

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
ESCOLA POLITÉCNICA
DEPARTAMENTO DE ELETRÔNICA E DE COMPUTAÇÃO

VISÃO ESTÉREO MULTIPLATAFORMA

Autor:

José Almeida Júnior

Orientador:

Prof. Mariane Rembold Petraglia, Ph.D.

Co-orientador:

Prof. José Gabriel R. C. Gomes, Ph.D.

Examinador:

Prof. Antônio Cláudio Gómez de Sousa, M.Sc.

DEL
JULHO DE 2008

À minha irmã.

AGRADECIMENTOS

Inicialmente, gostaria de agradecer aos meus pais, Ana Maria de Barros Dias Almeida e José Almeida Filho, pela educação que foi dada à mim e à minha irmã, pelo incentivo na busca dos nossos sonhos e ideais, pela liberdade na escolha e no traçado do nosso futuro, além de todo apoio nas decisões que tomamos diariamente.

Agradeço à minha irmã, Mariana Dias Almeida, a quem dedico este Projeto Final, por ser minha eterna companheira em todos os momentos da minha vida, sejam eles bons ou ruins, além de ser um exemplo de força, de dedicação e de que devemos sempre fazer aquilo que amamos, superando todas as barreiras e empecílios para isso.

À professora Mariane R. Petraglia por ter me aceito como aluno de iniciação científica, por confiar no meu trabalho e, sobretudo, por me oferecer trabalhar com processamento de imagem, o que resultou em uma nova paixão na minha vida.

Agradeço ao professor José Gabriel R. C. Gomes por ter me ajudado nas várias dúvidas que surgiram durante a realização deste projeto e por ter me incentivado na continuação da minha formação com o mestrado.

Ao professor Joarez Bastos por ter me indicado à iniciação científica com a professora Mariane.

Ao amigo Guilherme Filgueiras por ter me apresentado e ensinado a linguagem de programação C# (C Sharp), além de me ajudar nas várias dúvidas relativas à programação.

À toda equipe do laboratório PADS por oferecer toda a estrutura e suporte para este projeto. Ao Rafael Szendrodi por dar todo o suporte na instalação dos *softwares* no laboratório para a realização deste projeto e por manter tudo funcionando adequadamente.

À minha grande amiga Zoraide, por todo apoio, ensinamento e ajuda, além das lições de vida por ela transmitidas.

E, por fim, aos meus amigos Guilherme Filgueiras, Rafael Camel e Caio Amim que me acompanharam durante toda a jornada da graduação.

Resumo do Projeto Final apresentado Escola Politécnica - UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro Eletrônico e de Computação

VISÃO ESTÉREO MULTIPLATAFORMA

José Almeida Júnior

Julho/2008

Orientador: Mariane Rembold Petraglia

Co-orientador: José Gabriel R. C. Gomes

Departamento de Engenharia Eletrônica e de Computação

Este projeto tem como objetivo a adaptação de um *software* de Visão Estéreo utilizando a linguagem de programação C# (CSharp) e baseado no *software* desenvolvido no Laboratório de Processamento Analógico e Digital de Sinais (PADS). A principal característica desse novo *software* será a sua execução em várias plataformas operacionais graças a utilização do Mono Project. Este fornece o *software* necessário para desenvolver e executar aplicações .Net em Linux, Solaris, Mac OS X, Windows e Unix. Patrocinado pela Novell, o Mono Project é um *open source* com ativa cooperação coletiva de seus usuários.

Palavras Chave: Visão Estéreo, Visão Computacional, Calibração de Câmeras, Pontos Homólogos, Reconstruction of 3D Coordinates.

Abstract of Final Project presented to Escola Politécnica - UFRJ as a partial fulfillment of the requirements for the degree of Eletronic and Computer Engineer.

MULTIPLATFORM STEREO VISION

José Almeida Júnior

July/2008

Advisor: Mariane Rembold Petraglia

Co-advisor: José Gabriel R. C. Gomes

Eletronic and Computer Engineering Departament

This work describes the design adaptation of a Stereo Vision software developed by Laboratory of Processing Analog and Digital Signals (PADS) using C# (CSharp) programming language. The main characteristic of this new software is that it could be executed in diferent operating systems because of the use of Mono Project. Mono provides the necessary software to develop and run .NET client and server applications on Linux, Solaris, Mac OS X, Windows, and Unix. Sponsored by Novell, the Mono open source project has an active and enthusiastic contributing community

Index Terms: Stereo Vision, Computational Vision, Cameras Calibration, Homolog Points, Integrated Circuit.

SUMÁRIO

1	Introdução	1
2	Sistema de Visão Estéreo e Requisitos Funcionais do Software	3
2.1	Introdução	3
2.2	Estereofotogrametria	3
2.3	Casos de Uso	7
2.4	Diagrama de Classes	12
3	Descrição do Software Desenvolvido	18
3.1	Introdução	18
3.2	Imagens	18
3.2.1	Carregando as Imagens	18
3.2.2	Captura e Exibição das Coordenadas da Imagem	23
3.3	Tabelas	24
3.3.1	Estrutura	24
3.3.2	Operações	24
3.3.3	Visualização	25
3.4	Exportando as Tabelas	26
3.5	Importando as Tabelas	27
3.6	Matrizes	29
3.7	Calibração e Dimensionamento	30
4	Manual do Usuário	31
4.1	Introdução	31
4.2	Configurações do Sistema	31
4.2.1	Plataformas Suportadas	31
4.2.2	Sistemas Operacionais	31
4.2.3	Computador: Configuração Mínima	32
4.3	O Software	32
4.3.1	Janela Principal	33
4.3.1.1	Menu <i>Arquivo</i>	34

4.3.1.2	Menu <i>Calibração</i>	34
4.3.1.3	Menu <i>Dimensionar</i>	35
4.3.2	Calibrando o Sistema	37
4.3.2.1	Importar Tabela	38
4.3.2.2	Exportar Tabela	39
4.3.2.3	Calibrar	40
4.3.2.4	Fechar	40
4.3.3	Dimensionar	40
4.3.3.1	Importar Tabela	41
4.3.3.2	Exportar Tabela	42
4.3.3.3	Fechar	42

5 Conclusão 43

CAPÍTULO 1

INTRODUÇÃO

No ano de 2005, o Programa de Engenharia Elétrica e a PETROBRAS/GDS/MIS iniciaram um projeto de pesquisa sobre o levantamento e inspeção submarina em três dimensões (Projeto COPPETEC PEE-5799). O mesmo foi desenvolvido utilizando algoritmos de estereofotogrametria desenvolvidos no Laboratório de Processamento Analógico e Digital de Sinais (PADS), localizado na sala H-320 do Centro de Tecnologia da Universidade Federal do Rio de Janeiro (UFRJ). O resultado da pesquisa foi o desenvolvimento de um sistema de aquisição, processamento e reconstrução de imagens estéreo.

Um dos componentes do sistema é um *software*, desenvolvido na linguagem C++, o qual funciona na plataforma *Windows* e permite a visualização e captura de imagens, em tempo real, de duas câmeras localizadas em um veículo remotamente operado. Este *software* realiza, utilizando o par estéreo de imagens, a estimação de coordenadas tridimensionais, permitindo, por conseguinte, o cálculo de dimensionamento de objetos de interesse.

Este projeto tem como objetivo a adaptação do *software* de Visão Estéreo, desenvolvido originalmente em C para Linux e C++ para Windows no PADS, utilizando agora a linguagem de programação C# (C Sharp). A principal característica desse novo *software* será a sua execução em várias plataformas operacionais graças à utilização do Mono Project [1]. Este novo *software* será nomeado de Visão Estéreo Multiplataforma (VEM).

O Mono Project fornece o *software* necessário para desenvolver e executar aplicações .Net em Linux, Solaris, Mac OS X, Windows e Unix. Patrocinado pela Novell, o Mono Project é um projeto *open source* com ativa cooperação coletiva de seus usuários.

Dentre as principais funções executadas pelo *software* VEM, destacam-se:

- Visualização de imagens estéreo;
- Calibração de imagens estéreo;
- Dimensionamento de elementos visíveis no par de imagens;

- Importação e exportação de tabelas de calibração e dimensionamento.

A linguagem C# foi escolhida pois a mesma é de fácil aprendizagem e uso, robusta e com boa performance. Além disso, em conjunto com os demais recursos da arquitetura .Net, o C# também é a linguagem ideal para a criação de uma nova categoria de programas que aproveitam as oportunidades trazidas da internet. Para o desenvolvimento do *software* será utilizado o Visual C# 2005 Express Edition, o qual é fornecido e registrado gratuitamente na internet. A compilação das linhas de código será realizada pelo Mono Project.

Esta documentação está dividida em cinco capítulos. O Capítulo 2 aborda a teoria que envolve o processo de estereofotogrametria e os diagramas de Caso de Uso e de Classes, referentes aos requisitos funcionais do *software*. O Capítulo 3 explica as principais etapas de desenvolvimento do *software*, do projeto em questão, de forma que o mesmo possa ser implementado em outras linguagens de programação, se necessário. O Capítulo 4 exhibe os resultados obtidos neste projeto através do Manual do Usuário, explicando a operacionalidade e as características do *software*, além das diferentes funções disponíveis para o usuário, e ilustrações relativas à interface gráfica. Por fim, o Capítulo 5 apresenta as conclusões obtidas.

CAPÍTULO 2

SISTEMA DE VISÃO ESTÉREO E REQUISITOS FUNCIONAIS DO SOFTWARE

2.1 - INTRODUÇÃO

Para um melhor entendimento do objetivo e desenvolvimento deste projeto, são necessários alguns conceitos e especificações. Desta forma, este capítulo tem como objetivo explicar, de forma clara e simples, a teoria que envolve o processo de realização de medições através de pares estéreo de fotografias (estereofotogrametria) e os requisitos funcionais (Diagrama de Casos de Uso e Diagrama de Classes) do *software* a ser desenvolvido.

2.2 - ESTEREOFOTOGRAMETRIA

Etimologicamente, a palavra "fotogrametria" vem do grego *photon* - luz, *graphos* - descrição e *metron* - medições, podendo ser interpretada como medições realizadas através de fotografias [2]. De forma mais abrangente, pode-se definir fotogrametria como a ciência e tecnologia de se reconstruir o espaço tridimensional (ou parte do mesmo) a partir de imagens bidimensionais, utilizando fotografias capturadas no mesmo instante por câmeras calibradas em posições diferentes, sem contato físico com o objeto ou alvo de interesse. Já o termo "estereofotogrametria" refere-se à aplicação da fotogrametria com a utilização de pares estéreo de imagens [3].

Para a determinação de coordenadas tridimensionais, necessárias para a realização da fotogrametria, utilizam-se modelos matemáticos, constituindo a fotogrametria analítica. Ou seja, a fotogrametria analítica é formada por um conjunto de técnicas pelas quais são feitas deduções sobre a posição tridimensional, orientação e dimensões de partes do objeto tridimensional, observado em um referencial global, a partir de coordenadas das projeções em perspectiva bidimensional.

Através de uma translação, seguida de uma rotação, pode ser obtida a relação entre o referencial da câmera e o referencial global. Inicialmente, através de uma translação, um ponto $x = [x \ y \ z]^T$ no referencial global é representado em relação à posição da lente (Figura 2.1).

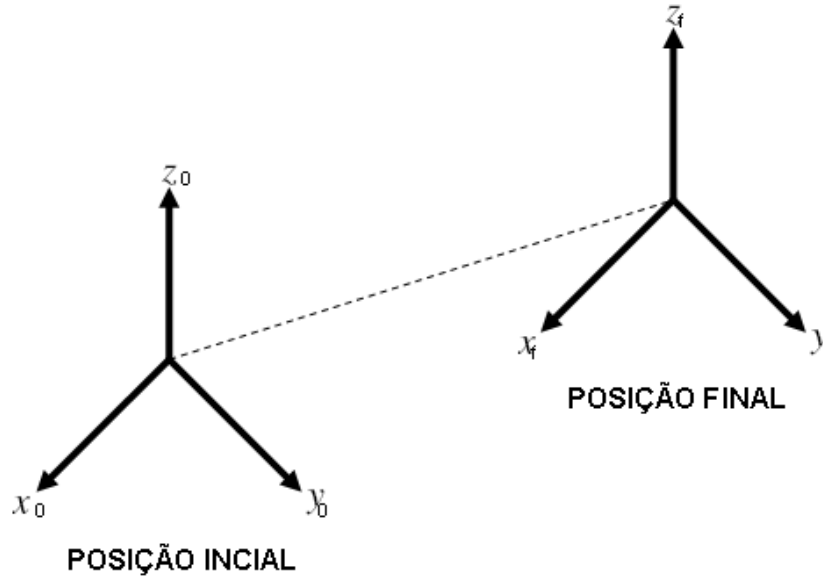


Figura 2.1: Translação entre dois referenciais

Em seguida, uma rotação no referencial global, em relação ao referencial da câmera, deve ser realizada, pois as direções x , y , z desta podem diferir das da primeira. Por ser um processo não comutativo, uma convenção foi adotada para a matriz de rotação: primeiramente é realizada uma rotação em relação ao eixo x ; em seguida, uma rotação em relação ao eixo y ; por fim, rotaciona-se o eixo z (Figura 2.2).

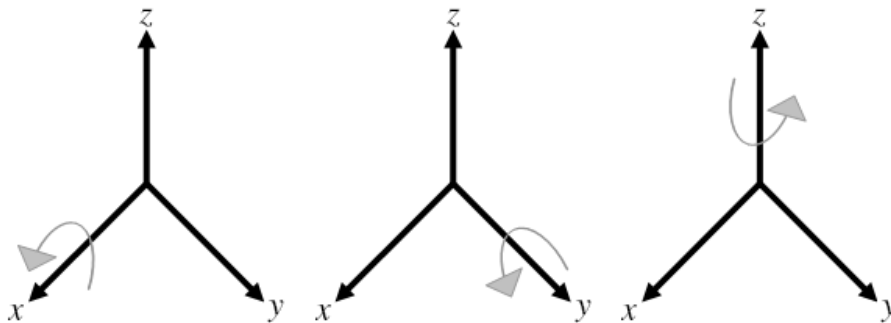


Figura 2.2: Rotação entre dois referenciais

Em resumo, um ponto $[x \ y \ z]^T$ localizado no referencial do objeto é representado pelo ponto $[p \ q \ s]^T$ no referencial da câmera da seguinte forma:

$$\begin{bmatrix} p \\ q \\ s \end{bmatrix} = \mathbf{R}(\omega, \phi, \kappa) \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix}, \quad (2.1)$$

onde ω , ϕ e κ são os ângulos da matriz \mathbf{R} de rotação.

Pode ser obtida a projeção em perspectiva 2D para o ponto 3D no referencial da câmera a partir da representação anterior. Essa projeção é dada por:

$$\begin{bmatrix} s \\ v \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + \frac{1}{s} \begin{bmatrix} \alpha_u p \\ \alpha_v q \end{bmatrix}, \quad (2.2)$$

onde $[u_0 \ v_0]^T$ são as coordenadas do ponto onde o eixo óptico da câmera intersecta o plano da imagem e α_u e α_v são os fatores de escala horizontal e vertical para a determinação de f , o qual é a distância do plano de projeção da imagem à lente (relacionada à distância focal)(Figura 2.3).

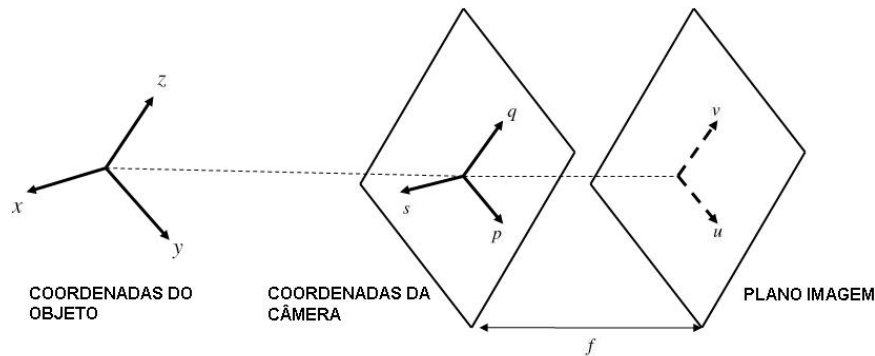


Figura 2.3: Sistema de coordenadas

Os parâmetros x_0 , y_0 , z_0 , ω , ϕ e κ são parâmetros de orientação exterior, enquanto u_0 , v_0 e f são parâmetros de orientação interior. Para obtê-los com precisão, e assim ser possível a realização da estereofotogrametria, são necessários um conjunto suficiente de pontos 3D, suas respectivas projeções em 2D, além de uma solução inicial aproximada.

A orientação exterior é especificada por parâmetros que determinam a posição da câmera em relação ao referencial global e requer a determinação de três parâmetros de translação (x_0 , y_0 , z_0) e três ângulos de rotação (ω , ϕ , κ). A orientação interior é especificada por: uma constante da câmera f , relativa à distância focal e determinada por α_u e α_v ; o ponto u_0, v_0 de interseção do eixo óptico com o plano da imagem; e as características de distorção da lente. Desta forma, tem-se um total de 10 parâmetros e, uma vez que são utilizadas duas imagens (um par estéreo), são necessários 20 parâmetros para a descrição do sistema de câmeras a ser usado para estereofotogrametria.

Para determinar os parâmetros (exteriores e interiores) de calibração, um sistema de coordenadas globais pode ser elaborado para facilitar a escolha dos pontos 3D. Os dados desse sistema são coletados de um *grid* de calibração. Este é constituído por dois planos perpendiculares e possui diversos alvos (Figura 2.4), os quais têm a distância medida entre seu centro e um ponto de referência escolhido do *grid* para a determinação das coordenadas x , y e z de cada ponto 3D. A Figura 2.5 mostra o *grid* de calibração, com seus alvos, inserido em um tanque de simulação submarina.



Figura 2.4: Alvos do grid de calibração

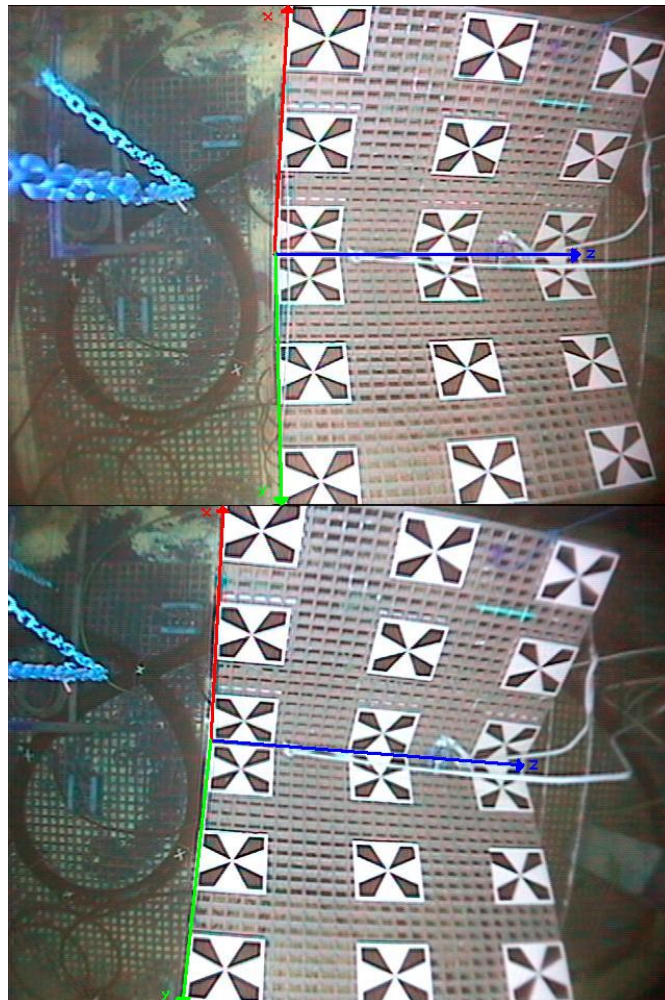


Figura 2.5: Grid de calibração dentro de um tanque de simulação

Os pontos 2D, referentes a cada um dos pontos 3D anteriormente escolhidos, são selecionados através do *software* para cada uma das imagens (esquerda e direita). Após a seleção, o sistema já pode ser calibrado.

A solução inicial aproximada tem como objetivo possibilitar uma rápida convergência do algoritmo de otimização não-linear, o qual é responsável pelo aumento da precisão no procedimento de calibração. Desta forma, a solução inicial apresenta uma estimativa da posição do mínimo global a ser calculado pelo método não-linear de mínimos quadrados iterativo [3].

2.3 - CASOS DE USO

Um caso de uso é uma descrição narrativa de uma sequência de eventos que ocorre quando um ator (agente externo) usa um sistema para realizar uma tarefa [4]. A Figura 2.6 exibe o diagrama de casos de uso deste projeto.

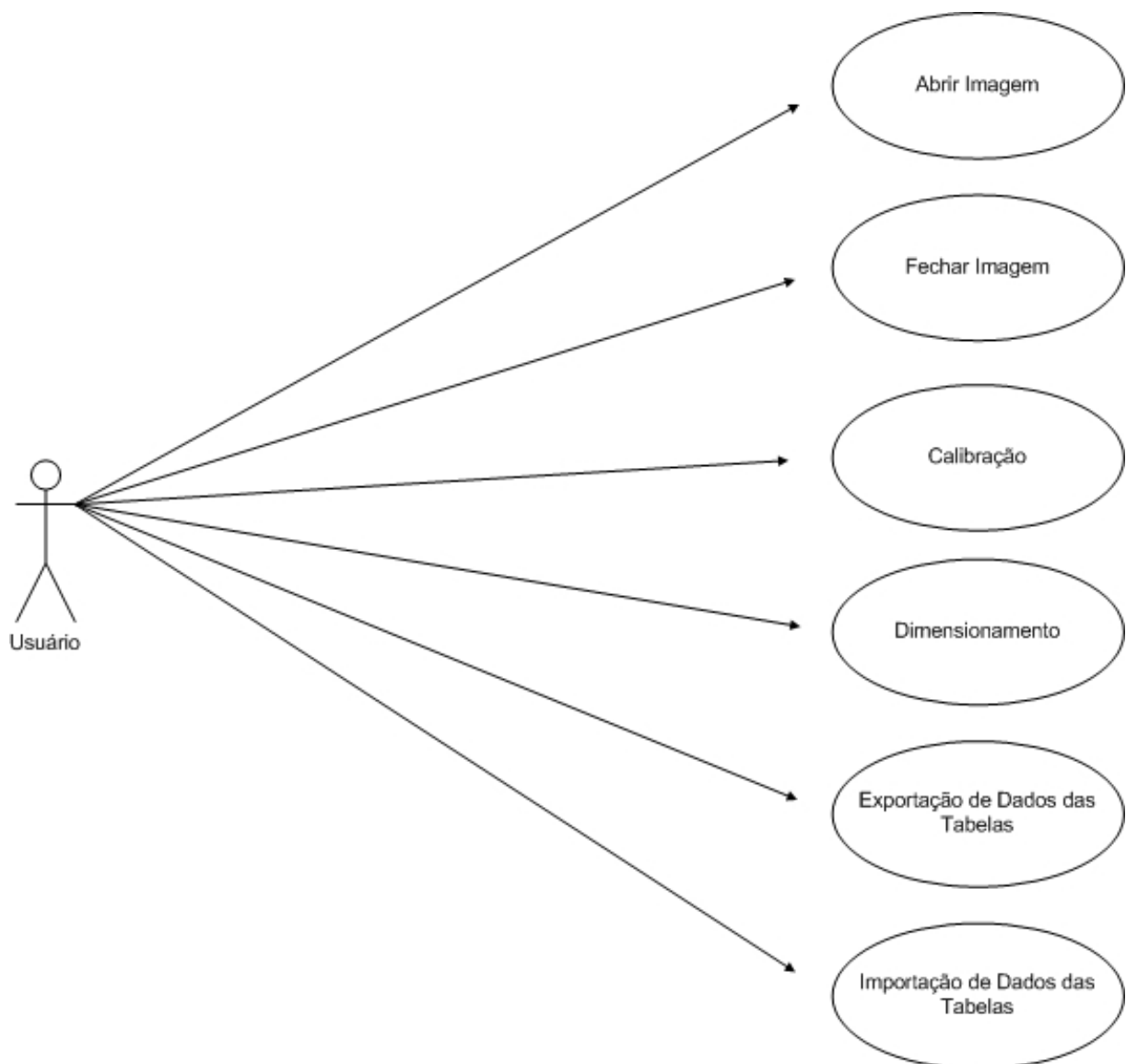


Figura 2.6: Diagrama de casos de uso

Abaixo, seguem as descrições de todos os casos de uso.

Tabela 2.1: Caso de Uso Abrir Imagem

Ator Primário	Usuário
Pré-Condição	Executar o <i>software</i>
Pós-Condição	Imagem sendo exibida

Fluxo principal Tabela 2.2 da Tabela 2.1:

Tabela 2.2: Fluxo Principal do Caso de Uso Abrir Imagem

Ator	Sistema
Solicita a opção de abertura de uma imagem	
	Solicita a escolha do arquivo que o usuário deseja abrir
Seleciona a imagem que deseja visualizar	
	Abre e exibe a imagem selecionada

Tabela 2.3: Caso de Uso Fechar Imagem

Ator Primário	Usuário
Pré-Condição	Abrir uma imagem
Pós-Condição	Fim da exibição da imagem anteriormente aberta

Fluxo principal Tabela 2.4 da Tabela 2.3:

Tabela 2.4: Fluxo Principal do Caso de Uso Fechar Imagem

Ator	Sistema
Solicita a opção para fechar a imagem que está sendo exibida	
	A imagem deixa de ser exibida

Tabela 2.5: Caso de Uso Calibração

Ator Primário	Usuário
Pré-Condição	Abrir uma imagem
Pós-Condição	Calibração do sistema realizada ou não

Fluxo principal Tabela 2.6 da Tabela 2.5:

Tabela 2.6: Fluxo Principal do Caso de Uso Calibração

Ator	Sistema
Seleciona a opção para calibrar o sistema	
	Verifica se o <i>status</i> do mesmo é "calibrado" ou "não calibrado"
	Se o <i>status</i> for "não calibrado", o usuário pode realizar a calibração (ir para o caso onde as coordenadas dos pontos escolhidos são exibidas em uma tabela)
	Se o <i>status</i> for "calibrado", uma pergunta é exibida para o usuário sobre a manutenção ou não da calibração atual
Caso deseje manter a calibração atual, o sistema não realiza uma nova calibração, encerrando o caso de uso	
Caso deseje realizar uma nova calibração, a calibração anterior é removida e o usuário poderá realizar uma nova calibração do sistema	
	As coordenadas dos pontos escolhidos são exibidas em uma tabela
	A partir das coordenadas dos pontos escolhidos pelo usuário e através de algoritmos que compõem o sistema, realiza-se a calibração
	É informado o percentual de erro médio da calibração, terminando o caso de uso

Tabela 2.7: Caso de Uso Dimensionamento

Ator Primário	Usuário
Pré-Condição	Calibração do sistema realizada
Pós-Condição	Exibição da distância entre os pontos selecionados

Fluxo principal Tabela 2.8 da Tabela 2.7:

Tabela 2.8: Fluxo Principal do Caso de Uso Dimensionamento

Ator	Sistema
Solicita a opção para dimensionamento no <i>menu</i>	
	Montagem da tabela de dimensionamento (inicialmente escondida)
Seleciona os pontos de interesse para a realização do dimensionamento	
	A partir das coordenadas dos pontos escolhidos e através de algoritmos que compõe o sistema, é exibida a distância espacial entre os pontos selecionados

Tabela 2.9: Caso de Uso Exportar Dados da Tabela

Ator Primário	Usuário
Pré-Condição	Preenchimento da tabela
Pós-Condição	Tabela exportada em um arquivo texto

Fluxo principal Tabela 2.10 da Tabela 2.9:

Tabela 2.10: Fluxo Principal do Caso de Uso Exportar Dados da Tabela

Ator	Sistema
Solicita a opção para exportar a tabela desejada (tabela de calibração e/ou tabela de dimensionamento)	
	Solicita que o usuário determine o <i>path</i> onde o arquivo texto, referente à tabela, será salvo e que escolha obrigatoriamente um nome para o arquivo
Escolhe o nome para o arquivo texto e confirma a execução da operação	
	Verifica se existe algum outro arquivo texto com o mesmo nome escolhido pelo usuário
	Caso não exista, o arquivo texto é salvo
	Caso exista, informa-se a situação para o usuário e o mesmo poderá decidir se deseja ou não sobrescrever o arquivo já existente
Se desejar sobrescrever, o arquivo já existente terá todo seu conteúdo apagado e, em seguida, preenchido com as novas informações provenientes da tabela	
Se não desejar sobrescrever, deve escolher um novo nome para o arquivo	

Tabela 2.11: Caso de Uso Importar Dados da Tabela

Ator Primário	Usuário
Pré-Condição	Opção de calibração ou dimensionamento selecionada
Pós-Condição	Exibição das informações contidas no arquivo texto na respectiva tabela

Fluxo principal Tabela 2.12 da Tabela 2.11:

Tabela 2.12: Fluxo Principal do Caso de Uso Importar Dados da Tabela

Ator	Sistema
Solicita a opção para importar a tabela desejada (tabela de calibração e/ou tabela de dimensionamento)	
	Solicita que o usuário selecione o arquivo que deseja abrir
Seleciona o arquivo de interesse	
	Abre o arquivo e exibe, nos devidos espaços existentes na tabela, os dados do mesmo

2.4 - DIAGRAMA DE CLASSES

Foi utilizada uma ferramenta existente no Microsoft Visual Studio 2005 que permite a representação do Diagrama de Classes com todas as classes utilizadas no projeto. Seguem abaixo, nas Figuras 2.7 a 2.10, primeiramente as classes separadas e sua características. Por último, na Figura 2.11, segue a Diagrama de Classes do projeto.

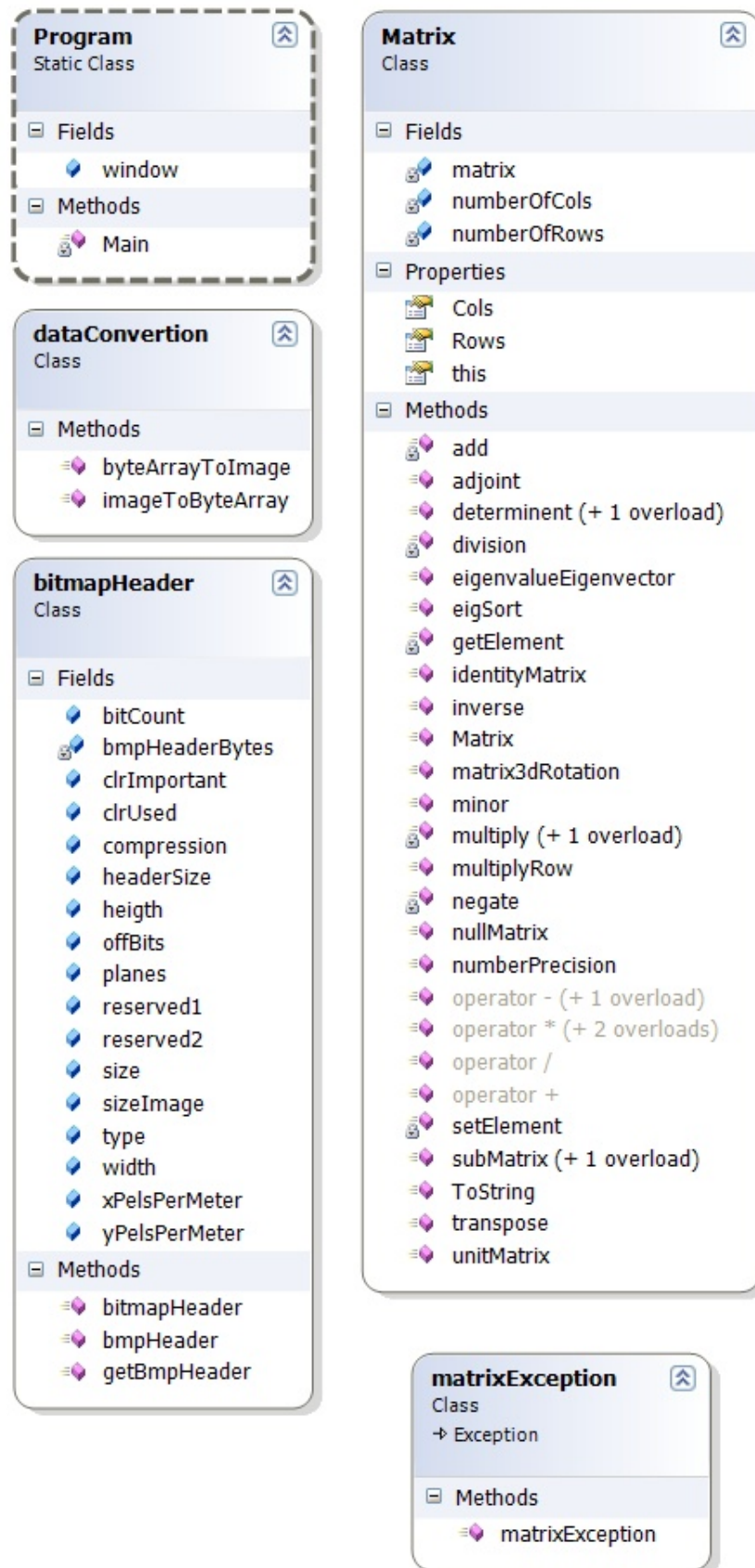


Figura 2.7: As classes utilizadas nesse projeto e suas características

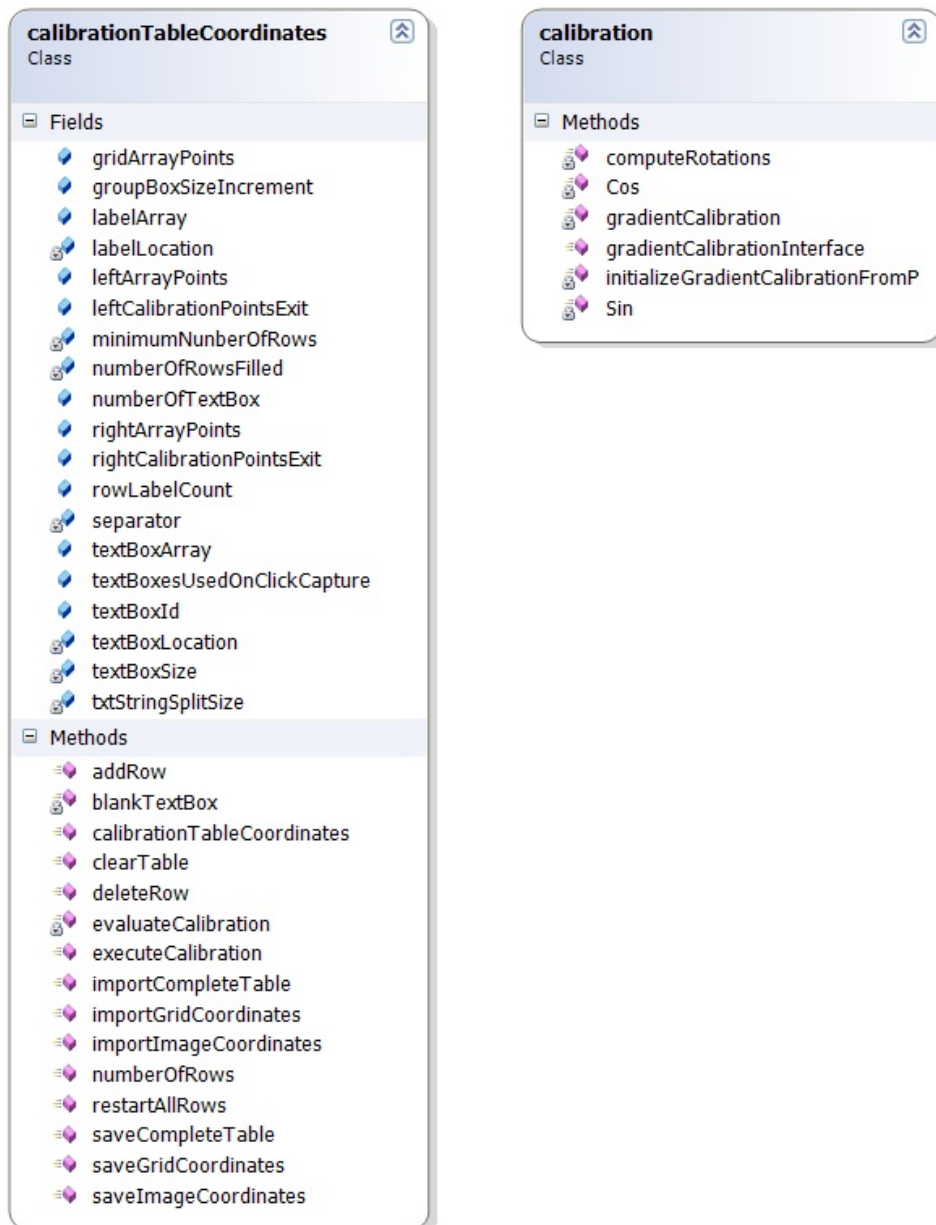


Figura 2.8: As classes relativas à calibração do sistema e à tabela de calibração

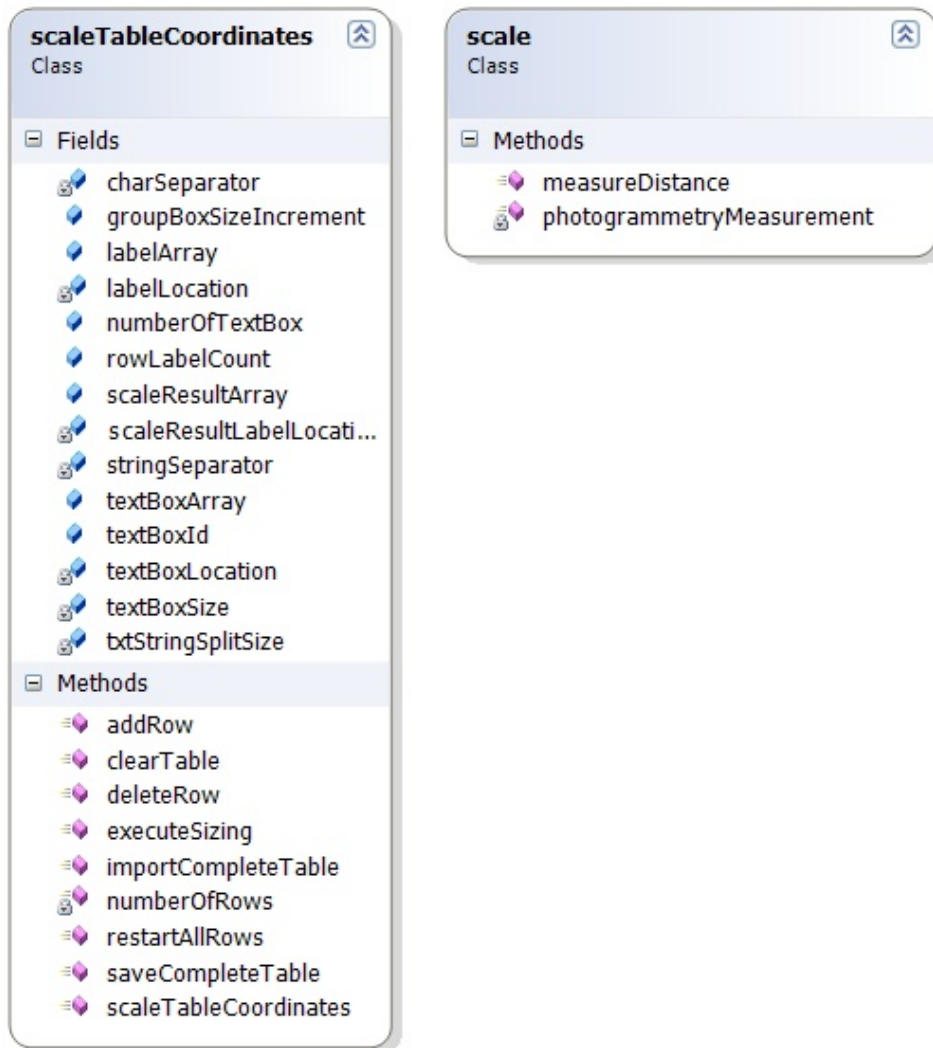


Figura 2.9: As classes relativas ao dimensionamento do sistema e à tabela de dimensionamento

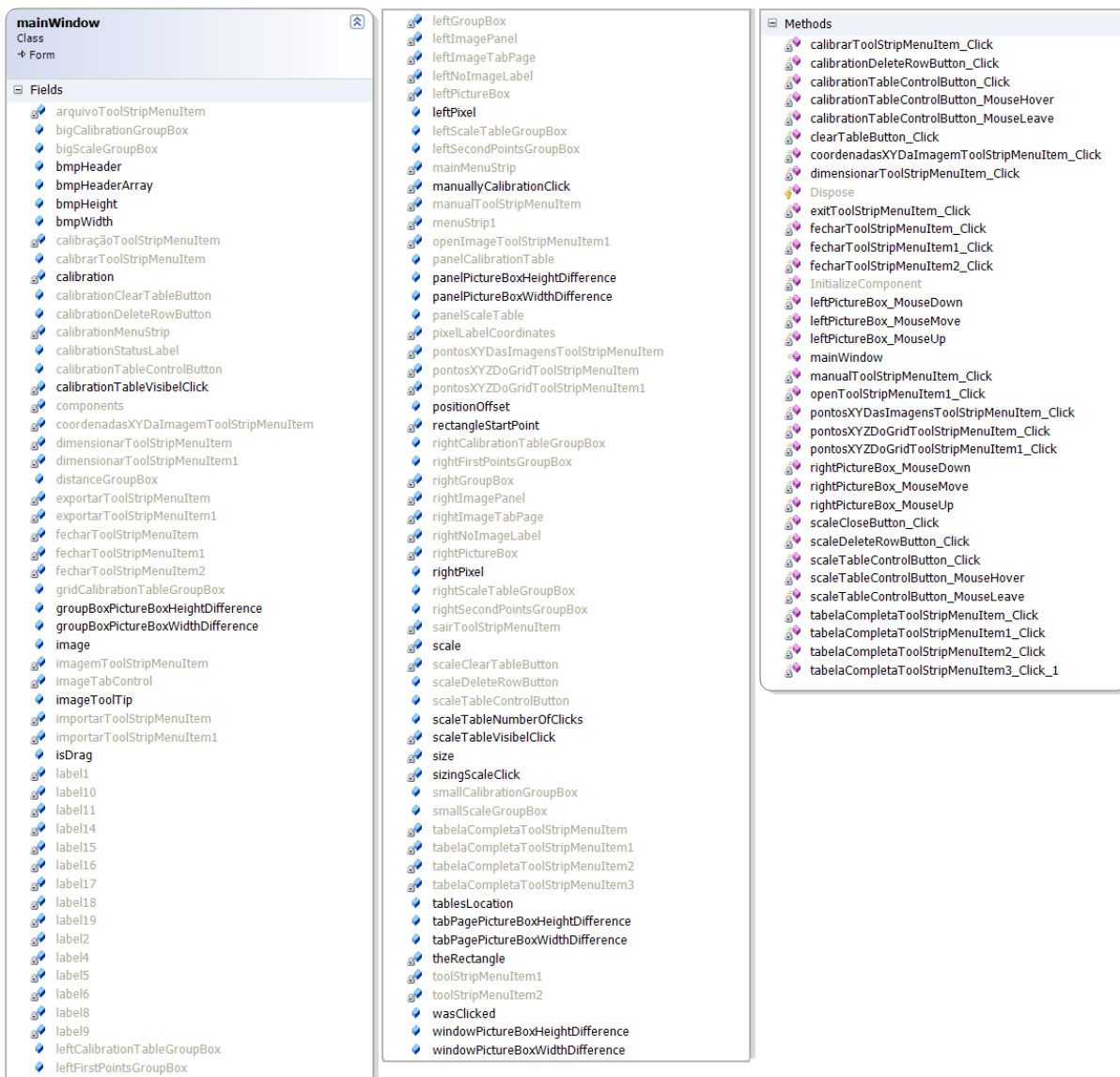


Figura 2.10: A classe mainWindow e suas características

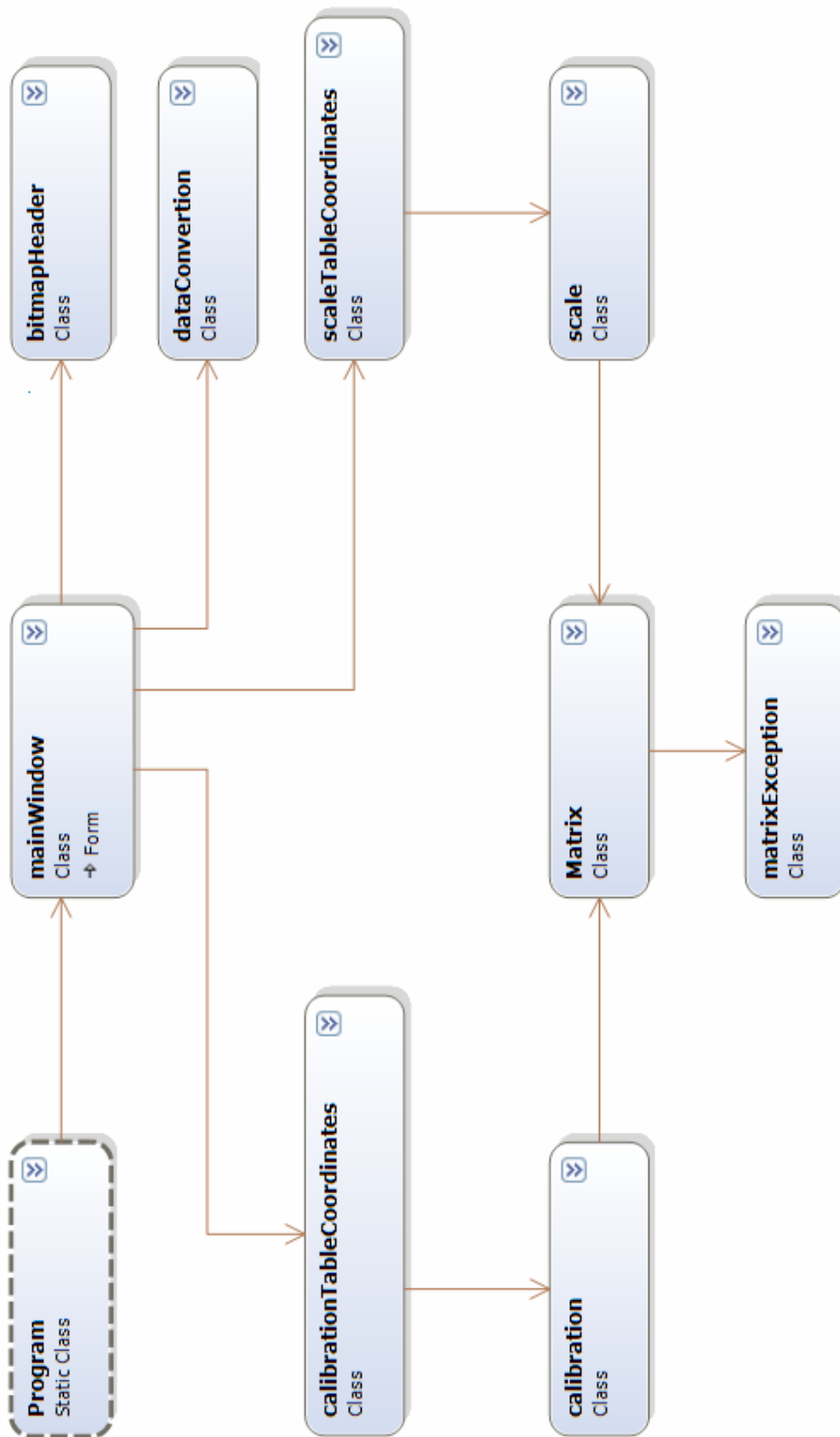


Figura 2.11: Diagrama de classes

CAPÍTULO 3

DESCRIÇÃO DO SOFTWARE DESENVOLVIDO

3.1 - INTRODUÇÃO

O desenvolvimento do *software* deste projeto envolve a criação de várias classes e de uma interface gráfica que tornem esta nova versão do programa mais simples e mais fácil de ser utilizada. Dessa forma, este capítulo tem como objetivo explicar as principais etapas de desenvolvimento deste *software* de forma que o mesmo possa ser implementado em outras linguagens de programação, se necessário.

3.2 - IMAGENS

3.2.1 - CARREGANDO AS IMAGENS

O primeiro passo para o desenvolvimento do *software* foi estabelecer que o mesmo pudesse exibir as imagens escolhidas pelo usuário. Estas são no formato *Bitmap* (.bmp) 24 bits e são constituídas por duas imagens unidas verticalmente: a imagem relativa à câmera esquerda acima da imagem relativa à câmera direita (Figura 3.1).



Figura 3.1: Bitmap com as imagens relativas à câmera esquerda e direita unidas verticalmente

Inicialmente, através de uma opção no *menu*, é exibido um *Open File Dialog*, que permitirá ao usuário selecionar a imagem que será aberta. Para ajudar na visualização dos arquivos, pode ser utilizado um filtro no *Open File Dialog* para que apenas os arquivos com extensão *.bmp* sejam exibidos. Caso contrário, permite-se a exibição de qualquer extensão (Tabela 3.1).

Tabela 3.1: Linhas de código sobre o *Open File Dialog*

```
OpenFileDialog openImage = new OpenFileDialog();  
System.IO.Stream file;  
  
openImage.Filter = "Imagem (*.bmp)|*.bmp|Todos os Arquivos|*.*";  
file = openImage.OpenFile();
```

Após a escolha, todos os dados existentes no arquivo .bmp são convertidos para bytes para que possam ser devidamente interpretados e manipulados pelo *software* (Tabela 3.2).

Tabela 3.2: Linhas de código para conversão do arquivo .bmp para bytes

```
public static byte[] imageToByteArray(System.Drawing.Image imageIn)
{
    MemoryStream ms = new MemoryStream();
    imageIn.Save(ms, System.Drawing.Imaging.ImageFormat.Bmp);
    return ms.ToArray();
}
```

Após a conversão, as informações relativas ao *header* do *bitmap* são tratadas pelo *software* para a correta exibição das imagens esquerda e direita separadas.

Todo arquivo *bitmap* é dividido em três ou quatro partes:

1. Cabeçalho de Arquivo: contém a assinatura do arquivo e informações sobre o tamanho e *layout* do BMP (disposição dos dados dentro do arquivo). Tamanho: 14 bytes (Tabela 3.3).
2. Cabeçalho de Mapa de Bits: contém as informações da imagem contida no arquivo. Define as dimensões, tipo de compressão (se houver) e informações sobre as cores da imagem. Tamanho: 40 bytes (Tabela 3.4).
3. Paleta ou Mapa de Cores (opcional): somente estará presente em arquivos de imagem que utilizam 16 ou 256 cores. Nas demais versões de *bitmap*, vem diretamente o item a seguir. Tamanho: (4 bytes × Número de Cores) (Tabela 3.5).
4. Área de Dados da Imagem Contida no Arquivo: dados que permitem a exibição da imagem (dados dos pixels a serem exibidos). Pode haver compressão. Tamanho: campo *SizeImage* do cabeçalho de informações da imagem.

Tabela 3.3: Estrutura do Cabeçalho de Arquivo

Campo	Bytes	Descrição
Type	2	Assinatura do arquivo: os caracteres ASCII "BM" ou $(424D)_h$ (identificação de ser realmente <i>bitmap</i>)
Size	4	Tamanho do arquivo em bytes
Reserved1	2	Campo reservado 1. Deve ser ZERO
Reserved2	2	Campo reservado 2. Deve ser ZERO
OffBits	4	Deslocamento em bytes para o início da área de dados da imagem (Item 4) a partir do início deste cabeçalho. - Se a imagem usa paleta (Item 3), este campo tem tamanho igual a: $14 + 40 + (4 \times \text{Número de Cores})$ - Se a imagem for <i>true color</i> , este campo tem tamanho igual a: $14 + 40 = 54$ bytes

Tabela 3.4: Estrutura do Cabeçalho de Mapa de Bits

Campo	Bytes	Descrição
Size	4	Tamanho deste cabeçalho (40 bytes). Sempre $(28)_h$
Width	4	Largura da imagem em <i>pixels</i>
Height	4	Altura da imagem em <i>pixels</i>
Planes	2	Número de planos da imagem. Sempre 1
BitCount	2	Quantidade de bits por <i>pixel</i> (1, 4, 8, 24, 32)
Compression	4	Compressão usada. - 0 = BI_RGB (sem compressão) - 1 = BI_RLE8 (compressão RLE 8 bits) - 2 = BI_RLE4 (compressão RLE 4 bits)
SizeImage	4	Tamanho da imagem (dados) em bytes
xPelsPerMeter	4	Resolução horizontal em <i>pixels</i> por metro
yPelsPerMeter	4	Resolução vertical em <i>pixels</i> por metro
ClrUsed	4	Número de cores usadas na imagem
ClrImportant	4	Número de cores importantes na imagem

Tabela 3.5: Estrutura da Paleta

Campo	Bytes	Descrição
Blue	1	Intensidade do Azul. De 0 a 255
Green	1	Intensidade do Verde. De 0 a 255
Red	1	Intensidade do Vermelho. De 0 a 255
Reservado	1	Campo reservado. Deve ser ZERO sempre

Os bytes referentes ao cabeçalho são separados dos demais. Em seguida, as informações contidas no mesmo são separadas e tratadas, como exibido nas Tabelas 3.3 a 3.5.

Com exceção dos dois primeiros bytes relativos à assinatura do arquivo, os demais são armazenados no formato Intel LSB (*Least Significant Bits*), sendo o primeiro byte o menos significativo e o último o mais significativo (Figura 3.2). Assim, operações de deslocamentos (*shift*) à esquerda devem ser feitas para que as informações existentes no cabeçalho contenham a representação hexadecimal dos números na forma correta. O exemplo abaixo (Tabela 3.6) refere-se ao *shift* relativo aos quatro bytes do campo Size da Tabela 3.4.

Tabela 3.6: *Shift* para a esquerda

```
header[2] = (byte)(size & (0x000000ff));
header[3] = (byte)((size & (0x0000ff00)) >> 8);
header[4] = (byte)((size & (0x00ff0000)) >> 16);
header[5] = (byte)((size & (0xff000000)) >> 24);
```



Figura 3.2: Representação dos bytes

Com as informações do cabeçalho devidamente organizadas e separadas, deve ser observado que, como o *bitmap* selecionado é constituído pelas imagens esquerda e direita das câmeras, a altura do mesmo será o dobro da altura de cada uma das duas imagens. Além disso, como os *bitmaps* utilizados estão no formato 24 bits (*true color*), cada sequência de 3 bytes na Área de Dados da Imagem corresponde a uma sequência RGB (composição da cor do *pixel*). Assim, o número de bytes correspondentes aos dados de exibição da imagem será:

$$\text{Altura da Imagem} \times \text{Largura da Imagem} \times 3 \quad (3.1)$$

Dessa forma, mesmo que o *bitmap* contenha possíveis informações inseridas após o término da Área de Dados da Imagem, apenas os bytes referentes aos *pixels* serão utilizados.

Os dados localizados na Área de Dados da Imagem são armazenados seqüencialmente da última linha para a primeira. Ou seja, o primeiro *pixel* será o inferior esquerdo, enquanto que o último será o superior direito. Assim, a metade do número de bytes da Equação (3.1) refere-se à imagem relativa à câmera direita, enquanto que a outra metade refere-se à imagem relativa à câmera esquerda.

As informações referentes à altura e à largura do *bitmap* também são utilizadas para dimensionar a janela onde serão exibidas as imagens devidamente separadas das câmeras. Caso a resolução da imagem seja maior que a utilizada pelo usuário em seu computador, a janela de exibição assumirá o maior tamanho possível e barras de rolagem (*scroll*) serão mostradas para poder visualizar as porções da imagem que não estão sendo mostradas no momento.

3.2.2 - CAPTURA E EXIBIÇÃO DAS COORDENADAS DA IMAGEM

Durante a realização da calibração do sistema e do dimensionamento, é muito importante informar ao usuário quais são as coordenadas dos *pixels* nas imagens quando move-se o cursor do *mouse* sobre as mesmas. Assim, além de mostrar as imagens, a região onde as mesmas estão localizadas deve exibir as coordenadas horizontal e vertical (X e Y, respectivamente) do *pixel* em questão quando o cursor estiver sobre este na imagem esquerda ou direita. Para isso, basta que um evento relativo ao movimento do *mouse* (*mouse move*) seja ativado quando o cursor estiver sobre a região das imagens, exibindo as coordenadas X e Y de cada um dos pontos indicados pelo cursor (Tabela 3.7).

Tabela 3.7: Linhas de código para exibição das coordenadas da posição do *mouse*

```
private void rightImage_MouseMove(object sender, MouseEventArgs e)
{
    coordinates.Text = "X , Y : " + e.X + ', ' + e.Y;
}
```

A captação das coordenadas para a realização dos cálculos de calibração e dimensionamento também deve ser disponibilizada através do *mouse*. A partir do evento *mouse click*, capta-se os valores das coordenadas horizontal e vertical do *pixel* selecionado, exibindo-os para o usuário na tabela de calibração ou dimensionamento (Seção 3.3).

Visando a praticidade, após a captação das coordenadas de cada um dos *pixels* escolhidos, deve-se exibir a imagem relativa à outra câmera para que outro ponto de interesse seja escolhido pelo usuário. Por exemplo, caso tenha sido escolhido um ponto na imagem relativa à câmera esquerda (direita), após essa escolha, exibe-se automaticamente a imagem direita (esquerda). Após a seleção de um novo ponto nesta imagem, retorna-se à imagem esquerda (direita) para a escolha de mais um ponto, e assim sucessivamente até o final da seleção manual dos pontos.

3.3 - TABELAS

As tabelas desenvolvidas para o *software* foram construídas a partir de componentes de interface gráfica, como *labels* e *textBox*. O objetivo era desenvolvê-las de forma simples, com operabilidade suficiente para realizar pequenas operações, como excluir linha, limpar tabela, entre outras, além de exibir as coordenadas dos *pixels* selecionados. Além disso, a forma como as tabelas são exibidas é de muita importância, pois deve-se disponibilizar a maior área possível do monitor para a visualização das imagens utilizadas, tanto para a calibração como para o dimensionamento.

3.3.1 - ESTRUTURA

A forma ideal para armazenar as coordenadas horizontal e vertical dos *pixels* selecionados é através de uma lista, pois além dos novos dados inseridos na tabela poderem ser automaticamente colocados no fim da lista, também é possível a remoção de dados anteriormente armazenados localizados entre outros dados, havendo, em seguida, uma nova ordenação. Em resumo, a organização e manipulação das informações armazenadas serão realizadas de forma mais fácil e segura. Na Tabela 3.8, segue a forma de representação de um *ArrayList* na linguagem C#.

Tabela 3.8: Linguagem de código utilizando o *ArrayList*

```
public ArrayList textBoxArray;  
public ArrayList labelArray;
```

Para cada um dos pontos selecionados pelo usuário (Seção 3.2.2), as coordenadas X e Y do ponto são exibidas separadamente na tabela e são inseridas na lista. Este processo deve ser realizado para todos os pontos escolhidos pelo operador do *software*, ou seja, a cada novo ponto escolhido, as coordenadas são exibidas e, em seguida, armazenadas no final da lista.

Como a tabela é constituída por componentes da interface gráfica (*labels*, *textBox*, entre outros), os mesmos devem possuir posições pré-determinadas e, a cada nova linha criada, um *offset* na posição de cada elemento deve ser somado, mantendo-se o alinhamento e a estrutura da tabela.

3.3.2 - OPERAÇÕES

Por ser uma tabela simples, apenas três operações de manuseio são necessárias: inserir nova linha, remover linha e limpar a tabela.

Visando maior praticidade na operação das tabelas, as novas linhas são inseridas automaticamente. Para tanto, basta que após o preenchimento da linha anterior, uma nova linha seja inserida na tabela.

No caso da tabela de calibração, cada linha contém sete colunas: as quatro primeiras são referentes às coordenadas X e Y dos *pixels* escolhidos e as três últimas são referentes às coordenadas X, Y e Z do *grid* de calibração. Como estas três últimas são inseridas pelo teclado ou por importação, quando as quatro primeiras colunas forem preenchidas (dois *pixels* selecionados), uma nova linha é inserida no final da tabela.

Para a tabela de dimensionamento, uma nova linha será inserida quando os dois pontos escolhidos para a medição da distância já tenham sido selecionados (oito colunas) e a distância, em centímetros, apresentada.

A opção para a remoção da linha também deve ser oferecida ao usuário. Para uma tabela mais simples, é possível a remoção apenas da última linha. Para isso, no caso da tabela de calibração, apagam-se os sete últimos dados da lista (referentes à última linha) e para a tabela de dimensionamento, apagam-se os nove últimos dados. Os componentes gráficos responsáveis pela exibição das coordenadas também devem ser excluídos, mostrando para o usuário que a última linha foi removida.

Por estar utilizando uma lista para armazenar as informações referentes às coordenadas dos pontos, é possível remover uma linha qualquer, desenvolvendo uma tabela mais completa. Para isso, deve-se utilizar uma variável responsável por indicar o número da linha que o usuário deseja excluir. A partir desta informação, os dados localizados da posição $(7 \times n)$ até a posição $(7 \times n) + 6$, para $0 \leq n \leq (\text{Número de Linhas} - 1)$ da lista serão removidos. Os componentes gráficos da linha removida também deverão ser excluídos e as linhas localizadas abaixo desta deverão ter sua posição vertical reduzida para não ficar nenhum espaço entre as linhas.

Para limpar a tabela, deve-se excluir todas as informações exibidas para o usuário e contidas na lista (Tabela 3.9). Para os componentes gráficos, uma vez que uma nova linha só é inserida após o preenchimento da anterior, deve-se apagar todas as linhas, deixando visível para o usuário apenas a primeira.

Tabela 3.9: Linhas de código relativas à ação "Limpar Tabela"

```
public void clearTable()
{
    ((TextBox)textBoxArray[i]).Dispose();
    ((Label)labelArray[i]).Dispose();
}
```

3.3.3 - VISUALIZAÇÃO

A principal região de operação e manuseio do *software* é a das imagens. Nelas, o usuário poderá selecionar os pontos de seu interesse para a realização da calibração e/ou dimensionamento (Seção 3.2.2). Dessa forma, a maior área possível da interface gráfica

deve ser voltada para a visualização das imagens esquerda e direita. Por outro lado, as tabelas são utilizadas apenas para a aquisição das coordenadas dos *pixels* de interesse, além da exibição do resultado do dimensionamento. A partir desses conceitos, a visualização das tabelas não precisa ser permanente. Deve ser disponibilizada a opção de visualização das mesmas apenas por um curto espaço de tempo ou, por escolha do usuário, permitir a exibição permanentemente.

Quando forem selecionadas, as tabelas devem ficar, sempre que possível, escondidas, mas em pleno funcionamento. Como o *mouse* é o instrumento utilizado para adquirir as coordenadas dos pontos de interesse, eventos relativos ao posicionamento do cursor do mesmo e sua movimentação, podem ser utilizados para controlar a visualização das tabelas. Um exemplo é permitir a visualização apenas enquanto o cursor estiver sobre um determinado componente da interface gráfica (como um botão, por exemplo). Este tipo de visualização seria apenas temporária, pois quando o cursor não estivesse sobre a região, a tabela seria novamente escondida.

A possibilidade de visualização permanente pode ser concebida a partir de um determinado controle, como o *click* do *mouse* sobre um determinado botão ou a marcação de um *checkBox*, deixando a tabela sempre visível. Para voltar a escondê-la, bastava que a ação feita sobre o controle de visualização fosse novamente realizada.

3.4 - EXPORTANDO AS TABELAS

Após o correto preenchimento da tabela de calibração ou dimensionamento, o usuário tem a possibilidade de armazenar os dados contidos nas tabelas, bem como o resultado de possíveis operações envolvendo as mesmas. Além disso, é muito importante que o arquivo texto exportado possa ser aberto em versões anteriores ao novo *software* que está sendo desenvolvido.

Através de uma opção no *menu*, é fornecida a possibilidade de exportação da tabela desejada. Sendo essa opção selecionada, será exibido um *Save File Dialog*, permitindo ao usuário escolher o nome e o local onde o arquivo texto deverá ser salvo. Não deverá ser permitido deixar o nome do arquivo texto em branco: neste caso a operação não é realizada. Além disso, caso o nome escolhido para o arquivo seja igual ao de outro arquivo já existente, deverá ser dada ao usuário a opção de sobrescrevê-lo ou não.

No arquivo texto a ser gerado, os dados de cada linha da tabela são separados por um único espaço em branco e cada linha é separada por um "`\r\n`", que significam, respectivamente, retorno do cursor do teclado e nova linha. Dessa forma, dependendo de qual tabela está sendo exportada, o número de linhas e colunas pode variar.

A tabela de calibração pode ter seus dados exportados de três formas diferentes:

- Coordenadas (X Y) da Imagem: apenas as quatro primeiras colunas são exportadas (coordenadas X e Y do *pixel* escolhido na imagem esquerda e na imagem direita);

- Pontos (X Y Z) do Grid: apenas as três últimas colunas são exportadas (coordenadas X, Y e Z referentes ao grid de calibração);
- Tabela Completa: são exportadas todas as sete colunas.

Para a tabela de dimensionamento, pode-se exportar a tabela completa. O arquivo texto será composto por oito colunas com as coordenadas dos *pixels* escolhidos para o dimensionamento, mais uma coluna que conterà o resultado desta operação informando a distância, em centímetros, entre os pontos escolhidos.

Na importação dos arquivos textos, relativos à tabela de dimensionamento (Seção 3.5), apenas as oito primeiras colunas serão utilizadas, pois o cálculo de dimensionamento será refeito. Assim, caso o usuário deseje criar manualmente um arquivo texto contendo as informações relativas à tabela de dimensionamento, poderá criá-lo apenas com as oito primeiras colunas, pois apenas estas serão utilizadas para o cálculo do dimensionamento entre os *pixels* escolhidos.

Após o término da exportação, uma mensagem deverá ser exibida para o usuário informando o sucesso da operação.

3.5 - IMPORTANDO AS TABELAS

Deve ser disponibilizada ao usuário a possibilidade de utilizar tabelas geradas em operações anteriores. Dependendo do tipo de informação que o usuário deseja obter, deve existir também a possibilidade da importação de apenas uma parte da tabela (algumas colunas, por exemplo). Além disso, esta operação deve poder ser realizada com os arquivos texto exportados em versões anteriores ao novo *software* que está sendo desenvolvido.

Para todas as diferentes possibilidades de importação das tabelas, deve ser fornecido ao usuário (novamente através do *menu*) a possibilidade de importação de um arquivo texto, contendo os dados que foram preenchidos na tabela. Seleccionada essa opção, será exibido ao usuário um *Open File Dialog*, que permitirá que seja escolhido o arquivo que será importado. Para ajudar na visualização dos arquivos texto, pode ser utilizado um filtro no *Open File Dialog* para que apenas os arquivos com extensão (.txt) sejam exibidos. Caso contrário, é permitida a exibição de qualquer extensão.

Nos arquivos textos que serão importados, os dados que preenchem a tabela são separados por um único espaço em branco e cada linha é separada por um "`\r\n`", que significam, respectivamente, retorno do cursor do teclado e nova linha. Dessa forma, para qualquer tipo de importação, após a abertura do arquivo texto, o conteúdo do mesmo deve ser lido e armazenado em uma *string*. Em seguida, devem ser localizados os caracteres "`\r\n`"(separador de linha), responsáveis pelo acréscimo de uma nova linha. A cada separador de linha encontrado, toda a informação anterior deve ser armazenada em uma nova variável, por exemplo um *array de string*. Para cada uma das linhas, agora

devidamente separadas, a cada espaço em branco encontrado (separador dos dados), tudo que foi lido anteriormente deve ser guardado em uma nova variável, por exemplo um *array de string*, armazenando assim cada dado contido na tabela separadamente. Por fim, estes são colocados em suas respectivas posições na tabela.

Para a tabela de calibração, a importação pode ser realizada de três formas:

- **Coordenadas (X Y) da Imagem:** apenas as quatro primeiras colunas são importadas (coordenadas X e Y do *pixel* escolhido na imagem esquerda e na imagem direita). Neste caso, o arquivo texto a ser importado pode ser de duas formas: conter apenas as quatro primeiras colunas da tabela ou ser uma tabela completa (sete colunas). A diferenciação entre as duas possibilidades pode ser feita através de uma contagem do número de espaços em branco (separador de dados) existentes em uma linha. Sendo a tabela a ser importada constituída por quatro colunas, os dados são lidos e exibidos nas suas respectivas posições. Já para a tabela completa, após a correta separação, para cada conjunto de sete dados, os quatro primeiros são exibidos nas posições corretas na tabela. Já os três últimos dados, referentes às coordenadas X, Y e Z do *grid* de calibração, são descartados, não sendo exibidos.
- **Pontos (X Y Z) do *Grid*:** apenas as três últimas colunas são exportadas (coordenadas X, Y e Z referentes ao *grid* de calibração). Neste caso, o arquivo texto a ser importado pode ser de duas formas: conter apenas as três últimas colunas da tabela ou ser uma tabela completa (sete colunas). A diferenciação entre as duas possibilidades pode ser feita através de uma contagem do número de espaços em branco (separador de dados) existentes em uma linha. Sendo a tabela a ser importada constituída por três colunas, os dados são lidos e exibidos nas suas respectivas posições. Já para a tabela completa, após a correta separação, para cada conjunto de sete dados, os quatro primeiros, referentes às coordenadas X e Y da imagem, são descartados, não sendo exibidos. Já os três últimos dados, são exibidos nas posições corretas na tabela.
- **Tabela Completa:** são importadas todas as sete colunas.

Para o dimensionamento, a única opção de importação de tabela é a completa. Apesar dos arquivos textos exportados da tabela de dimensionamento possuírem nove colunas (oito colunas com as coordenadas dos *pixels* escolhidos para o dimensionamento mais uma coluna contendo o resultado do dimensionamento), apenas as oito primeiras são importadas. Após a separação de cada dado e sua correta exibição no local devido, a operação de dimensionamento é refeita para cada linha, sendo o resultado do cálculo de dimensionamento exibido na nona coluna da tabela.

3.6 - MATRIZES

Para os cálculos da calibração e dimensionamento são utilizadas várias operações com matrizes. Assim, uma classe contendo as operações e os manuseios das operações deve ser desenvolvida.

A classe deve permitir a criação de uma matriz de qualquer dimensão. O tipo da variável deve possuir um elevado número de casas decimais, pois as operações realizadas necessitam muita precisão.

Para cada uma das matrizes criadas, algumas características das mesmas devem ser acessíveis, sendo utilizadas diversas vezes, ajudando na realização dos cálculos. Uma delas é possibilitar que seja informado o número de linhas e colunas que constituem uma determinada matriz anteriormente criada (Tabela 3.10).

Tabela 3.10: Linhas de código sobre algumas características das matrizes criadas

```
private int numberOfRows;  
private int numberOfCols;  
private decimal[,] matrix;  
  
public Matrix(int rows, int cols)  
{  
    numberOfRows = rows;  
    numberOfCols = cols;  
  
    matrix = new decimal[rows, cols];  
}
```

Os elementos que compõem as matrizes devem poder ser visualizados isoladamente, se assim for necessário. Através da indicação do índice na matriz ([linha, coluna]), o elemento solicitado pode ser visualizado, alterado ou utilizado para alguma outra operação que seja necessária. O mesmo conceito pode ser aplicado para a manipulação de linhas e colunas.

As operações matriciais podem ser realizadas diretamente a partir da chamada da função desejada. Entretanto, através da sobrecarga dos operadores de adição, multiplicação e divisão, a realização de várias operações matriciais envolvendo diversas matrizes pode ser mais facilmente interpretada e implementada (Tabela 3.11). Dessa forma, não há a necessidade da chamada de várias funções em sequência que tornariam o código possivelmente confuso.

Tabela 3.11: Linhas de código sobre a sobrecarga do operador de adição

```
public static Matrix operator +(Matrix matrix1, Matrix matrix2)  
{ return Matrix.add(matrix1, matrix2); }
```

As operações matriciais utilizadas pelo *software* são:

- Adição: além da soma, esta operação também pode ser utilizada para a realização da subtração a partir da multiplicação do subtraendo por -1;
- Divisão
- Multiplicação: além da multiplicação entre matrizes, deve ser possível a realização da multiplicação de uma matriz por um escalar;
- Matriz Adjunta
- Matriz Identidade
- Matriz Inversa
- Matriz Transposta
- Submatrizes
- Determinante
- Autovalores e Autovetores
- Rotação em Três Dimensões

3.7 - CALIBRAÇÃO E DIMENSIONAMENTO

Voltado para a realização dos cálculos da calibração e do dimensionamento, este torna-se o setor mais importante de todo o código. A partir das coordenadas selecionadas pelo usuário, vários algoritmos serão executados, sendo exibidos, posteriormente, os resultados finais de cada processo.

Esta parte do *software* foi desenvolvida a partir da transição da linguagem original do programa (C++) para a linguagem da nova versão do *software*, o C# (*C Sharp*). Nenhuma alteração relativa aos algoritmos foi realizada, mantendo-se assim todas as etapas das operações. A única diferença nesse setor, entre o *software* original e a nova versão, foi a utilização da nova classe de matrizes desenvolvida neste projeto (Seção 3.6) e a utilização de um tipo de variável com mais casas decimais para a representação das matrizes. Além disso, pelo uso de sobrecarga de operadores para as operações matriciais, as linhas de código com expressões matemáticas ficaram mais claras e de mais fácil interpretação.

Em alguns trechos do código original havia a repetição acentuada de algumas linhas de código. Estes trechos foram transformados em funções, mantendo-se o conteúdo original, e as funções passaram a ser executadas sempre que necessário. Como consequência, houve uma redução no número de linhas de código e uma melhoria na interpretação do mesmo.

CAPÍTULO 4

MANUAL DO USUÁRIO

4.1 - INTRODUÇÃO

Buscando mostrar da forma mais clara possível os resultados obtidos neste projeto, segue, nas seções a seguir, o Manual do Usuário. Este contém explicações sobre a operabilidade e as características do *software*, além das diferentes funções disponíveis para o usuário, e ilustrações relativas à interface gráfica.

4.2 - CONFIGURAÇÕES DO SISTEMA

4.2.1 - PLATAFORMAS SUPORTADAS

Por utilizar o Mono [1], o *software* VEM possui suporte para sistemas 32 bits e 64 bits em uma série de arquiteturas e sistemas operacionais.

4.2.2 - SISTEMAS OPERACIONAIS

- Linux
 - OpenSUSE 10.2
 - SUSE Linux Enterprise 9 x86 / x86_64 / ia64 / S390 / S390X / ppc
 - Novell Linux Desktop 9 x86 / x86_64 /
 - SUSE Linux 10.1 x86 / SLED 10 x86 / SLES 10 x86 / x86_64 / ppc
 - SUSE Linux 10.2 x86 / x86_64 / pcc
 - SUSE Linux 10.3 x86 / x86_64
 - Red Hat Linux Enterprise 4.0 x86
 - Também suportado em outras distribuições Linux x86

- MacOS X
 - MacOS X 10.4 (Tiger)
 - MacOS X 10.5 (Leopard)
- Sun Solaris 8 (Sparc)
- Microsoft Windows
 - Windows XP
 - Windows Vista

4.2.3 - COMPUTADOR: CONFIGURAÇÃO MÍNIMA

- Processador: 1,65 GHz;
- 512 MB RAM;
- Um dos sistemas operacionais citados na Seção 4.2.2;
- Possuir o Mono [1] instalado;
- Aproximadamente 300 Mb para a instalação do Mono [1] e o arquivo executável do VEM. Além disso, é recomendado que tenha espaço suficiente para salvar as imagens geradas pelo software e os dados de calibração e dimensionamento;
- Monitor com resolução mínima de 1024x768.

4.3 - O SOFTWARE

O VEM é composto basicamente por uma janela principal, a qual permite ao usuário realizar todas as operações existentes no *software*. Dentre estas, podemos citar:

- Abertura de *bitmaps* de diferentes resoluções;
- Calibração do sistema;
- Importação de tabelas de calibração;
- Exportação de tabelas de calibração;
- Cálculo da distância espacial entre dois pontos escolhidos;
- Importação de tabelas de dimensionamento;
- Exportação de tabelas de dimensionamento.

4.3.1 - JANELA PRINCIPAL

Ao abrir o arquivo VEM.exe, inicia-se o *software*, sendo exibida para o usuário a janela principal (Figura 4.1). Esta contém um *menu* com as seguintes opções: *Arquivo*, *Calibrar* e *Dimensionar*. Inicialmente, apenas a opção *Arquivo* está disponível. A opção *Calibrar* poderá ser acionada após a abertura de uma imagem (Seção 4.3.1.1). Já *Dimensionar* estará acessível após a realização da calibração do sistema (Seção 4.3.2).

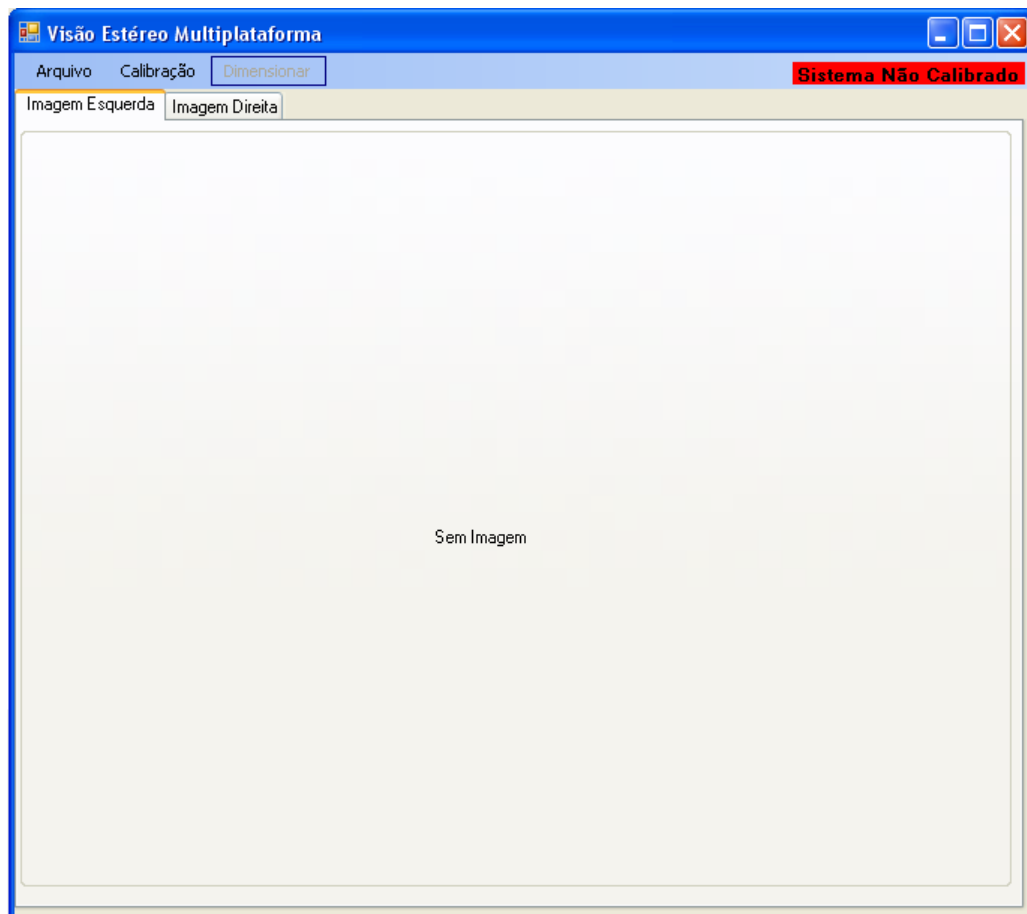


Figura 4.1: Janela Principal

Além do *menu*, existe um *tabpage* com duas abas assim nomeadas: *Imagem Esquerda* e *Imagem Direita*, onde serão exibidas, respectivamente, as imagens relativas à câmera esquerda e à câmera direita. O *label* "Sem Imagem" informa que não há nenhuma imagem sendo exibida inicialmente.

No canto superior da janela principal, é exibido um *label* informativo sobre o sistema, o qual pode estar em duas situações: "Sistema Não Calibrado" e "Sistema Calibrado". Inicialmente, o sistema não está calibrado e, somente após a calibração do mesmo (Seção 4.3.2), o estado do *label* é alterado. Além disso, para dar maior destaque à situação do sistema, o *label* possui uma cor de fundo para cada situação: vermelho para sistema *não* calibrado e verde para sistema calibrado.

4.3.1.1 - MENU *Arquivo*

Dentro deste submenu, existem duas opções: *Imagem* e *Sair* (Figura 4.2) .



Figura 4.2: Menu *Arquivo* e suas opções

- *Imagem* → *Abrir Imagem*

Ao ser selecionada, o usuário poderá escolher qual a imagem será aberta. O formato da imagem suportado pelo *software* é o *bitmap* (.bmp). Após a escolha, o *label* "Sem Imagem" desaparece, e o *bitmap* selecionado será exibido no *tabpage*, sendo dividido automaticamente nas imagens relativas à câmera esquerda e à câmera direita. Acima de cada uma das imagens abertas é exibido o *File Path*, informando a localização do arquivo no sistema operacional.

- *Imagem* → *Fechar Imagem*

Esta opção só está disponível quando alguma imagem foi aberta anteriormente pelo usuário. Sendo este o caso, ao ser selecionada, a imagem exibida será removida e o *label* "Sem Imagem" será exibido nas abas esquerda e direita.

- *Sair*

O programa é encerrado.

4.3.1.2 - MENU *Calibração*

Dentro deste submenu existe uma única opção: *Manual* (Figura 4.3) .

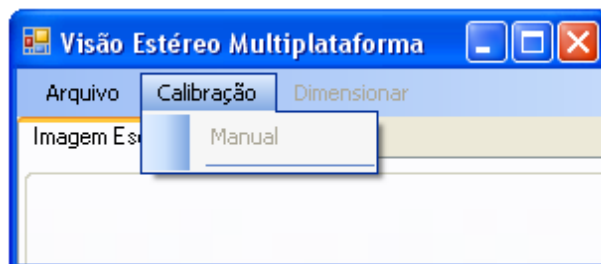


Figura 4.3: Menu *Calibração*

- *Calibração* → *Manual*

Inicialmente esta opção não está disponível. O usuário só poderá acessá-la após abrir uma imagem (Seção 4.3.1.1).

Ao ser selecionada esta opção, será exibido um botão, localizado no lado direito da janela principal, inicialmente intitulado "Mostrar Tabela".

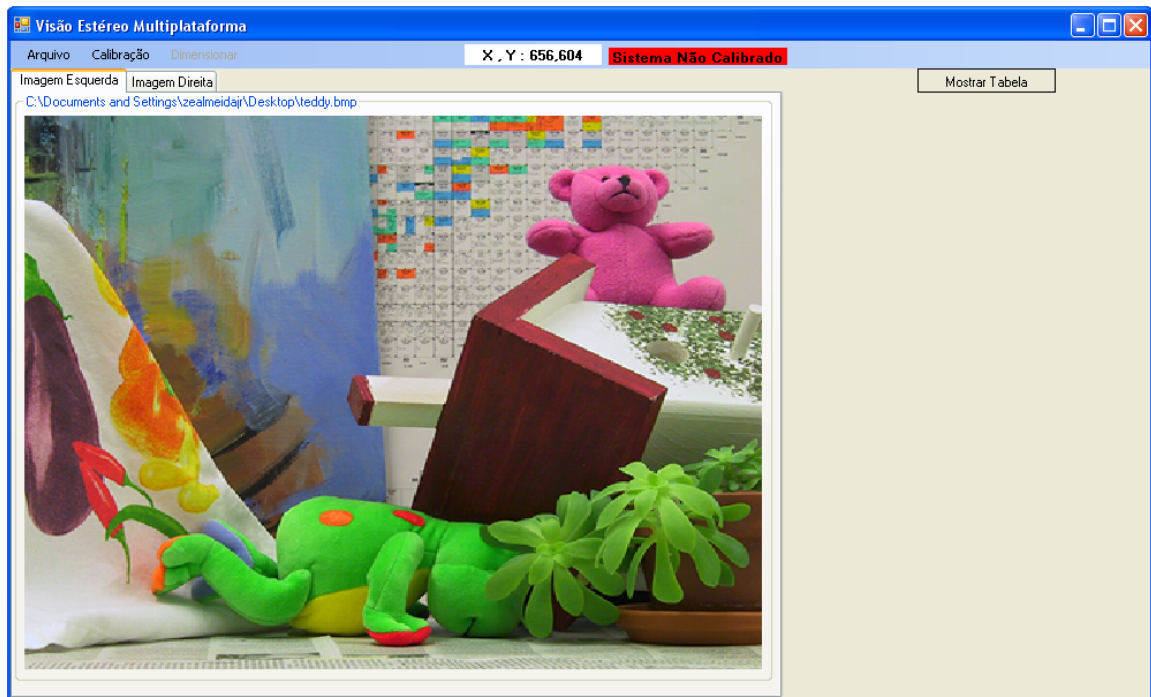


Figura 4.4: Botão "Mostrar Tabela"

Este botão é responsável pelo controle de exibição da tabela de calibração (Figura 4.5). Inicialmente, a tabela não é exibida para o usuário. Existem duas formas de visualização da mesma:

- Exibição temporária: a tabela é exibida enquanto o cursor do *mouse* estiver sobre o botão. Quando esta situação deixa de ser satisfeita, a tabela é escondida.
- Exibição permanente: quando o botão é clicado, mostra-se a tabela permanentemente. Quando o botão é clicado novamente, a tabela volta a ser escondida.

Também passa a ser exibido um *label*, localizado ao lado do *menu*, informando as coordenadas X e Y do cursor do *mouse* quando este está sobre uma das imagens. Para realizar a calibração do sistema, seguir para a Seção 4.3.2.

4.3.1.3 - MENU *Dimensionar*

Inicialmente esta opção não está disponível. O usuário só poderá acessá-la após abrir uma imagem (Seção 4.3.1.1) e calibrar o sistema (Seção 4.3.2).

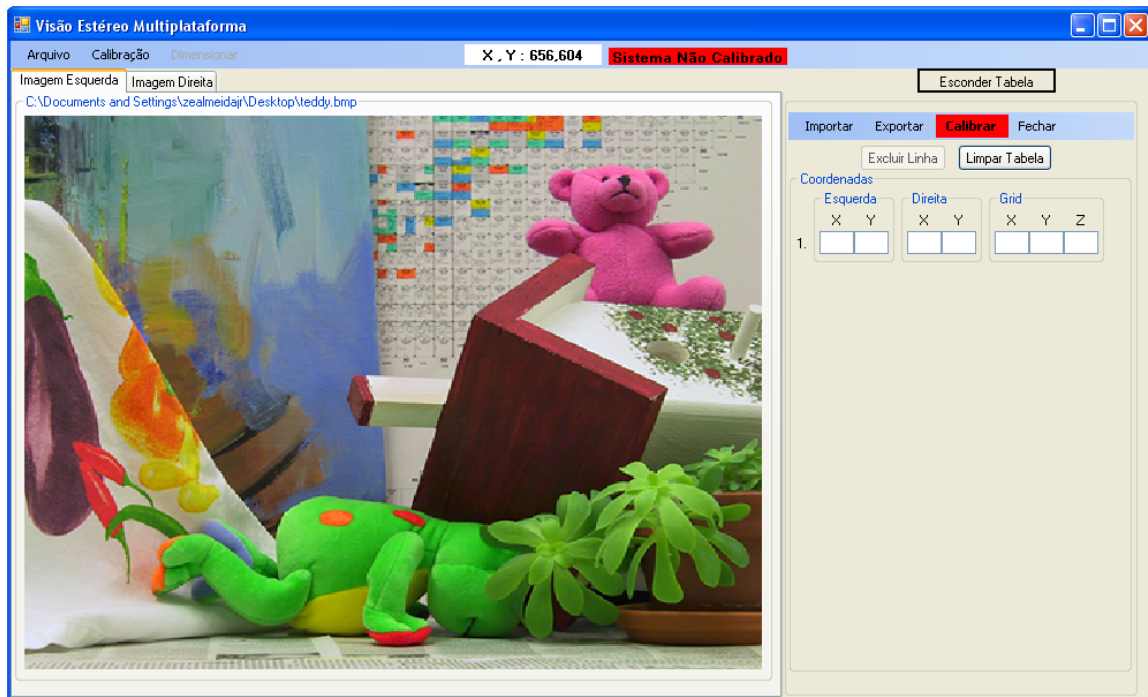


Figura 4.5: Exibição da tabela

- *Dimensionar*

Ao ser selecionada esta opção, será exibido um botão, localizado no lado direito da janela principal, inicialmente intitulado "Mostrar Tabela" , o qual é responsável pelo controle de exibição da tabela de dimensionamento. Inicialmente, a tabela não é exibida para o usuário. Existem duas formas de visualização da mesma:

- Exibição temporária: a tabela é exibida enquanto o cursor do *mouse* está sobre o botão. Quando esta situação deixa de ser satisfeita, a tabela é escondida.
- Exibição permanente: quando o botão é clicado, mostra-se a tabela permanentemente. Quando o botão é clicado novamente, a tabela volta a ser escondida.

A Figura 4.6 mostra um exemplo da janela referente ao dimensionamento do sistema.

Para realizar medições utilizando o sistema, seguir para a Seção 4.3.3.

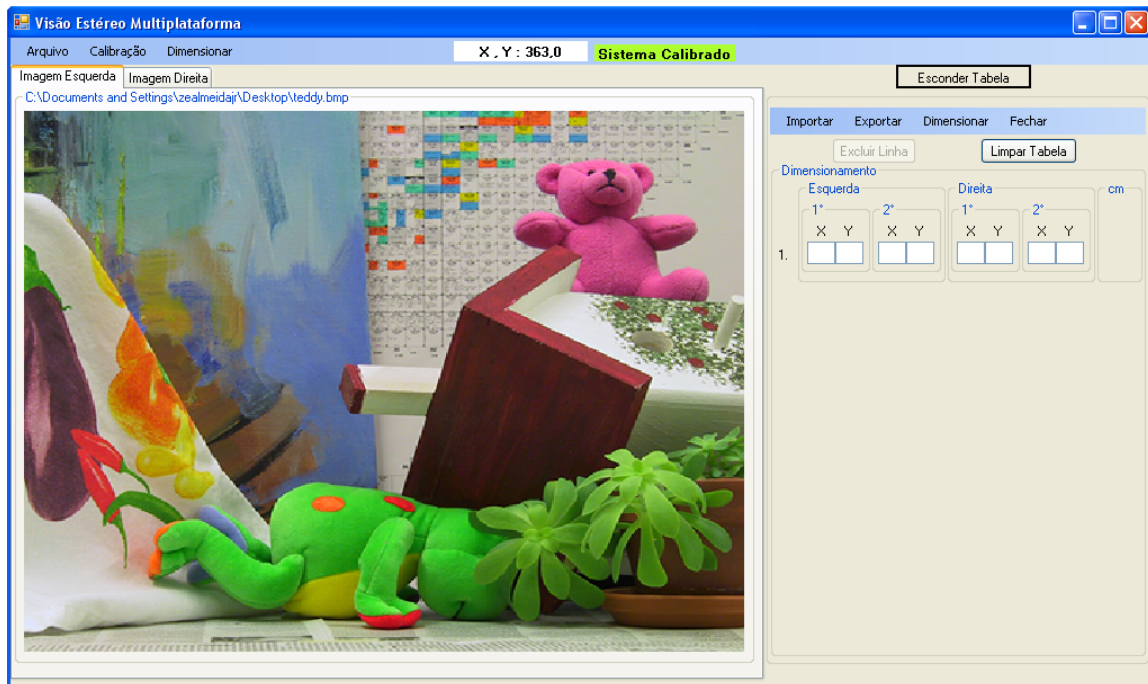


Figura 4.6: Exibição da tabela de dimensionamento

4.3.2 - CALIBRANDO O SISTEMA

Após abrir a imagem desejada (Seção 4.3.1.1) e, em seguida, selecionar *Calibração* → *Manual* (Seção 4.3.1.2), o sistema já pode ser calibrado.

Inicialmente a tabela de calibração está escondida e, como foi explicado na Seção 4.3.1.2, pode ser visualizada temporariamente ou permanentemente. Como exibido na Figura 4.5, a tabela possui sete colunas: as duas primeiras são relativas às coordenadas X e Y do *pixel* escolhido na imagem esquerda. A terceira e quarta colunas são relativas às coordenadas X e Y do *pixel* escolhido na imagem direita. As três últimas colunas referem-se às coordenadas X, Y e Z do *grid* de calibração (Seção 3.3.2).

Para capturar as coordenadas X e Y dos pixels da imagem esquerda e direita, utiliza-se o botão esquerdo do *mouse*. Ao movimentar o cursor sobre a imagem esquerda e/ou direita, são exibidas as coordenadas X e Y de cada *pixel* no *label* localizado ao lado do *menu* da janela principal. Para a captura de um ponto de interesse, basta clicar o botão esquerdo sobre o mesmo e, automaticamente, os valores das coordenadas são transferidos para suas respectivas posições na tabela. Em seguida, a imagem exibida pelo *software* passa a ser a da segunda câmera (esquerda ou direita). Nesta imagem, escolhe-se novamente o ponto de interesse e, após a captura, mais uma vez as coordenadas são transferidas para suas respectivas posições na tabela. Terminada a captura das coordenadas do segundo *pixel*, uma nova linha é inserida automaticamente na tabela e, mais uma vez, a imagem exibida pelo *software* passa a ser a da primeira câmera (esquerda ou direita). As coordenadas dos *pixels* podem ser também digitadas diretamente pelo usuário ou então

importadas de um arquivo texto (Seção 4.3.2.1).

Em relação as coordenadas X, Y e Z do *grid* de calibração, os valores podem ser preenchidos manualmente ou através da importação de um arquivo texto contendo as informações desejadas. Para as coordenadas do *grid*, não há a captação das mesmas utilizando o *mouse*.

Existem dois botões relativos ao manuseio da tabela de calibração. O primeiro deles é o "Excluir Linha", que remove da tabela a última linha. Já o "Limpar Tabela", retira todas as linhas preenchidas, deixando apenas a primeira linha em branco.

Um *menu* também está presente na tabela de calibração, permitindo ao usuário realizar as seguintes funções:

- Importar tabela de calibração
- Exportar tabela de calibração
- Calibrar o sistema
- Fechar a tabela

4.3.2.1 - IMPORTAR TABELA

Permite importar tabelas de calibração anteriormente salvas pelo usuário. Os dados a serem utilizados no preenchimento podem ser obtidos de três formas diferentes: importar apenas as coordenadas X, Y e Z do *grid*, importar apenas as coordenadas X e Y relativas às imagens esquerda e direita, ou importar uma tabela de calibração completa (Figura fig:ImportarTabela).

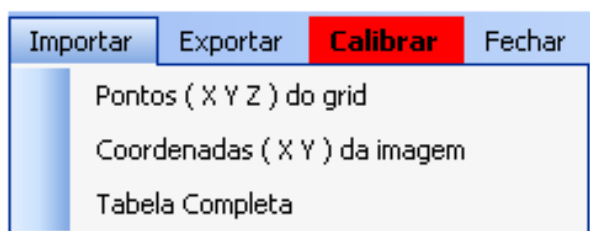


Figura 4.7: Opções para a importação da tabela de calibração

- *Importar* → *Pontos (X Y Z) do Grid*

Apenas as coordenadas do *grid* de calibração (três últimas colunas) são importadas. O arquivo a ser importado pode conter apenas essas três últimas colunas ou, caso seja necessário, as coordenadas podem ser importadas de uma tabela completa, onde apenas os dados relativos ao *grid* de calibração são importados. Após a escolha do arquivo (.txt), as coordenadas são preenchidas automaticamente em suas respectivas posições.

- *Importar* → *Coordenadas (X Y) da Imagem*

Apenas as coordenadas X e Y da imagem esquerda e direita são importadas. O arquivo a ser importado pode conter apenas as quatro colunas relativas às coordenadas ou, caso seja necessário, estas podem ser importadas de uma tabela completa, onde apenas os dados relativos a X e Y são importados. Após a escolha do arquivo (.txt), os valores das coordenadas são preenchidos automaticamente em suas respectivas posições.

- *Importar* → *Tabela Completa*

É importada uma tabela com as coordenadas X e Y da câmera esquerda e direita e as coordenadas do *grid* que foram preenchidas e salvas anteriormente.

4.3.2.2 - EXPORTAR TABELA

Permite ao usuário exportar os dados contidos na tabela para um arquivo texto (.txt). Dependendo de quais informações deseja-se exportar, existem três possibilidades: exportar apenas as coordenadas X, Y e Z do *grid*, exportar apenas as coordenadas X e Y relativas às imagens esquerda e direita, ou exportar a tabela de calibração completa.

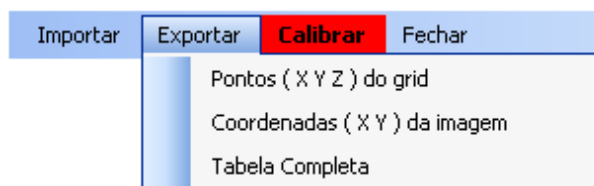


Figura 4.8: Opções para exportação da tabela de calibração

- *Exportar* → *Pontos (X Y Z) do Grid*

Apenas as coordenadas do *grid* de calibração (três últimas colunas) são exportadas.

- *Exportar* → *Coordenadas (X Y) da Imagem*

Apenas as coordenadas X e Y da imagem esquerda e direita são exportadas.

- *Exportar* → *Tabela Completa*

É exportada a tabela com as coordenadas X e Y das câmeras esquerda e direita e as coordenadas do *grid* anteriormente preenchidas.

Para o arquivo atual a ser exportado, caso seja escolhido o mesmo nome de um arquivo já existente, o sistema pergunta ao usuário se deseja sobrescrevê-lo. Em caso de resposta afirmativa, o antigo arquivo terá todo seu conteúdo removido, sendo agora preenchido com os dados existente na tabela atual. No caso de resposta negativa, retorna-se ao *Save File Dialog* para que um novo nome para o arquivo seja escolhido. Após o término da operação, uma confirmação é exibida informando o êxito do processo.

4.3.2.3 - CALIBRAR

Nesta opção, realiza-se a calibração do sistema a partir dos dados inseridos na tabela. São necessários no mínimo seis pares de *pixels* e suas respectivas coordenadas no *grid* para que a calibração possa ser realizada. Caso esse mínimo de pontos não seja satisfeito, uma mensagem é exibida informando o número mínimo de pontos necessários para a calibração. Além disso, caso alguma coordenada não tenha sido informada e a opção para calibração tenha sido selecionada, uma mensagem será exibida informando que a tabela não está devidamente preenchida. Só após a correção do(s) erro(s), poderá ser realizada a calibração.

Com a tabela corretamente preenchida, o sistema realiza a calibração informando, ao término do processo, que o sistema foi calibrado e qual foi o Erro Médio Relativo do processo. Também é exibida uma janela com o erro médio relativo a cada linha da tabela. Após esta confirmação, o *label* informativo sobre o estado do sistema, localizado na janela principal (Seção 4.3.1), tem sua cor de fundo alterada para verde, indicando que o sistema está calibrado.

Sendo necessária a realização de uma nova calibração, estando o sistema já calibrado, ao selecionar a opção *Calibração* → *Manual* o usuário receberá uma mensagem que será exibida informando que o sistema atual já está calibrado e que a calibração será removida, caso o usuário deseje continuar. Caso a resposta seja positiva, a calibração atual será removida, uma nova tabela de calibração será exibida e o *label* informativo sobre o estado do sistema terá sua cor de fundo alterada para vermelho. Sendo a resposta negativa, retorna-se à janela principal.

4.3.2.4 - FECHAR

Caso o sistema tenha sido calibrado, apenas ocorrerá o fechamento da tabela de calibração. Caso contrário, uma mensagem e uma pergunta serão exibidas: a mensagem informa que o sistema *não* está calibrado e que, caso a tabela seja fechada, todos os dados da mesma serão perdidos. A pergunta verifica se o usuário deseja prosseguir. Caso a resposta seja positiva, a tabela será fechada e os dados contidos na mesma serão perdidos. Caso contrário, retorna-se à janela principal e à tabela.

4.3.3 - DIMENSIONAR

Esta opção permite calcular a distância espacial entre dois pontos selecionados. Esta opção só está disponível após a abertura da imagem (Seção 4.3.1.1) e a calibração do sistema (Seção 4.3.2).

Inicialmente a tabela de dimensionamento está escondida e, como foi explicado na Seção 4.3.1.3, pode ser visualizada temporariamente ou permanentemente. Como exibido na Figura 4.6, a tabela é formada por nove colunas: as duas primeiras colunas são relativas

às coordenadas X e Y do primeiro ponto na imagem esquerda; a terceira e quarta colunas são relativas às coordenadas X e Y do primeiro ponto na imagem direita; a quinta e sexta colunas referem-se às coordenadas X e Y do segundo ponto na imagem esquerda; a sétima e oitavas colunas referem-se às coordenadas X e Y do segundo ponto na imagem direita; a última coluna informa a distância espacial entre o primeiro e o segundo pontos escolhidos.

Para capturar as coordenadas X e Y dos *pixels* da imagem esquerda e direita, utiliza-se o botão esquerdo do *mouse*. Ao movimentar o cursor sobre a imagem esquerda e/ou direita, são exibidas as coordenadas X e Y de cada *pixel* no *label* localizado ao lado do *menu* da janela principal. Para a captura de um ponto de interesse, basta clicar o botão esquerdo sobre o mesmo e, automaticamente, os valores das coordenadas são transferidos para suas respectivas posições na tabela. Em seguida, a imagem exibida pelo *software* passa a ser a da segunda câmera (esquerda ou direita). Nesta imagem, escolhe-se o ponto mais parecido com o selecionado anteriormente e, após a captura, mais uma vez as coordenadas são transferidas para suas respectivas posições na tabela e a imagem exibida pelo *software* passa a ser a da primeira câmera (esquerda ou direita). Dessa forma, as coordenadas do primeiro ponto já foram selecionadas. Repete-se o processo para a aquisição das coordenadas do segundo ponto. Ao término desta etapa, será exibida automaticamente na última coluna, a distância espacial (em centímetros) entre o primeiro ponto e segundo ponto. Em seguida, uma nova linha é inserida na tabela para a aquisição das coordenadas de dois novos pontos.

Existem dois botões relativos ao manuseio da tabela de dimensionamento. O primeiro deles é o "Excluir Linha", que remove da tabela a última linha. Já o "Limpar Tabela", retira todas as linhas preenchidas, deixando apenas a primeira linha em branco.

Um *menu* também está presente na tabela de dimensionamento, permitindo ao usuário realizar as seguintes funções:

- Importar tabela de dimensionamento
- Exportar tabela de dimensionamento
- Fechar a tabela

4.3.3.1 - IMPORTAR TABELA

É importada uma tabela com as coordenadas X e Y relativas ao primeiro e segundo pontos de interesse (Figura 4.9). Apesar do arquivo possuir as nove colunas da tabela anteriormente exportada (Seção 4.3.3.2), os dados referentes à última coluna não são importados, sendo recalculados no sistema a medida que cada linha vai sendo inserida.

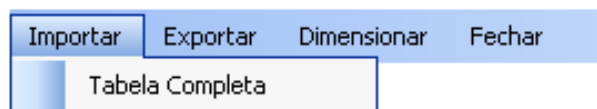


Figura 4.9: Opção para importação da tabela de dimensionamento

4.3.3.2 - EXPORTAR TABELA

A tabela completa com as coordenadas de cada um dos pares de pontos escolhidos para o dimensionamento é exportada. Todas as nove colunas são exportadas para um arquivo texto (.txt).

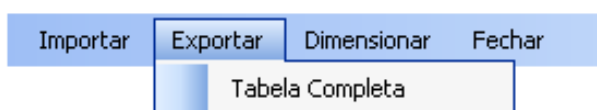


Figura 4.10: Opção para exportação da tabela de dimensionamento

Para o arquivo atual a ser exportado, caso seja escolhido o mesmo nome de um arquivo já existente, o sistema pergunta ao usuário se deseja sobrescrevê-lo. Em caso de resposta afirmativa, o antigo arquivo terá todo seu conteúdo removido, sendo agora preenchido com os dados existente na tabela atual. No caso de resposta negativa, retorna-se ao *Save File Dialog* para que um novo nome para o arquivo seja escolhido. Após o término da operação, uma confirmação é exibida informando o êxito do processo.

4.3.3.3 - FECHAR

Realiza o fechamento da tabela de dimensionamento. Uma mensagem e uma pergunta são exibidos para o usuário. A primeira informando que todos os dados contidos na tabela serão perdidos. A segunda desejando confirmar o fechamento da tabela. Em caso de resposta afirmativa, os dados exibidos na tabela são perdidos e a mesma é fechada. Caso contrário, retorna-se à janela principal e à tabela.

CAPÍTULO 5

CONCLUSÃO

Através deste trabalho, foi feita a adaptação de um *software* de Visão Estéreo, anteriormente desenvolvido no Laboratório de Processamento Analógico e Digital de Sinais (PADS), utilizando a linguagem de programação C# (C Sharp) e o Mono [1], tornando esta nova versão do *software* multiplataforma. Além disso, alterações e melhorias na interface gráfica também foram feitas, tornando o *software* mais prático e mais fácil de ser utilizado. As principais funções existentes nas versões anteriores foram mantidas e devidamente adaptadas, permitindo que este novo *software* possa realizar as tarefas fundamentais de calibração e dimensionamento.

As propostas futuras deste projeto incluem as adaptações de todas as funções presentes nos *softwares* anteriores para esta nova versão, realizando alterações, sempre que necessário, para a melhoria na relação entre o usuário e a interface gráfica. Pode também ser incluída a conexão de câmeras no sistema para captação de imagens em tempo real, seguida das manipulações necessárias ao dimensionamento 3D.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] “<http://www.mono-project.com>”, Último acesso: 06/03/2008.
- [2] “http://www.efoto.eng.uerj.br/_media/doc:cap1.pdf”, Último acesso: 15/02/2008.
- [3] SILVA, L. C., *Método Robusto para a Calibração de Câmeras em Estereofotogrametria*. Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, RJ - Brasil, Agosto 2003.
- [4] JACOBSON, I., *Object Oriented Software Engineering*. Addison-Wesley, 1992.