

## **4 Implementação, Resultados e Testes**

Este capítulo apresenta as duas aplicações implementadas, uma para criar os modelos e outra para visualizar e comparar resultados. Em seguida são apresentados alguns resultados, incluindo uma discussão sobre o sistema implementado. A seção de testes conclui o capítulo com dois testes realizados para verificar a precisão do sistema implementado.

### **4.1. Implementação**

Como resultado desta dissertação, obtemos duas aplicações. A primeira aplicação, chamada Shaper, é utilizada para capturar imagens de objetos e criar um modelo correspondente. E a segunda aplicação, chamada simplesmente de Visualizer, é utilizada para visualizar modelos criados e compará-los.

Para a camada de interface das aplicações utilizou-se GLUI [24] que é uma biblioteca de extensão para a popular GLUT. Como biblioteca gráfica utilizou-se OpenGL e para o desenvolvimento utilizou-se o Visual Studio .NET da Microsoft.

#### **4.1.1. Shaper**

De um modo geral, esta aplicação é responsável por executar todos os itens apresentados na seção anterior, como projetar padrões de luz estruturada codificada, capturar imagens e processá-las, fazer triangulação entre pontos correspondentes etc. Também deve disponibilizar todas as opções e parâmetros discutidos. Para a captura de imagens através das câmeras de vídeo, utilizou-se a biblioteca IM Capture [25], enquanto a biblioteca OpenCV [26] foi utilizada extensivamente para o processamento de imagens. No apêndice A foi incluído uma descrição do OpenCV e de seus principais recursos.

A aplicação possui uma janela principal, onde são apresentadas as informações principais. Inicialmente esta janela mostra o par de imagens das câmeras capturadas em tempo real, permitindo o enquadramento do objeto capturado, ajustes de parâmetros de captura e calibração das câmeras.

Junto com a janela principal, existe uma outra janela para acessar as diversas opções do sistema. A figura 29 mostra esta janela e a janela principal ao fundo. As opções estão divididas em três grupos. O grupo *File* possui as opções de operação do sistema. A primeira opção *Live* está habilitada quando a aplicação é inicializada e as imagens das câmeras capturadas em tempo real são exibidas. A segunda opção *Scan* inicia o processo de varredura, capturando a seqüência de imagens do objeto modelo iluminado pelos diferentes padrões da luz estruturada. Após a varredura, as imagens são processadas. A terceira opção *Correspondence* utiliza as imagens processadas e realiza a correspondência entre pixels das duas câmeras. Na janela principal o usuário pode clicar em um ponto da imagem de uma câmera e descobrir os pontos correspondentes na imagem da outra câmera. Por último, a opção *3D Model* cria o modelo geométrico. As duas últimas opções não estão habilitadas antes que a etapa de captura das imagens esteja concluída.

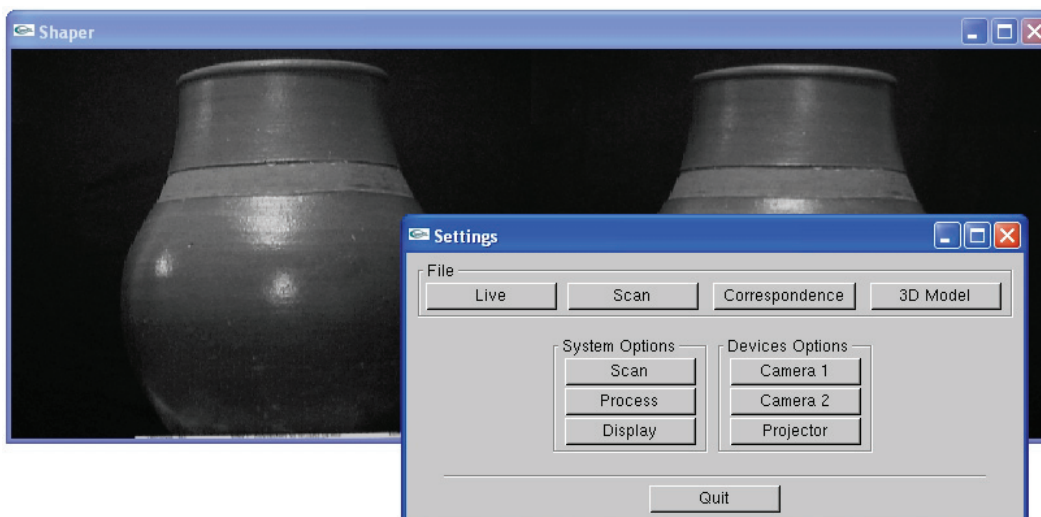


Figura 29 – Janela de opções da aplicação Shaper.

O segundo grupo reúne as opções do sistema, onde cada uma abre uma nova janela. Na primeira opção *Scan* o usuário pode modificar as opções de varredura relacionadas à luz estruturada e as imagens capturadas. O segundo botão *Process* permite ao usuário modificar opções de processamento das imagens capturadas na

etapa de varredura. A última opção deste grupo permite selecionar os diferentes modos de visualização do modelo criado.

No último grupo encontram-se as opções referentes aos equipamentos utilizados. Quanto às câmeras utilizadas, podemos modificar os seus vários parâmetros, como brilho, ganho, nitidez, foco etc. O usuário também pode calibrar os parâmetros extrínsecos e intrínsecos de cada uma. A opção *Salvar* armazena tanto a calibração quanto o ajuste da câmera. Em relação ao projetor, devemos informar a sua resolução física, de forma que as linhas projetadas tenham a mesma largura em pixels, conforme foi discutido anteriormente.

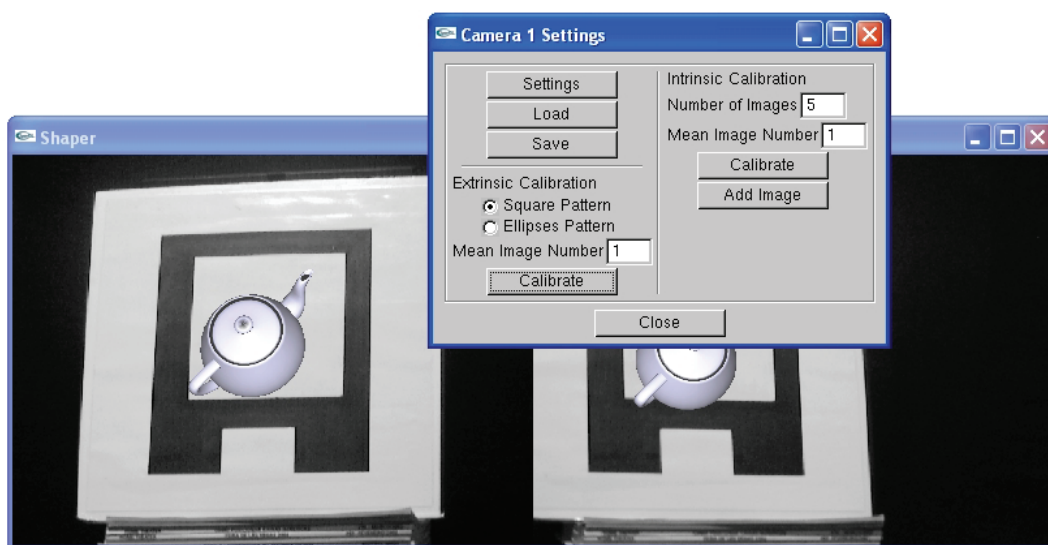


Figura 30 – Calibração de parâmetros extrínsecos das duas câmeras.

A calibração dos parâmetros intrínsecos das câmeras utiliza a implementação do método de Zhang [18] fornecida pelo OpenCV. Esta função calibra tanto os parâmetros extrínsecos quanto intrínsecos, porém utilizamos apenas estes últimos. Não foi objetivo desta dissertação comparar o método de calibração de câmeras de Tsai com o método proposto por Zhang.

#### 4.1.2. Visualizer

Esta simples aplicação foi criada para visualizar e comparar os modelos criados na aplicação apresentada anteriormente, sem a necessidade do uso de câmeras. Assim como o Shaper, o usuário pode selecionar um dos modos de visualização e habilitar textura. É possível carregar dois modelos simultaneamente

e alterar entre um e outro, comparando visualmente o resultado. Pode-se girar o modelo visualizado através de uma interface *arcball* [27] com o mouse ou pelo teclado. O usuário também pode calcular a distância entre dois pontos do modelo, selecionando com o botão direito do mouse os pontos desejados. Por fim, há uma função para ajustar um cilindro ao modelo carregado. O Visualizer foi utilizado nos testes que serão apresentados em seção posterior.

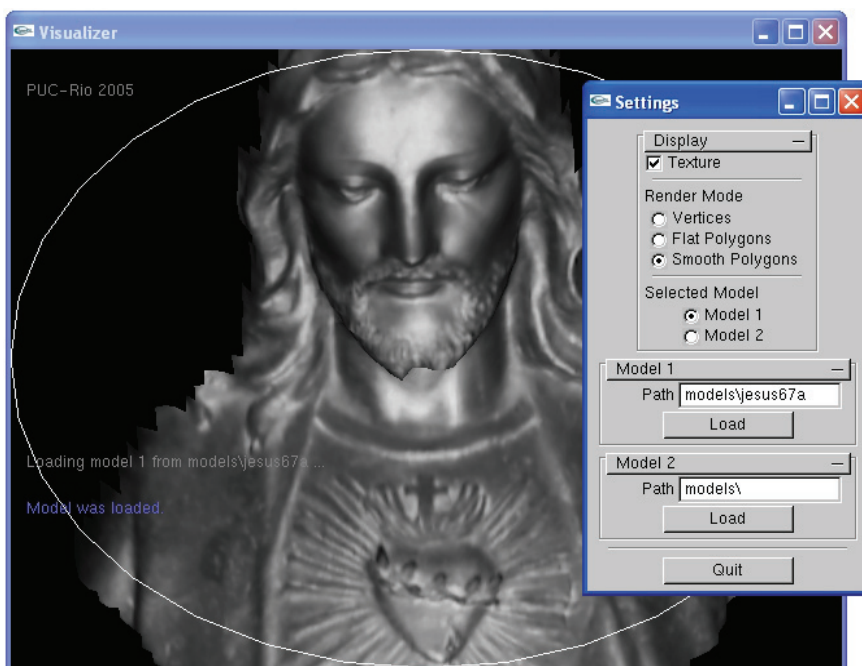


Figura 31 – Modelo carregado na aplicação Visualizer.

#### 4.1.3. Arquivo de Saída

Em relação ao armazenamento de modelos, ambas as aplicações utilizam um formato de arquivo criado que segue uma estrutura simples de interpretação. Este arquivo texto descreve os vértices e polígonos criados para o objeto capturado, como é mostrado em (4.1). A primeira linha contém dois inteiros: o número de vértices  $n$  e o número de polígonos  $m$ . O bloco de linhas seguinte descreve cada vértice do modelo. Cada linha corresponde a um vértice  $i$  e possui dez números reais. Os três primeiros números dão a posição  $(x_i, y_i, z_i)$  do vértice no mundo e os três números seguintes dão a normal  $(nx_i, ny_i, nz_i)$  do vértice, já normalizada. Em seguida temos as coordenadas de textura  $(re_i, se_i)$  para a textura da câmera

esquerda e por último são dadas as coordenadas de textura  $(rd_i, sd_i)$  para a textura da outra câmera.

$$\begin{array}{cccccccccc}
 n & m & & & & & & & & & \\
 x_1 & y_1 & z_1 & nx_1 & ny_1 & nz_1 & re_1 & se_1 & rd_1 & td_1 \\
 x_2 & y_2 & z_2 & nx_2 & ny_2 & nz_2 & re_2 & se_2 & rd_2 & td_2 \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 x_n & y_n & z_n & nx_n & ny_n & nz_n & re_n & se_n & rd_n & td_n \\
 a_1 & b_1 & c_1 & nx_1 & ny_1 & nz_1 & & & & \\
 a_2 & b_2 & c_2 & nx_2 & ny_2 & nz_2 & & & & \\
 \dots & \dots & \dots & \dots & \dots & \dots & & & & \\
 a_m & b_m & c_m & nx_m & ny_m & nz_m & & & & 
 \end{array} \tag{4.1}$$

Cada linha no segundo bloco de linhas descreve um polígono  $j$  e possui três números inteiros seguidos de três números reais. Os três primeiros números  $(a_j, b_j, c_j)$  dão o índice  $i$  de cada um dos três vértices que formam o polígono. E os três números reais dão a normal  $(nx_j, ny_j, nz_j)$  do polígono.

Junto a cada arquivo são fornecidos dois arquivos de imagem, no formato BMP, referentes às duas texturas, uma de cada câmera. Estas imagens são carregadas junto com o arquivo correspondente.

## 4.2. Resultados

O sistema foi utilizado em uma plataforma constituída de processador AMD Athlon XP1600+, com 512MB de memória e placa gráfica NVIDIA FX5600. O par de câmeras utilizadas são do modelo DFW-VL500 da Sony, enquanto que foram utilizados dois projetores distintos no sistema. O primeiro projetor, modelo CTX EzPro 610, é um projetor LCD, enquanto que o segundo, modelo Infocus X1, é um projetor DLP.

O sistema foi utilizado para capturar formas de diversos objetos de natureza diferentes. Vasos, estátuas, caixas de som, tubulações e até pessoas serviram de modelos. O modelo na figura 32a foi obtido a partir de uma estátua de bronze. A figura 32c mostra o modelo de um vaso, enquanto que as figuras 32b e 32d apresentam modelos capturados a partir de tubulações de PVC.

De uma forma geral, a qualidade dos modelos estão intimamente ligados com o número de linhas utilizadas pela luz estruturada. Este número variou de  $32 \times 32$  a  $128 \times 128$  e é limitado pelo hardware utilizado. Quanto maior o número de linhas, maior o nível de detalhes. As figuras 33a e 33b ilustram a diferença entre modelos capturados sob as mesmas condições, porém utilizando um número de linhas diferentes. O primeiro utilizou  $64 \times 128$  linhas, enquanto que o segundo apenas  $32 \times 64$ . As outras duas figuras 33c e 33d mostram a diferença entre as bordas de modelos capturados utilizando-se  $64 \times 128$  e  $32 \times 64$  linhas respectivamente.

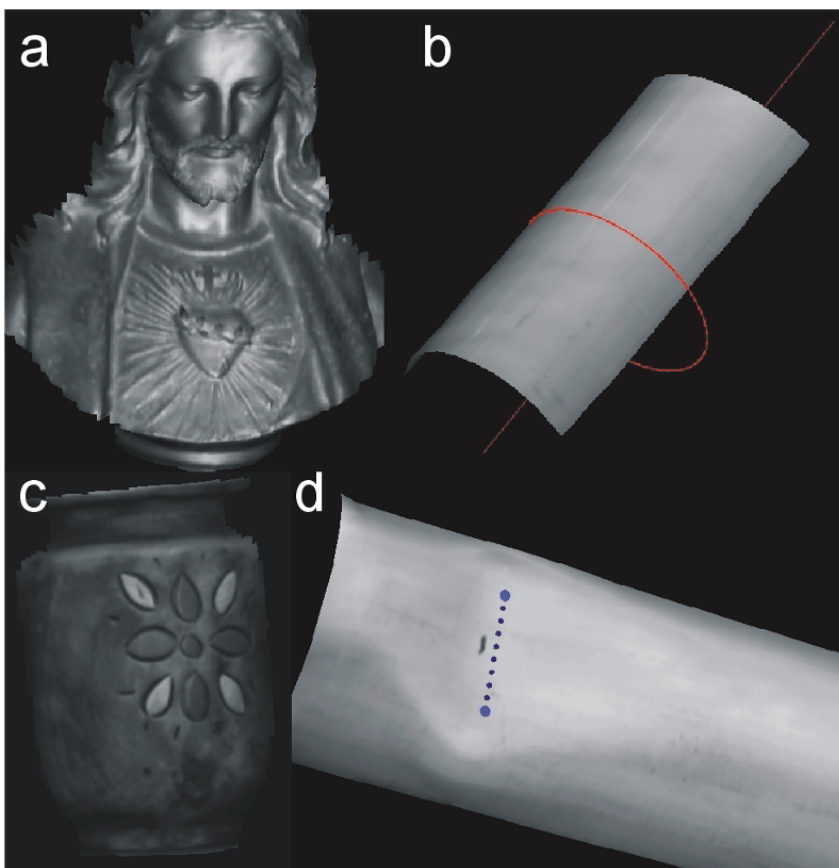


Figura 32 – Modelos criados a partir de objetos reais: (a) estátua de bronze; (b) tubulação com cilindro ajustado; (c) vaso; (d) tubulação danificada.

Porém quando o número de linhas cresce e a resolução das câmeras é mantido (no nosso caso de  $640 \times 480$ ), é introduzido irregularidades na superfície dos modelos capturados que seguem um padrão de interferência, como mostra a figura 33e. Isso acontece, pois na fase de processamento de imagens padrões de interferência são gerados nas imagens processadas. Estes padrões são repassados

para o modelo final, apresentando irregularidades na superfície. Esses padrões de interferência tendem a diminuir quando a resolução das imagens capturadas cresce. Por isso que, para imagens com a mesma resolução, quanto maior o número de linhas utilizadas pela luz estruturada mais interferências são geradas. Esta irregularidade gerada é mais visível em modelos com muita superfície plana. Para câmeras com resolução de  $640 \times 480$ , o ideal seria utilizar um número de linhas de até  $32 \times 32$ . Cada vez que dobrarmos o número de linhas, também devemos dobrar a resolução das câmeras.

Os modelos das figuras 32b e 32d foram capturados utilizando-se este número recomendado de linhas e possuem uma superfície quase sem ruído, ao contrário do modelo da figura 33e que foi capturado utilizando-se  $128 \times 128$  linhas. O mesmo acontece com os modelos das figuras 33c e 33d. Apesar do último possuir bordas com menos detalhes, a sua superfície é livre de ruído.

Em relação às restrições do sistema, a primeira restrição é o tamanho dos objetos capturados. A princípio estamos interessados em objetos de tamanho variando de 30cm até 2m. Porém nada impede que se capture formas de objetos maiores. Para isso seriam necessários um projetor com maior brilho e câmeras com lentes adequadas.

Uma segunda restrição deste sistema é capturar objetos e cenas estáticas, como foi visto. Há uma quantidade grande de imagens a serem geradas para modelar cada objeto, de maneira que o sistema não consegue capturar objetos em movimento. Por exemplo, para capturar um objeto utilizando  $64 \times 64$  linhas, utilizando duas câmeras a 30fps com apenas um valor de exposição e sem calcular imagem média, seria necessário  $0.8s$ . Porém na prática este número é maior. As câmeras ficaram limitadas a 15fps e há um problema de sincronia entre a imagem desenhada no buffer e a imagem projetada pelo projetor LCD, de modo que é necessário esperar cerca de  $0.2s$  até que a imagem correspondente ao padrão desejado seja projetada. Como são necessários projetar 24 padrões, temos um atraso total de  $4.8s$ . Somando mais o tempo necessário para salvar as imagens capturadas em disco e outras operações necessárias, no total o sistema está levando em torno de  $7.0s$  para completar uma operação que deveria durar  $1.6s$ . A maior parte do atraso é devido ao problema de sincronismo que carece de melhor solução.

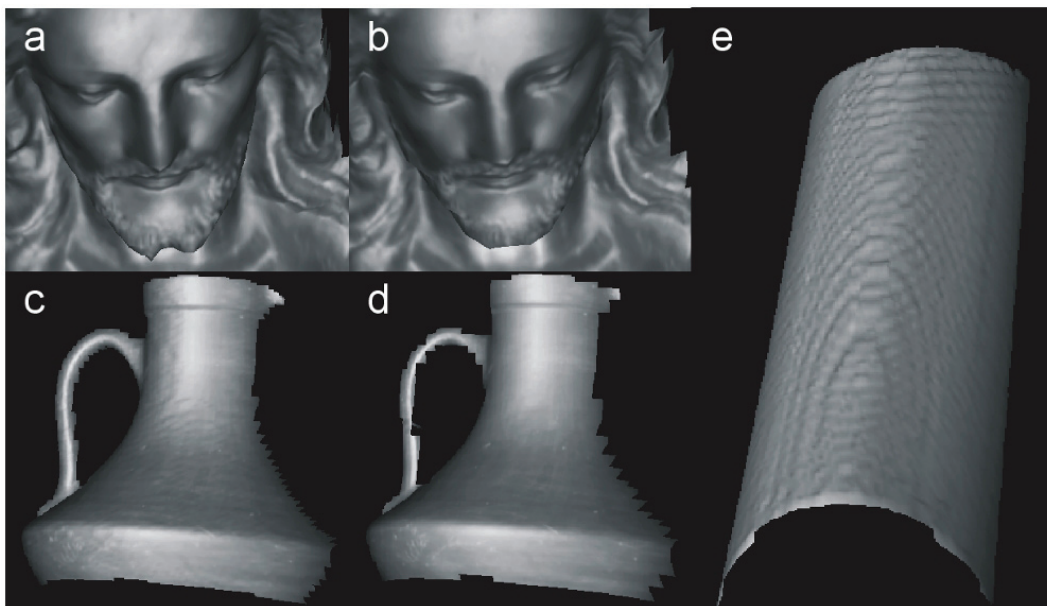


Figura 33 – Nível de detalhes em modelos criados: (a) estátua de bronze; (b) estátua de bronze com menos pontos; (c) moringa; (d) moringa com menos pontos; (e) tubulação com padrões de interferência.

A terceira restrição é que o objeto modelado é limitado por dois fatores na captura. O primeiro é que só são modeladas regiões iluminadas pela luz estruturada codificada. Segundo, só são modeladas regiões capturadas por ambas as câmeras. Regiões capturadas pelas câmeras, porém não iluminadas pelo projetor, são consideradas regiões de sombra. Regiões não capturadas por ambas as câmeras são regiões ocultas.

#### 4.2.1. Padrões de Calibração

Comparando os dois padrões de calibração, os resultados dos testes, apresentados posteriormente nesta seção, não mostram uma diferença considerável de precisão entre os dois. O padrão com vértices é mais fácil de ser manuseado, pois independe de quaisquer outros objetos capturados na imagem. Este padrão é robusto contra ruídos introduzidos pela câmera e funciona com a câmera posicionada em ângulos variados em relação ao seu plano. O seu ponto negativo é quando a câmera é colocada exatamente no eixo perpendicular ao plano do padrão e que intercepta o seu centro. Nesta posição a calibração se torna falha, pois não há como se obter informação de profundidade em relação a este eixo. Por último, este padrão é utilizado em uma outra aplicação em tempo real,



conseguindo uma taxa de 22 quadros por segundo, com imagens de tamanho  $640 \times 480$ . Este bom desempenho é devido à grande eficiência da biblioteca OpenCV, com a qual foram implementadas as aplicações, e ao método de reconhecimento de padrão proposto.

O padrão com círculos é o mais fácil de ser implementado. O seu posicionamento é mais difícil já que necessita ocupar a imagem capturada. Porém também é um padrão robusto contra ruídos e possui precisão menor do que um pixel. Na prática, este padrão quase não apresenta erros devido à câmera estar no eixo perpendicular ao plano do padrão e que intercepta o seu centro.

Como uma última observação em relação aos padrões, a calibração apresenta melhores resultados quando estes são colocados na mesma posição do objeto capturado, principalmente o padrão com vértices que possui maior liberdade de posicionamento, se tornando mais sensível à sua posição do que o padrão com círculos. Todos os testes foram feitos posicionando o padrão junto ao objeto capturado. A seção seguinte apresenta os testes realizados e o método de ajuste de cilindro a um modelo capturado.

#### **4.2.2. Câmeras**

As câmeras utilizadas foram um par de câmeras Sony DFW-VL500, as quais possuem sensores CCD e resolução máxima de  $640 \times 480$  pixels. A grande vantagem desta câmera em relação a câmeras *Web* convencionais está no seu conjunto óptico que possui zoom e focagem digitais, os quais são utilizados pela aplicação para melhor enquadrar o objeto modelado nas imagens capturadas, além de um menor nível de distorções ópticas.

Foi constatado em ambas as câmeras um problema que ocorre nas fronteiras das imagens capturadas entre listras claras e escuras projetadas sobre o objeto capturado. A câmera não é capaz de determinar com exatidão a fronteira entre as regiões e introduz erros sistemáticos. Possivelmente estes erros podem estar sendo introduzidos em algum estágio de compressão da imagem onde as frequências, como as encontradas entre regiões com as propriedades descritas, são altas. A figura 34a mostra uma imagem capturada onde tal erro ocorre entre as listras projetadas sobre o vaso capturado.

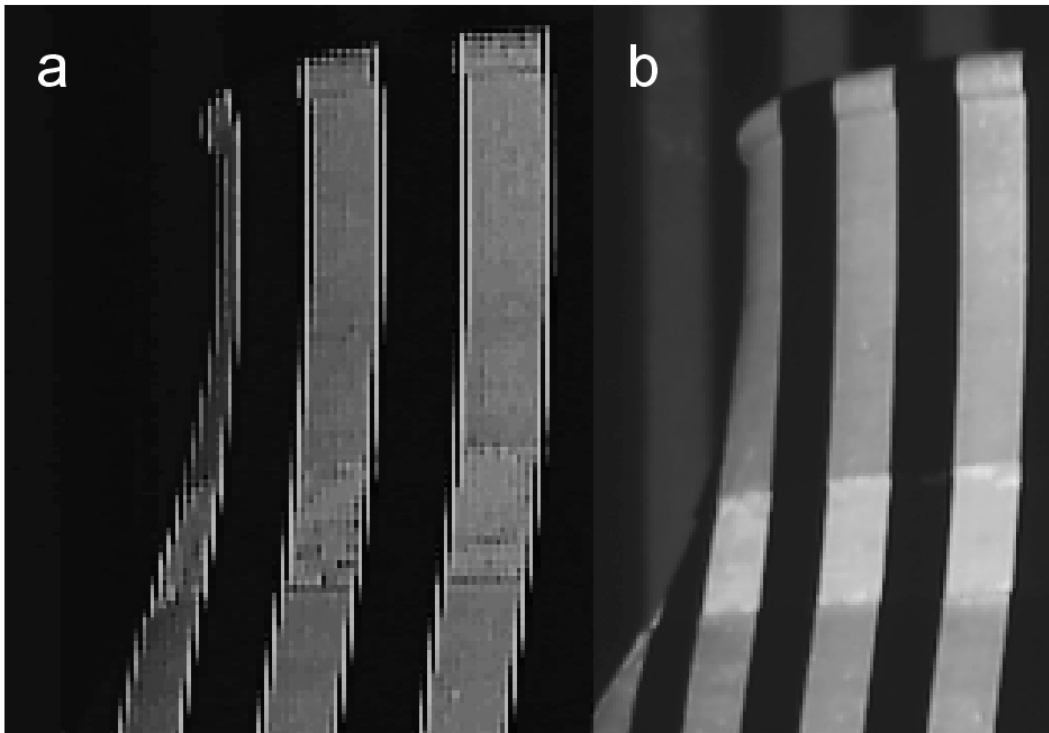


Figura 34 – Captura de fronteiras entre regiões claras e escuras: (a) fronteiras capturadas com erros; (b) fronteiras capturadas corretamente.

Curiosamente este problema não ocorre quando, ao invés de utilizarmos luz branca para projetarmos as listras claras dos padrões de luz estruturada, utilizamos luz verde. A figura 34b ilustra uma imagem do mesmo vaso modelado com luz estruturada verde. Conseqüentemente, utilizou-se esta cor como padrão neste trabalho. Utilizando as cores vermelho e azul o problema permanecia.

Esse problema não ocorre no padrão de calibração com vértices, que também possui regiões claras e escuras vizinhas. O padrão é sempre capturado sob luz ambiente. Nestas condições a câmera é capaz de capturar as fronteiras com clareza.

### 4.2.3. Projetores

Foram utilizados dois projetores distintos, um de tecnologia LCD (liquid crystal display) e o outro DLP (digital light processing). A maneira como um projetor DLP funciona [28], na qual cada componente de cor é projetada separadamente, torna o seu uso inadequado para esta aplicação, principalmente se

a codificação utilizada envolver padrões coloridos. Isso foi constatado na prática. Utilizando luz estruturada de cor verde, algumas imagens capturadas ficavam escuras porque a captura não era sincronizada com a projeção da componente de cor verde. O projetor utilizado foi um Infocus X1, com velocidade  $2x$  para a roda de cores. Como era de se esperar, o projetor LCD não apresentou este problema e foi utilizado como projetor padrão neste trabalho.

Como foi discutido anteriormente, o projetor LCD utilizado apresentou um atraso entre a imagem desenhada no buffer e a imagem projetada. Este problema deve ser inerente ao modelo CTX EzPro 610 utilizado. Outros modelos de projetores LCD devem ser testados para se tirar uma conclusão definitiva.

### **4.3. Testes**

Foram elaborados e executados dois testes que visavam verificar a precisão do sistema implementado. O primeiro teste consistiu simplesmente em medir a distância entre pontos conhecidos de um modelo escolhido. Para o segundo teste, utilizou-se um pedaço de tubulação cilíndrica de PVC como modelo e deveríamos obter os parâmetros de um cilindro correspondente à tubulação como resultado.

#### **4.3.1. Primeiro Teste**

Para este teste, utilizou-se o segundo padrão de calibração apresentado como modelo, cujos pontos estão todos em um mesmo plano e em posições conhecidas. A figura 15 ilustra os pontos deste padrão. Então para cada par de pontos, mediu-se a distância correspondente no modelo computacional gerado. Foram utilizados os dois padrões de calibração apresentados para calibrar o par de câmeras antes de se obter o modelo. Todos os modelos foram capturados utilizando a calibração de parâmetros intrínsecos das câmeras.

Para cada objeto capturado, obtém-se uma nuvem de pontos com a qual é criado um modelo poligonal. O processo de medir as distâncias entre dois pontos conhecidos requer identificá-los e selecioná-los sobre o modelo. Isso é feito de maneira visual, utilizando o mouse para selecionar os pontos desejados, assim como é feito em vários softwares comerciais. A textura capturada tem papel

fundamental na seleção, dando maior detalhamento da superfície do modelo. Sem esta informação visual, não seria possível selecionar pontos sobre os modelos capturados para este teste e outras aplicações do sistema.

Uma vez selecionado um ponto na tela pelo usuário, utiliza-se um método semelhante ao de Ray Tracing para calcular qual ponto do modelo corresponde ao ponto selecionado pelo usuário na imagem visualizada. O OpenGL permite identificar qual polígono do modelo foi selecionado pelo usuário com o mouse, utilizando o modo de renderização para seleção. Mais informações sobre este procedimento pode ser encontrado em [29]. Uma vez conhecido o polígono interceptado, a posição da câmera, o ângulo de visão, resolução da tela e o ponto da tela selecionado pelo usuário, lança-se um raio que sai da câmera, passa pelo ponto correspondente ao ponto no plano da imagem e intercepta o polígono. Basta então calcular o ponto de interseção entre este raio e o plano definido pelo polígono para encontrar o ponto selecionado no modelo.

<b>Segmento</b>	<b>Distância (cm) ± 0,05</b>	<b>Modelo Um Distância (cm)</b>	<b>Modelo Dois Distância (cm)</b>	<b>Modelo Três Distância (cm)</b>
$\overline{0\ 1}$	21,00	21,013	20,943	21,057
$\overline{2\ 7}$	21,00	21,068	21,004	20,934
$\overline{8\ 9}$	15,00	14,872	14,921	14,964
$\overline{10\ 11}$	14,90	14,896	14,833	14,853
$\overline{0\ 7}$	24,90	25,004	24,912	24,787
$\overline{1\ 2}$	24,90	24,858	24,900	24,950
$\overline{8\ 11}$	15,00	14,992	14,930	14,949
$\overline{9\ 10}$	15,00	14,883	14,943	14,991

Tabela 3 – Distâncias médias medidas utilizando-se o padrão com círculos para a calibração no primeiro teste.

Este processo de seleção de pontos é subjetivo devido à dificuldade de se identificar os vértices do padrão com exatidão. A textura possui uma resolução limitada, assim como as imagens capturadas, o que dá margem à interpretação na hora em que o usuário seleciona os pontos. Sendo assim, as medidas de cada modelo capturado foram feitas por três usuários distintos. Os valores encontrados

por cada usuário podem ser vistos no apêndice B. A figura abaixo ilustra a distância medida entre dois pontos selecionados no modelo do padrão utilizando-se o Visualizer.



Figura 35 – Distância medida entre dois pontos no modelo.

A tabela 3 apresenta os valores médios dos valores encontrados pelos usuários para cada modelo capturado utilizando-se o padrão de calibração com círculos, enquanto que a tabela 4 mostra os valores médios para cada modelo capturado utilizando-se o padrão de calibração com vértices. Podemos observar que todas as medidas médias estão dentro da margem de erro de 1%. As médias das medidas obtidas por cada usuário para cada segmento também estão dentro da margem de erro de 1%. Os desvios padrões calculados a partir destas tabelas foram  $0,066\text{cm}$  para o padrão com círculos e  $0,073\text{cm}$  para o padrão com vértices. Não há diferença relevante entre os resultados dos dois padrões de calibração utilizados neste teste. Caso uma determinada medida seja de fundamental importância, pode-se capturar o mesmo modelo várias vezes e calcular a distância média para se obter um valor com maior precisão. Podemos obter uma precisão próxima a de uma régua milimétrica.

Segmento	Distância (cm) ± 0,05	Modelo Um Distância (cm)	Modelo Dois Distância (cm)	Modelo Três Distância (cm)
$\overline{0\ 1}$	21,00	21,093	20,911	21,046
$\overline{2\ 7}$	21,00	21,007	20,931	20,970
$\overline{8\ 9}$	15,00	14,914	14,942	14,962
$\overline{10\ 11}$	14,90	14,869	14,921	14,853
$\overline{0\ 7}$	24,90	25,081	24,942	24,858
$\overline{1\ 2}$	24,90	24,854	24,878	25,052
$\overline{8\ 11}$	15,00	15,062	14,974	15,037
$\overline{9\ 10}$	15,00	14,940	14,905	15,037

Tabela 4 - Distâncias médias medidas utilizando-se o padrão com vértices para a calibração no primeiro teste.

A figura 32d ilustra um caso de uso do sistema como uma ferramenta de medição. O modelo analisado foi obtido a partir de uma tubulação amassada e o usuário deseja saber as dimensões do dano, para poder decidir como reparar a linha. Outra vantagem da utilização de um sistema deste tipo é que um especialista pode fazer análises remotamente sem necessidade de estar fisicamente presente no local da inspeção. Com isso ele pode analisar modelos provenientes de diversas instalações simultaneamente.

#### 4.3.2. Segundo Teste

Como foi dito, o objetivo deste teste era levantar os parâmetros da equação paramétrica que melhor descreve a tubulação utilizada. No final, o valor do raio no modelo capturado deve ser próximo ao valor do modelo real.

A equação paramétrica de um cilindro é dada por:

$$p(s, \theta) = p_0 + s \text{ dir}Z + R \cos(\theta) \text{ dir}X + R \sin(\theta) \text{ dir}Y \quad (4.2)$$

Temos que  $p_0$  é um ponto por onde passa o eixo central, enquanto o vetor  $\text{dir}Z$  dá a direção deste mesmo eixo. O valor do raio é dado por  $R$  e os vetores

$dirX$ ,  $dirY$  e  $dirZ$  dão a direção do cilindro. A figura 36 ilustra os parâmetros da equação.

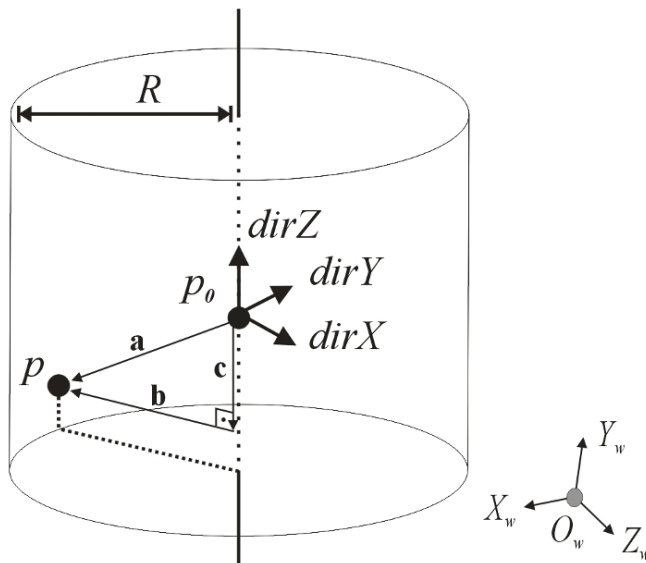


Figura 36 – Cilindro parametrizado.

Para cada cilindro devemos definir os parâmetros  $R$ ,  $p_0$  e  $dirZ$ . Os vetores  $dirX$ ,  $dirY$  e  $dirZ$  são ortogonais entre si, bastando então definir  $dirZ$ . Podemos escrever  $p_0$  e  $dirZ$  como:

$$p_0 = (a, b, c) \quad (4.3)$$

$$dirZ = (u, v, w) \quad (4.4)$$

Podemos eliminar mais alguns parâmetros. Podemos considerar que o ponto  $p_0$  pertence ao plano  $XZ$ . Logo  $b = 0$ . Se também considerarmos que o vetor  $dirZ$  não é normalizado, podemos escrever  $v = 1$ , obtendo assim:

$$p_0 = (a, 0, c) \quad (4.5)$$

$$dirZ = (u, 1, w) \quad (4.6)$$

Com isso, o eixo do cilindro não pode ser paralelo ao plano  $XZ$ . Então, dado um conjunto de pontos referente a um cilindro, gostaríamos de determinar os valores  $R$ ,  $a$ ,  $c$ ,  $u$  e  $w$  que melhor caracterizam este cilindro. Podemos empregar a mesma técnica utilizada para encontrar a melhor reta ajustada a um grupo de pontos. Ao invés de minimizar a equação (3.24), utilizamos uma outra equação que dá a distância do ponto ao eixo do cilindro, subtraída pelo valor do raio. Esta função calcula o erro de um ponto em relação ao cilindro determinado pelos parâmetros e é dada por:

$$\varepsilon(r_i) = \|\mathbf{b}\| - R \quad (4.7)$$

Observando a figura 4.10, temos que:

$$\|\mathbf{b}\| = \sqrt{\|\mathbf{a}\|^2 - \|\mathbf{c}\|^2} \quad (4.8)$$

Então a equação (4.7) pode ser reescrita como:

$$\varepsilon(r_i) = \sqrt{\|\mathbf{a}\|^2 - \|\mathbf{c}\|^2} - R \quad (4.9)$$

O vetor  $\mathbf{a}$  é dado por:

$$\mathbf{a} = p - p_0 = (x - a, y, z - c) \quad (4.10)$$

onde  $p = (x, y, z)$ . Então temos:

$$\|\mathbf{a}\|^2 = (x - a)^2 + y^2 + (z - c)^2 \quad (4.11)$$

O vetor  $\mathbf{c}$  é dado pelo vetor  $\mathbf{a}$  projetado na direção de  $dirZ$ . Então temos

$$\mathbf{c} = \left( \frac{\mathbf{a} \cdot dirZ}{dirZ \cdot dirZ} \right) dirZ = \left( \frac{(x - a, y, z - c) \cdot (u, 1, w)}{u^2 + 1 + w^2} \right) (u, 1, w) \quad (4.12)$$

Simplificando a norma, obtemos:

$$\|\mathbf{c}\|^2 = \frac{(u(x - a) + y + w(z - c))^2}{(u^2 + 1 + w^2)} \quad (4.13)$$

Para minimizar a soma dos erros para todos os pontos, utilizamos a biblioteca MINPACK [30], destinada a resolver sistemas de equações não lineares e problemas de mínimos quadrados não lineares. Há uma função que minimiza a soma dos quadrados de funções não lineares de  $n$  variáveis com o algoritmo de Levenberg-Marquardt modificado, sendo o usuário responsável por fornecer a função. No nosso caso, fornecemos a função que implemente o somatório da equação (4.9) aplicada a todos os pontos, com cinco variáveis correspondentes aos parâmetros do cilindro. O usuário também deve fornecer valores iniciais razoáveis para as variáveis, a fim de diminuir o número de chamadas à função.

Para calcular os valores iniciais, primeiramente aplicou-se o método dos mínimos quadrados para achar a melhor reta que passa pelos pontos fornecidos. A direção da reta é utilizada como a direção inicial do eixo do cilindro. Tomando esta direção achada como o novo eixo  $Y$  e realizando a respectiva transformação dos pontos, aplicamos o método Modified Least Squares, descrito em [21], para encontrar o melhor círculo que passa pelos pontos, ignorando as suas coordenadas do eixo  $Y$ . O valor inicial para o raio é igual ao raio do círculo encontrado.



Aplicando a transformação inversa ao centro do círculo calculado, obtemos o valor inicial de  $p_0$ . Estes valores iniciais calculados são muito próximos aos valores finais encontrados quando os pontos do modelo estão igualmente distribuídos ao longo do seu comprimento e curvatura.

O raio do tubo utilizado neste teste possui  $(150,3 \pm 0,5)mm$  de diâmetro médio, medido com um paquímetro em diferentes pontos. O tubo foi colocado em três diferentes posições em relação às câmeras para os testes na tabela abaixo. Cada um dos testes A, B e C referem-se a uma posição distinta. Para cada posição capturou-se a seqüência de imagens, referentes aos padrões projetados, uma vez. Utilizando esta seqüência, foram gerados oito diferentes modelos para cada posição. Cada modelo foi gerado duas vezes, uma para cada padrão de calibração. Os padrões eram reposicionados e as câmeras calibradas novamente para cada um dos modelos. Sempre a mesma seqüência de imagens era utilizada. Para isso, as câmeras permaneceram na mesma posição ao longo de cada teste. Também foram utilizados diferentes números de linhas codificadas na luz estruturada.

O objetivo deste procedimento era verificar como a posição do padrão influencia a calibração das câmeras, comparar os dois padrões e a precisão dos resultados. As tabelas completas com os resultados encontram-se no apêndice B. A figura 32b ilustra um dos modelos capturados já com os parâmetros do cilindro ajustados. O segmento de reta em vermelho representa o eixo do cilindro, enquanto que o arco, perpendicular ao eixo, possui o mesmo valor que o raio do cilindro calculado.

Testes	Padrão com Círculos		Padrão com Vértices	
	Diâmetro Médio (mm)	Desvio Padrão (mm)	Diâmetro Médio (mm)	Desvio Padrão (mm)
Teste A	150,730	2,473	149,389	1.636
Teste B	147,457	2.777	147.875	2.140
Teste C	153,904	1.036	152,49	1.887

Tabela 5 – Resultados obtidos pelos padrões de calibração no segundo teste.

Nas tabelas dos resultados finais obtidos nos testes A, B e C, a primeira observação é que a calibração é dependente da posição do padrão em relação ao objeto capturado e as câmeras. Na prática, observou-se que quanto mais perto do

centro do objeto capturado se posicionar o padrão de calibração, melhor são os resultados. O ângulo de inclinação do padrão em relação às câmeras também influencia na calibração.

Na tabela acima podemos observar que a posição do objeto em relação à câmera também influencia os resultados. Isso acontece, pois a função do OpenCV para calibração dos parâmetros intrínsecos, apesar de melhorar os resultados, não torna as imagens capturadas totalmente livres de distorções introduzidas pelas lentes. O OpenCV implementa o método de Zhang para calibração de câmera.

Também podemos observar que não há novamente uma diferença considerável entre os resultados alcançados pelos padrões de calibração. Ambos obtiveram resultados próximos para as três posições distintas da tubulação capturada. Possivelmente este é mais um indicativo da existência de distorções ópticas nas imagens capturadas.

Em relação aos diâmetros médios calculados, os resultados alcançados nos testes apresentam erros variando entre 0,3% e 2,4%. Entre os resultados individuais de cada modelo, os piores resultados foram obtidos pelos modelos B1, calibrado com o padrão de círculos, e o modelo C4, calibrado com o padrão de vértices. Os erros foram 3,6% e 3,5% respectivamente. Estes erros refletem principalmente a diferença na calibração devido ao posicionamento dos padrões de calibrações. Procurou-se neste teste testar uma variedade de posições diferentes para os padrões de calibração, de modo a avaliar a robustez da calibração. Os resultados mostram que a calibração deve ser melhorada.