opc=="-c") criar_arquivo();

    string rfn, pfn;
    double zf, h;
    string ofn;

    opcoes(opc, rfn, pfn, zf, h, ofn);
}

```

```
imagef rimg;
imagef pimg;

try
{
    loadbmp(rfn, rimg);
    loadbmp(pfn, pimg);
}
catch(bad_file)
{
    erro();
}

if(rimg.w()!=pimg.w()) erro();
if(rimg.h()!=pimg.h()) erro();

imagef oimg(pimg.w(),pimg.h());

projeta(rimg, pimg, zf, h, oimg);

try
{
    savebmp(oimg, ofn);
}
catch(bad_file)
{
    erro();
}

return 0;
}
```

C.5 – Simulação de Captura

```
#include <iostream>
#include <fstream>
#include <string>
#include <cstdlib>
#include <cmath>
#include "biblioteca.cpp"

using namespace std;

const long double pi = 3.141592654;

void erro()
{
    cout << "=====" << endl;
    cout << "Captura de Projecao" << endl;
    cout << "=====" << endl << endl;
    cout << "Uso: CAPTURA <nome do arquivo de opcoes> or" << endl;
    cout << "      CAPTURA -c (criar arquivo de opcoes)" << endl;
    exit(1);
}
```

```

}

void criar_arquivo()
{
    ofstream ofs("captura.opc", ios::out);

    ofs << "#####" << endl;
    ofs << "#          Opcoes de Captura          #" << endl;
    ofs << "#####" << endl << endl;

    ofs << "# Arquivo de Relevô" << endl;
    ofs << "bloco.bmp" << endl << endl;

    ofs << "# Arquivo de Estampa Projetada" << endl;
    ofs << "projeta.bmp" << endl << endl;

    ofs << "# Altura do bloco" << endl;
    ofs << "100" << endl << endl;

    ofs << "# Plano de Captura - Angulo & Posicao (Direção x)" <<
endl;
    ofs << "-15" << endl;
    ofs << "650" << endl << endl;

    ofs << "# Posicao do Ponto Focal (Direcao x & z)" << endl;
    ofs << "650" << endl;
    ofs << "950" << endl << endl;

    ofs << "# Output file name" << endl;
    ofs << "captura.bmp" << endl;

    exit(0);
}

void opcoes(string& opc, string& rfn, string& pfn, double& l,
double& a, double& xh, double& xg, double& zg, string& ofn)
{
    ifstream ifs(opc.c_str(), ios::in);
    if(!ifs) erro();

    string aux;

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) rfn = aux;
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) pfn = aux;
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) l = atof(aux.c_str());
    else erro();
}

```

```

    getline(ifs, aux);
    while(((aux[0]=='#')||(aux==""))&&!ifs.eof()) getline(ifs,
aux);
    if(!ifs.eof()) a = atof(aux.c_str());
    else erro();

    getline(ifs, aux);
    while(((aux[0]=='#')||(aux==""))&&!ifs.eof()) getline(ifs,
aux);
    if(!ifs.eof()) xh = atof(aux.c_str());
    else erro();

    getline(ifs, aux);
    while(((aux[0]=='#')||(aux==""))&&!ifs.eof()) getline(ifs,
aux);
    if(!ifs.eof()) xg = atof(aux.c_str());
    else erro();

    getline(ifs, aux);
    while(((aux[0]=='#')||(aux==""))&&!ifs.eof()) getline(ifs,
aux);
    if(!ifs.eof()) zg = atof(aux.c_str());
    else erro();

    getline(ifs, aux);
    while(((aux[0]=='#')||(aux==""))&&!ifs.eof()) getline(ifs,
aux);
    if(!ifs.eof()) ofn = aux;
    else erro();

    if(zg<0) erro();
}

void tamanho_imagem(int& dx, int& dy, int& w, int& h, imagef& ring,
double a, double xh, double xg, double zg)
{
    double yg = (double)(ring.h()-1)/2;
    double ar = a/180*pi;
    double sa = sin(ar);
    double ca = cos(ar);

    int xmin, ymin, xmax, ymax;

    double xp = 0;
    double yp = 0;
    double zp = 0;
    int xqd = (int)((-1*zg*xp+zp*xg-zp*xh+zg*xh)/(-
1*sa*xp+sa*xg+ca*zp-ca*zg)+0.5);
    int yqd = (int)((-1*yg*sa*xp+yg*ca*zp+yp*sa*xg-yp*ca*zg-
yp*sa*xh+yg*sa*xh)/(-1*sa*xp+sa*xg+ca*zp-ca*zg)+0.5);
    xmin = xqd;
    xmax = xqd;
    ymin = yqd;
    ymax = yqd;

    xp = ring.w()-1;
    yp = 0;

```

```

        xqd = (int)((-1*zg*xp+zp*xg-zp*xh+zg*xh)/(-
1*sa*xp+sa*xg+ca*zp-ca*zg)+0.5);
        yqd = (int)((-1*yg*sa*xp+yg*ca*zp+yp*sa*xg-yp*ca*zg-
yp*sa*xh+yg*sa*xh)/(-1*sa*xp+sa*xg+ca*zp-ca*zg)+0.5);
        if(xmin>xqd) xmin = xqd;
        if(xmax<xqd) xmax = xqd;
        if(ymin>yqd) ymin = yqd;
        if(ymax<yqd) ymax = yqd;

        xp = 0;
        yp = ring.h()-1;
        xqd = (int)((-1*zg*xp+zp*xg-zp*xh+zg*xh)/(-
1*sa*xp+sa*xg+ca*zp-ca*zg)+0.5);
        yqd = (int)((-1*yg*sa*xp+yg*ca*zp+yp*sa*xg-yp*ca*zg-
yp*sa*xh+yg*sa*xh)/(-1*sa*xp+sa*xg+ca*zp-ca*zg)+0.5);
        if(xmin>xqd) xmin = xqd;
        if(xmax<xqd) xmax = xqd;
        if(ymin>yqd) ymin = yqd;
        if(ymax<yqd) ymax = yqd;

        xp = ring.w()-1;
        yp = ring.h()-1;
        xqd = (int)((-1*zg*xp+zp*xg-zp*xh+zg*xh)/(-
1*sa*xp+sa*xg+ca*zp-ca*zg)+0.5);
        yqd = (int)((-1*yg*sa*xp+yg*ca*zp+yp*sa*xg-yp*ca*zg-
yp*sa*xh+yg*sa*xh)/(-1*sa*xp+sa*xg+ca*zp-ca*zg)+0.5);
        if(xmin>xqd) xmin = xqd;
        if(xmax<xqd) xmax = xqd;
        if(ymin>yqd) ymin = yqd;
        if(ymax<yqd) ymax = yqd;

        dx = -1*xmin;
        dy = -1*ymin;

        w = xmax-xmin+1;
        h = ymax-ymin+1;
    }

void captura(imagef& ring, imagef& pimg, double& l, double& a,
double& xh, double& xg, double& zg, imagef& oimg, int& dx, int& dy)
{
    double yg = (double)(ring.h()-1)/2;
    double ar = a/180*pi;
    double sa = sin(ar);
    double ca = cos(ar);

    for(int k=0; k<oimg.s(); ++k)
        oimg(k) = 0;

    double zp;
    int xqd, yqd;

    for(int xp=0; xp<=(int)xg; ++xp)
    {
        for(int yp=0; yp<=(int)yg; ++yp)
        {
            zp = ring(xp,yp)*l;

```

```

        xqd = (int)((-1*zg*xp+zp*xg-zp*xh+zg*xh)/(-
1*sa*xp+sa*xg+ca*zp-ca*zg)+0.5);
        yqd = (int)((-1*yg*sa*xp+yg*ca*zp+yp*sa*xg-
yp*ca*zg-yp*sa*xh+yg*sa*xh)/(-1*sa*xp+sa*xg+ca*zp-ca*zg)+0.5);

        xqd += dx;
        yqd += dy;

        oimg(xqd,yqd) = pimg(xp,yp);
        oimg(xqd+1,yqd) = pimg(xp,yp);
        oimg(xqd,yqd+1) = pimg(xp,yp);
        oimg(xqd+1,yqd+1) = pimg(xp,yp);
    }

    for(int yp=(int)(rimg.h()-1); yp>(int)yg; --yp)
    {
        zp = rimg(xp,yp)*1;

        xqd = (int)((-1*zg*xp+zp*xg-zp*xh+zg*xh)/(-
1*sa*xp+sa*xg+ca*zp-ca*zg)+0.5);
        yqd = (int)((-1*yg*sa*xp+yg*ca*zp+yp*sa*xg-
yp*ca*zg-yp*sa*xh+yg*sa*xh)/(-1*sa*xp+sa*xg+ca*zp-ca*zg)+0.5);

        xqd += dx;
        yqd += dy;

        oimg(xqd,yqd) = pimg(xp,yp);
        oimg(xqd+1,yqd) = pimg(xp,yp);
        oimg(xqd,yqd-1) = pimg(xp,yp);
        oimg(xqd+1,yqd-1) = pimg(xp,yp);
    }
}

for(int xp=(rimg.w()-1); xp>(int)xg; --xp)
{
    for(int yp=0; yp<=(int)yg; ++yp)
    {
        zp = rimg(xp,yp)*1;

        xqd = (int)((-1*zg*xp+zp*xg-zp*xh+zg*xh)/(-
1*sa*xp+sa*xg+ca*zp-ca*zg)+0.5);
        yqd = (int)((-1*yg*sa*xp+yg*ca*zp+yp*sa*xg-
yp*ca*zg-yp*sa*xh+yg*sa*xh)/(-1*sa*xp+sa*xg+ca*zp-ca*zg)+0.5);

        xqd += dx;
        yqd += dy;

        oimg(xqd,yqd) = pimg(xp,yp);
        oimg(xqd-1,yqd) = pimg(xp,yp);
        oimg(xqd,yqd+1) = pimg(xp,yp);
        oimg(xqd-1,yqd+1) = pimg(xp,yp);
    }

    for(int yp=(rimg.h()-1); yp>(int)yg; --yp)
    {
        zp = rimg(xp,yp)*1;

        xqd = (int)((-1*zg*xp+zp*xg-zp*xh+zg*xh)/(-
1*sa*xp+sa*xg+ca*zp-ca*zg)+0.5);

```



```
        yqd = (int)((-1*yg*sa*xp+yg*ca*zp+yp*sa*xg-
yp*ca*zg-yp*sa*xh+yg*sa*xh)/(-1*sa*xp+sa*xg+ca*zp-ca*zg)+0.5);

        xqd += dx;
        yqd += dy;

        oimg(xqd,yqd) = pimg(xp,yp);
        oimg(xqd-1,yqd) = pimg(xp,yp);
        oimg(xqd,yqd-1) = pimg(xp,yp);
        oimg(xqd-1,yqd-1) = pimg(xp,yp);
    }
}

int main(int argc, char* argv[])
{
    if(argc!=2) erro();

    string opc = argv[1];

    if(opc=="-c") criar_arquivo();

    string rfn, pfn;
    double l, a, xh;
    double xg, zg;
    string ofn;

    opcoes(opc, rfn, pfn, l, a, xh, xg, zg, ofn);

    imagef ring;
    imagef pimg;

    try
    {
        loadbmp(rfn, ring);
        loadbmp(pfn, pimg);
    }
    catch(bad_file)
    {
        erro();
    }

    if(ring.w()!=pimg.w()) erro();
    if(ring.h()!=pimg.h()) erro();

    int dx, dy, w, h;

    tamanho_imagem(dx, dy, w, h, ring, a, xh, xg, zg);

    imagef oimg(w,h);

    captura(ring, pimg, l, a, xh, xg, zg, oimg, dx, dy);

    try
    {
        savebmp(oimg, ofn);
    }
    catch(bad_file)
    {
```

```

        erro();
    }

    return 0;
}

```

C.6 – Calibração

```

#include <iostream>
#include <fstream>
#include <string>
#include <cstdlib>
#include <ctime>
#include <cmath>
#include <math.h>
#include "biblioteca.cpp"

using namespace std;

void erro()
{
    cout << "=====" << endl;
    cout << "      Calibracao      " << endl;
    cout << "=====" << endl << endl;
    cout << "Uso: CALIBRA <nome do arquivo de opcoes> ou" << endl;
    cout << "      CALIBRA -c (criar arquivo de opcoes)" << endl;
    exit(1);
}

void criar_arquivo()
{
    ofstream ofs("calibra.opc", ios::out);

    ofs << "#####" << endl;
    ofs << "#      Opcoes de Calibracao      #" << endl;
    ofs << "#####" << endl <<
endl;

    ofs << "# Arquivo de Calibracao" << endl;
    ofs << "bloco.bmp" << endl << endl;

    ofs << "# Altura do bloco" << endl;
    ofs << "100" << endl << endl;

    ofs << "# Arquivo de Referencia" << endl;
    ofs << "referencia.bmp" << endl << endl;

    ofs << "# Area de Calibracao (relacao de comprimento da
imagem)" << endl;
    ofs << "0.4" << endl << endl;

    ofs << "# Area de Calibracao (relacao de altura da imagem)" <<
endl;
}

```

```
ofs << "0.4" << endl << endl;

ofs << "# Posicao da Area de Calibracao (relacao na direcao
x)" << endl;
ofs << "0.27" << endl << endl;

ofs << "# Arquivo de Saida" << endl;
ofs << "calibra.clb" << endl << endl;

ofs << "# Verbose" << endl;
ofs << "0" << endl;

exit(0);
}

void opcoes(string& opc, string& cfn, double& bh, string& rfn,
double& wr, double& hr, double& pr, string& ofn, bool& v)
{
    fstream ifs(opc.c_str(), ios::in);
    if(!ifs) erro();

    string aux;

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) cfn = aux;
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) bh = atof(aux.c_str());
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) rfn = aux;
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) wr = atof(aux.c_str());
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) hr = atof(aux.c_str());
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) pr = atof(aux.c_str());
    else erro();
}
```

```

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) ofn = aux;
    else erro();

    getline(ifs, aux);
    while(((aux[0]!='#')||(aux==""))&&(!ifs.eof())) getline(ifs,
aux);
    if(!ifs.eof()) v = atoi(aux.c_str());
    else erro();

    if(bh<0) erro();
    if((wr<0)||(wr>1)) erro();
    if((hr<0)||(hr>1)) erro();
    if((pr<0)||(pr>1)) erro();
}

void set_sons(double* zfs, double* xgs, double* zgs, double st)
{
    int p = 1;

    for(int i=-1; i<=1; ++i)
        for(int j=-1; j<=1; ++j)
            for(int k=-1; k<=1; ++k)
                if((i!=0)||(j!=0)||(k!=0))
                    {
                        zfs[p] = zfs[0]+i*st;
                        xgs[p] = xgs[0]+j*st;
                        zgs[p] = zgs[0]+k*st;
                        ++p;
                    }
}

double get_ae(imagef& vimg, imagef& ximg, double bh, int xs, int ys,
int xe, int ye, double zf, double xg, double zg)
{
    double xf = (double)(ximg.w()-1)/2;
    double yg = (double)(ximg.h()-1)/2;

    int dx;
    int lf=1;
    double xp, yp, zp, bx, by, e, ae = 0;

    for(int xr=xs; xr<xe; ++xr)
        for(int yr; yr<ye; ++yr)
            {
                int xo = xr;
                int yo = yr;
                double dh = vimg(xr,yr)-ximg(xo,yo);
                dx = 0;

                if(dh<0)
                    {
                        while((dh<0)&&(xo<ximg.w()))
                            {
                                ++xo;

```

```

                dh = vimg(xr,yr)-ximg(xo,yo);
            }
            lf=lf-1;
            dx = (xr-xo)+lf;
        }

        else
        {
            dx = 0;
            lf=1;
        }

        zp = (-1*dx*zg*zf)/(-1*dx*zg-zg*xf+xr*zg+xg*zf-zf*xr);
        xp = (-1*xr*zp+xr*zg+zp*xg)/zg;
        yp = (-1*yr*zp+yr*zg+yg*zp)/zg;

        bx = (-1*xr*bh+xr*zg+bh*xg)/zg;
        by = (-1*yr*bh+yr*zg+yg*bh)/zg;

        e= (bh-zp)*(bh-zp)+(bx-xp)*(bx-xp)+(by-yp)*(by-yp);

        ae += e;
    }

    return ae;
}

double diagonal(imagef& img)
{
    double hipot = (img.w()*img.w()+img.h()*img.h());
    double hipoten = sqrt (hipot);
    return hipoten;
}

void calibracao(imagef& vimg, imagef& ximg, double bh, double wr,
double hr, double pr, double& zf, double& xg, double& zg, bool v)
{
    cout.setf(ios::fixed, ios::floatfield);

    int ws = (int)(wr*vimg.w());
    int hs = (int)(hr*vimg.h());
    int xs = (int)(pr*vimg.w())-ws/2;;
    int ys = (vimg.h()-hs)/2;
    int xe = xs+ws-1;
    int ye = ys+hs-1;

    double* zfs = new double[27];
    double* xgs = new double[27];
    double* zgs = new double[27];
    double* aes = new double[27];

    zfs[0] = 1490.9;
    xgs[0] = 1041.8;
    zgs[0] = 2000;

    aes[0] = get_ae(vimg, ximg, bh, xs, ys, xe, ye, zfs[0],
xgs[0], zgs[0]);
}

```

```
double st = 29.6;
double aemin;
int kmin, t = 1;

while((st>=1)&&(t<=10000))
{
    set_sons(zfs, xgs, zgs, st);

    if(v)
    {
        cout << "(" << t << "x - " << st << ")" << endl;
        cout << "0 -> " << aes[0] << " " << zfs[0] << " "
<< xgs[0] << " " << zgs[0] << endl;
    }

    for(int k=1; k<27; ++k)
    {
        aes[k] = get_ae(vimg, ximg, bh, xs, ys, xe, ye,
zfs[k], xgs[k], zgs[k]);

        if(v) cout << k << " -> " << aes[k] << " " <<
zfs[k] << " " << xgs[k] << " " << zgs[k] << endl;
    }

    aemin = aes[0];
    kmin = 0;

    for(int k=0; k<27; ++k)
        if(aemin>aes[k])
        {
            aemin = aes[k];
            kmin = k;
        }

    if(v)
    {
        cout << endl;
        cout << kmin << " -> " << aes[kmin] << " " <<
zfs[kmin] << " " << xgs[kmin] << " " << zgs[kmin] << endl;
        cout << "(" << t << "x - " << st << ")" << endl;
        cout << endl;
    }

    if(kmin==0)
    {
        st /= 2;
        t = 1;
    }
    else
    {
        zfs[0] = zfs[kmin];
        xgs[0] = xgs[kmin];
        zgs[0] = zgs[kmin];
        aes[0] = aes[kmin];
        ++t;
    }
}
```

```
    zf = zfs[0];
    xg = xgs[0];
    zg = zgs[0];
}

void save(string& fn, double zf, double xg, double zg)
{
    ofstream ofs(fn.c_str(), ios::out);
    if(!ofs) erro();

    ofs.setf(ios::fixed, ios::floatfield);

    ofs << "# Posicao Z do Projetor" << endl;
    ofs << zf << endl << endl;

    ofs << "# Posicao X da Camera" << endl;
    ofs << xg << endl << endl;

    ofs << "# Posicao Z da Camera" << endl;
    ofs << zg << endl;
}

int main(int argc, char* argv[])
{
    int itime = time(0);

    if(argc!=2) erro();

    string opc = argv[1];

    if(opc=="-c") criar_arquivo();

    string cfn, rfn;
    double bh, wr, hr, pr;
    string ofn;
    bool v;

    opcoes(opc, cfn, bh, rfn, wr, hr, pr, ofn, v);

    imagef vimg, ximg;

    try
    {
        loadbmp(cfn, vimg);
        loadbmp(rfn, ximg);
    }
    catch(bad_file)
    {
        erro();
    }

    if(vimg.w()!=ximg.w()) erro();
    if(vimg.h()!=ximg.h()) erro();

    double zf, xg, zg;

    calibracao(vimg, ximg, bh, wr, hr, pr, zf, xg, zg, v);
}
```

```

    save(ofn, zf, xg, zg);

    if(v)
    {
        cout << endl;
        cout << "Elapsed time: " << (time(0)-itime) << "s" <<
endl;
    }

    return 0;
}

```

C.7 – Processamento

```

#include <iostream>
#include <string>
#include <cstdlib>
#include <cmath>
#include <glut.h>
#include "biblioteca.cpp"

double pi = 4.0*atan(1);
double r2 = sqrt(2);
double c45 = cos(pi/4);

double ang;
int quad;
imagef timg;
imagef cimg;
imagef uimg;
imagef eimg;

void erro()
{
    cout << "=====" << endl;
    cout << "          Reconstrucao do Objeto          " << endl;
    cout << "=====" << endl <<
endl;
    cout << "Uso: REC <ang> <quad> <od> <rd> <oe> <re>" << endl;
    exit(1);
}

void init()
{
    glClearColor(1,1,1,0);

    glEnable(GL_DEPTH_TEST);
    glEnable(GL_COLOR_MATERIAL);
    glEnable(GL_LINE_SMOOTH);
}

void display()
{
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
}

```



```

glColor3d(0,0,0);

double xf = (double)(cimg.w()-1)/2;
double yg = (double)(cimg.h()-1)/2;
int dx, dxs;
int lf = 1;
double zpd, zpe;
double zfd = 1500;
double xgd = 912.5;
double zgd = 1775.6;
double zfe = 1402;
double xge = 992.5;
double zge = 1935.6;

int xpd, ypd, xpe, ype;

for(int xre=0; xre<uimg.w(); ++xre)
{
glBegin(GL_LINE_STRIP);
for(int yre=0; yre<uimg.h(); ++yre)
{
int yse = yre;
int xse = xre;
double dhe = uimg(xre,yre)-eimg(xse,yse);
dxs = 0;

if(dhe<0)
{
while((dhe<0)&&(xse>0))
{
++xse;
dhe = uimg(xre,yre)-eimg(xse,yse);
}
lf = lf-1;
dxs = (xre-xse)+lf;
}

else
{
dxs = 0;
lf = 1;
}

zpe = (-1*dxs*zge*zfe)/(-1*dxs*zge-zge*xf+xre*zge+xge*zfe-zfe*xre);
xpe = (int)((-1*xre*zpe+xre*zge+zpe*xge)/zge+0.5);
ype = (int)((-1*yre*zpe+yre*zge+yg*zpe)/zge+0.5);

glVertex3f(uimg.w()-xpe,ype,zpe);

glBegin(GL_TRIANGLES);
glVertex3d((GLdouble)(uimg.w()-xpe), (GLdouble)ype, (GLdouble)(zpe-1));
glVertex3d((GLdouble)(uimg.w()-xpe+1), (GLdouble)ype, (GLdouble)(zpe-1));
glVertex3d((GLdouble)(uimg.w()-xpe), (GLdouble)(ype-1), (GLdouble)(zpe-1));
glEnd();

glBegin(GL_TRIANGLES);

```

```

        glVertex3d((GLdouble)(uimg.w()-xpe-1), (GLdouble)ype,
(GLdouble)(zpe-1));
        glVertex3d((GLdouble)(uimg.w()-xpe+1), (GLdouble)(ype-1),
(GLdouble)(zpe-1));
        glVertex3d((GLdouble)(uimg.w()-xpe+1), (GLdouble)(ype+1),
(GLdouble)(zpe-1));
    }
    glEnd();
}

for(int xrd=0; xrd<timg.w(); ++xrd)
{
    glBegin(GL_LINE_STRIP);
        for(int yrd=0; yrd<timg.h(); ++yrd)
        {
            int ysd = yrd;
            int xsd = xrd;
            double dhd = timg(xrd,yrd)-cimg(xsd,ysd);
            dx = 0;

            if(dhd<0)
            {
                while((dhd<0)&&(xsd<cimg.w()))
                {
                    ++xsd;
                    dhd = timg(xrd,yrd)-cimg(xsd,ysd);
                }
                lf = lf-1;
                dx = (xrd-xsd)+lf;
            }

            else
            {
                dx = 0;
                lf = 1;
            }

            zpd = (-1*dx*zgd*zfd)/(-1*dx*zgd-zgd*xf+xrd*zgd+xgd*zfd-zfd*xrd);
            xpd = (int)((-1*xrd*zpd+xrd*zgd+zpd*xgd)/zgd+0.5);
            ypd = (int)((-1*yrd*zpd+yrd*zgd+yg*zpd)/zgd+0.5);

            glVertex3f(xpd, ypd, zpd);

            glBegin(GL_TRIANGLES);
            glVertex3d((GLdouble)xpd, (GLdouble)ypd, (GLdouble)(zpd-1));
            glVertex3d((GLdouble)(xpd+1), (GLdouble)ypd, (GLdouble)(zpd-1));
            glVertex3d((GLdouble)xpd, (GLdouble)(ypd-1), (GLdouble)(zpd-1));
            glEnd();

            glBegin(GL_TRIANGLES);
            glVertex3d((GLdouble)(xpd-1), (GLdouble)ypd, (GLdouble)(zpd-1));
            glVertex3d((GLdouble)(xpd+1), (GLdouble)(ypd-1), (GLdouble)(zpd-1));
            glVertex3d((GLdouble)(xpd+1), (GLdouble)(ypd+1), (GLdouble)(zpd-1));
            glEnd();
        }
    }
}

```

```

    }

    glBegin(GL_LINE_LOOP);
        glVertex3f(0,0,0);
        glVertex3f(0,timg.h(),0);
        glVertex3f(timg.w(),timg.h(),0);
        glVertex3f(timg.w(),0,0);
    glEnd();

    glColor3d(0.75,0.75,0.75);
    for(int i=0; i<timg.w(); ++i)
    {
        glBegin(GL_LINE_STRIP);
        for(int j=0; j<timg.h(); ++j)
        {
            glVertex3f(i,j-1,0);
            glVertex3f(i-1,j,0);
        }
    }

    glFlush();
}

void reshape(int w, int h)
{
    glViewport(0,0,(GLsizei)w,(GLsizei)h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    double l, r, d;
    switch(quad)
    {
        case 1:
            l = c45*timg.h();
            r = c45*timg.w();
            d = (l+r)*0.1;
            glOrtho(-l-d,r+d,-d,(l+r)*h/w,0,(l+r));
            gluLookAt(0,0,0,1,1,-r2*tan(ang/180*pi),0,0,1);
            break;

        case 2:
            l = c45*timg.w();
            r = c45*timg.h();
            d = (l+r)*0.1;
            glOrtho(-l-d,r+d,-d,(l+r)*h/w,0,(l+r));
            gluLookAt(timg.w(),0,0,timg.w()-1,1,-
r2*tan(ang/180*pi),0,0,1);
            break;

        case 3:
            l = c45*timg.w();
            r = c45*timg.h();
            d = (l+r)*0.1;
            glOrtho(-l-d,r+d,-d,(l+r)*h/w,0,(l+r));
            gluLookAt(0,timg.h(),0,1,timg.h()-1,-
r2*tan(ang/180*pi),0,0,1);
            break;

        case 4:

```

```

        l = c45*timg.h();
        r = c45*timg.w();
        d = (l+r)*0.1;
        glOrtho(-l-d,r+d,-d,(l+r)*h/w,0,(l+r));
        gluLookAt(timg.w(),timg.h(),0,timg.w()-1,timg.h()-
1,-r2*tan(ang/180*pi),0,0,1);
        break;
    }

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

int main(int argc, char* argv[])
{
    glutInit(&argc, argv);

    if(argc!=7) erro();

    ang = atof(argv[1]);
    quad = atoi(argv[2]);
    string ffn = argv[3];
    string bfn = argv[4];
    string ufn = argv[5];
    string efn = argv[6];

    if((ang<0)|| (ang>45)) erro();
    if((quad<1)|| (quad>4)) erro();
    try
    {
        loadbmp(ffn, timg);
        loadbmp(bfn, cimg);
        loadbmp(ufn, uimg);
        loadbmp(efn, eimg);
    }
    catch(bad_file)
    {
        erro();
    }

    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB|GLUT_DEPTH);
    glutInitWindowSize(800,600);
    glutInitWindowPosition(0,0);
    glutCreateWindow(argv[0]);
    init();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMainLoop();
    return 0;
}

```

C.8 – Biblioteca

```

#include <iostream>
#include <fstream>
#include <string>

```

```
#include <memory>

using namespace std;

class bad_file { };

class imagef
{
public:
    imagef();
    imagef(int, int);
    imagef(const imagef&);
    imagef& operator=(const imagef&);
    ~imagef();

    void resize(int, int);

    int w() const;
    int h() const;
    int s() const;

    float& operator()(int);
    float& operator()(int, int);
    float& bg();

private:
    int _w_, _h_, _s_;
    float* _v_;

    void _a_(int, int);
    void _d_();
};

void loadbmp(const string&, imagef&);
void savebmp(imagef&, const string&);

imagef::imagef()
{
    _a_(0,0);
}

imagef::imagef(int _w, int _h)
{
    if((_w<=0)||(_h<=0))
    {
        _w = 0;
        _h = 0;
    }

    _a_(_w, _h);
}

imagef::imagef(const imagef& _s)
{
    _a_(_s._w_, _s._h_);

    for(int k=0; k<=_s._s_; ++k)
        _v_[k] = _s._v_[k];
}
}
```

```
imagef& imagef::operator=(const imagef& _s)
{
    if(this==&_s) return *this;

    _d_();
    _a_(_s._w_,_s._h_);

    for(int k=0; k<=_s; ++k)
        _v_[k] = _s._v_[k];

    return *this;
}

imagef::~~imagef()
{
    _d_();
}

void imagef::resize(int _w, int _h)
{
    if((_w<=0)||(_h<=0))
    {
        _w = 0;
        _h = 0;
    }

    _d_();
    _a_(_w,_h);
}

inline int imagef::w() const
{
    return _w_;
}

inline int imagef::h() const
{
    return _h_;
}

inline int imagef::s() const
{
    return _s_;
}

float& imagef::operator()(int _p)
{
    if((_p<0)||(_p>_s_)) _p = _s_;

    return _v_[_p];
}

float& imagef::operator()(int _x, int _y)
{
    int _p = _y*_w_+_x;

    if((_x<0)||(_x>=_w_)||(_y<0)||(_y>=_h_)) _p = _s_;
}
```

```
        return _v[_p];
    }

float& imagef::bg()
{
    return _v[_s_];
}

void imagef::_a(int _w, int _h)
{
    _w_ = _w;
    _h_ = _h;
    _s_ = _w*_h_;

    try
    {
        _v_ = new float[_s_+1];
    }
    catch(bad_alloc)
    {
        cerr << "in imagef: bad_alloc" << endl;
        exit(1);
    }
}

void imagef::_d()
{
    delete[] _v_;
}

void loadbmp(const string& _fn, imagef& _s)
{
    ifstream _ifs(_fn.c_str(), ios::binary|ios::in);
    if(!_ifs) throw bad_file();

    unsigned short type = 0;

    struct file_header
    {
        unsigned long size;
        unsigned short res1;
        unsigned short res2;
        unsigned long offs;
    } fh;

    struct info_header
    {
        unsigned long hsize;
        unsigned long width;
        unsigned long heigh;
        unsigned short plane;
        unsigned short bitco;
        unsigned long compr;
        unsigned long isize;
        unsigned long xreso;
        unsigned long yreso;
        unsigned long cluse;
        unsigned long climp;
    } ih;
```

```

    _ifs.read((char*)&type, 2);
    _ifs.read((char*)&fh, 12);
    _ifs.read((char*)&ih, 40);

    if(type!=19778) throw bad_file();
    if(ih.bitco!=8) throw bad_file();
    if(ih.compr!=0) throw bad_file();

    for(int k=0; k<ih.cluse; ++k)
        _ifs.ignore(4);

    _s.resize(ih.width,ih.heigh);

    int _dw = _s.w();
    if(_dw%4) _dw += 4-_dw%4;

    unsigned char* _l = new unsigned char[_dw];

    for(int j=(_s.h()-1); j>=0; --j)
    {
        _ifs.read((char*)_l, _dw);

        for(int i=0; i<_s.w(); ++i)
            _s(i,j) = (double)_l[i]/255;
    }

    delete[] _l;
}

void savebmp(imagef& _s, const string& _fn)
{
    ifstream _ofs(_fn.c_str(), ios::binary|ios::out);
    if(!_ofs) throw bad_file();

    unsigned short type;

    struct file_header
    {
        unsigned long size;
        unsigned short res1;
        unsigned short res2;
        unsigned long offs;
    } fh;

    struct info_header
    {
        unsigned long hsize;
        unsigned long width;
        unsigned long heigh;
        unsigned short plane;
        unsigned short bitco;
        unsigned long compr;
        unsigned long isize;
        unsigned long xreso;
        unsigned long yreso;
        unsigned long cluse;
        unsigned long climp;
    } ih;
}

```



```
struct ctag_header
{
    unsigned char blu;
    unsigned char gre;
    unsigned char red;
    unsigned char res;
} ch;

int _dw = _s.w();
if(_dw%4) _dw += 4-_dw%4;

type = 19778;
fh.size = 1078+_dw*_s.h();
fh.res1 = 0;
fh.res2 = 0;
fh.off5 = 1078;
ih.hsize = 40;
ih.width = _s.w();
ih.heigh = _s.h();
ih.plane = 1;
ih.bitco = 8;
ih.compr = 0;
ih.isize = 0;
ih.xreso = 0;
ih.yreso = 0;
ih.cluse = 256;
ih.climp = 0;

_ofs.write((char*)&type, 2);
_ofs.write((char*)&fh, 12);
_ofs.write((char*)&ih, 40);

for(int k=0; k<256; ++k)
{
    ch.blu = k;
    ch.gre = k;
    ch.red = k;
    ch.res = 0;

    _ofs.write((char*)&ch, 4);
}

unsigned char* _l = new unsigned char[_dw];

for(int j=(_s.h()-1); j>=0; --j)
{
    for(int i=0; i<_s.w(); ++i)
        _l[i] = (unsigned char)(_s(i,j)*255+0.5);

    _ofs.write((char*)_l, _dw);
}

delete[] _l;
}
```

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] **SearchSmallBizIT.comDefinitions**, disponível em:
http://searchsmallbizit.techtarget.com/sDefinition/0,,sid44_gci507015,00.html,
Acesso em 09 de Agosto de 2004;
- [2] Anonymous. **Designing forward and back**. Tooling & Production, Solon,
Maio de 2001;
- [3] Varady, T.; Martin, R. R.; Coxt, J. **Reverse engineering of geometric models – an introduction**. Computer Aided Design, Vol 29, N° 4, pp. 255-268, 1997;
- [4] Dankwort, C. W.; Podehl, G. **A new aesthetic design workflow – results from the european project FIORES**. CAD Tools and algorithms for product design, Cap.1, pp 16-30, 2000;
- [5] Germani, M.; Mandorli, S.; Berti, S. **Design process: The impact of reverse engineering methodologies**. XII ADM International Conference – Rimini – Italy , 2001;
- [6] Busch, T.; Harlow, R.; Thompson, R. **Fundamentals of dimensional metrology**, 1998;
- [7] Becker, K.; Heynacher, E. **M400 – A Coordinate measuring machine with 10 nm resolution**, SPIE Vol. 802 In-Process Optical Metrology for Precision Engineering, pags. 209-216, 1987;
- [8] Jarvis, R. **A Perspective on range finding techniques for computer vision**, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-5, No. 2, pags. 122-139, 1983;
- [9] Chugui, Y. V., et al. **Experience of triangulation method using for industrial dimensional inspection**, Proceedings of the International Symposium on Laser Metrology for Precision Measurement and Inspection in Industry, 1999;
- [10] Folsom, T. C.; Bondurant, P. D. **Non-contact internal thread inspection**, SAE Technical Paper Series, 1999;

- [11] Fan, K.-C.; Tsai, T.-H. **Optimal shape error analysis of the matching image for a free-form surface**, Robots and Computer Integrated Manufacturing, 2001;
- [12] Tome, A. J.; Stahl, H. P. **Phase-measuring interferometry: applications and techniques**, Proceedings of SPIE – The International Society of Optical Engineering, pags. 71-77, 1988;
- [13] Grant, R. M.; Brown, G. M. **Holographic nondestructive testing (HNDT) in the automotive industry**, SAE Technical Papers;
- [14] Kim, T.; Hong, H.; Choi, J. **Estimation of hybrid reflectance properties and shape reconstruction using the LMS method**, Pattern Recognition, 2000;
- [15] Kosmopoulos, D.; Varvarigou, T. **Automated inspection of gaps on the automobile production line through stereo vision and specular reflection**, Computers in Industry, 2001;
- [16] Zheng, J.; Fukagawa, Y.; Abe, N. **3D Surface estimation and model construction from specular motion in image sequences**, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-19, No. 5, pags. 513-520, 1997;
- [17] Fox, J. S. **Range from translational motion blurring**, IEEE Transactions on Pattern Analysis and Machine Intelligence, pags. 360-365, 1988;
- [18] Dropps, S. H. **Reverse engineering using X-ray technique**, SAE, 1998;
- [19] Gupta, N. K.; Isaacson B. **Tangential computed tomography for industrial inspection**, SAE, 2001;
- [20] Whitehouse, D. J. **Handbook of surface metrology**, pags. 486-491, 1994;
- [21] Neto, P. S. **Shadow moire as a tool for measurements and tridimensional quality control in curved surfaces**, SAE, 2000;
- [22] Wykes, C.; Morshedizadeh R. **Surface topography measurements using digital moirè contouring – errors and limitations**, Journal of Engineering Manufacture, 1995;

- [23] Gäsvisk, K. J. **Moirè contouring of objects generated by CAD**, SPIE Vol. 814 Photomechanics and Speckle Metrology, págs. 303–307, 1987;
- [24] Kim, S.; Choi Y.; Oh, J. **Reverse engineering: high speed digitization of free-form surfaces by phase-shifting grating projection moirè topography**, International Journal of Machine Tools & Manufacture, 1999;
- [25] Sato, Y.; Kitagawa, H.; Fujita, H. **Shape measurement of curved objects using multiple slit-ray projections**, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-4, No. 6, págs. 641-646, 1982;
- [26] Blake, A.; et al. **Trinocular active Range sensing**, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 5, págs. 477-483, 1993;
- [27] Vuylsteke, P.; Oosterlinck, A. **Range image acquisition with a single binary-encoded light pattern**, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 2, págs. 148-164, 1990;
- [28] Valkenburg, R. J.; McIvor, A.M. **Accurate 3D measurement using a structured light system**, Image and Vision Computing, 1998;
- [29] Osawa, S.; et al. **3-D Shape measurement by self-referenced pattern projection method**, Measurement, 1999;
- [30] Miyasaka, T.; Araki, K.; **Development of real time 3-D measurement system using intensity ratio**. In: International Society for Photogrammetry and Remote Sensing (ISPRS) Commission III, Symposium 2002;
- [31] Salvi, J.; Batlle, J.; Mouaddib, E.; **A robust-coded pattern projection for dynamic 3D scene measurement**, Pattern Recognition Letters, 1998;
- [32] Claus, M. **Experiences with InduSURF in 3D measurement of industrial surfaces**, In: International Archives of Photogrammetry and Remote Sensing, Vol. 27, Part B5, 1988;
- [33] Maruyama, M.; Abe, S. **Range sensing by projecting multiple slits with random cuts**, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 6, págs. 647-651, 1993;

- [34] Lee, K. H.; Park, H. **Automated inspection planning of free-form shape parts by laser scanning**, Robotics and Computer Integrated Manufacturing, 2000;
- [35] Chen, C.; Zheng, Y. F. **Passive and active stereo vision for smooth surface deflection of deformed plates**, IEEE Transactions on Industrial Electronics, Vol. 42, No.3, 1995;
- [36] Liu, L.; Sawada, T.; Sakamoto, M. **Evaluation of the surface deflections in pressed automobile panels by an optical reflection method**, Journal of Materials Processing Technology, 2000;
- [37] Niel, A.; Kölbl, S. H.; Burgstaller, M. **Robotic three-dimensional measurement system for complex parts using structured light**. In: International Society for Photogrammetry and Remote Sensing (ISPRS) Commission III, Symposium 2002;
- [38] Nishida, L. **Estudo e implementação de um rainbow range finder**, Dissertação apresentada à Escola Politécnica/USP, 2002;
- [39] Fu, K.S., **Robotics: control, sensing, vision and intelligence**, McGraw-Hill International Editions, 1987;
- [40] Nitz, M.; Galha, R. **Calcule com o mathcad versão 11**, Editora Erica, 1º Edição, 2003;
- [41] Mizrahi, Victorine V. **Treinamento em linguagem C++**, Módulos 1 e 2, Makron Books, 1995;
- [42] Stroustrup, Bjarne. **A linguagem de programação C++**, 3º Edição, Bookman, 2002;
- [43] Parker, J. R. **Algorithms for image processing and computer vision**, Wiley Computer Publishing, 1997;
- [44] Zhang, Z. **Flexible camera calibration by viewing a plane from unknown orientations**, Technical Report MSR-TR-98-71, Microsoft Research, 1998;
- [45] Zhang, G.; Wei, A. **A novel calibration approach to structured light 3D vision inspection**, Optics & Laser Technology, 2002;

[46] **Sony Cybershot DSC P10 Specification**, disponível em: <http://www.digital-camera-reviews.info/specs/sony-cybershot-dsc-p10.cfm> Acesso em 10 de agosto de 2004;