

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA – CAMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA
BACHARELADO EM ENGENHARIA MECATRÔNICA**

FELIPE SIEDSCHLAG YOPÁN

DESENVOLVIMENTO DE UM SOFTWARE DE MEDIÇÃO POR IMAGEM

Florianópolis, 2020

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA – CAMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA
BACHARELADO EM ENGENHARIA MECATRÔNICA**

FELIPE SIEDSCHLAG YOPÁN

DESENVOLVIMENTO DE UM SOFTWARE DE MEDIÇÃO POR IMAGEM

**Trabalho de conclusão de curso submetido
ao Instituto Federal de Educação, Ciência e
Tecnologia de Santa Catarina como parte
dos requisitos para obtenção do título de en-
genheiro mecatrônico.**

**Orientador: Prof. Dr. André Roberto de
Sousa**

Florianópolis, 2020

Ficha de identificação da obra elaborada pelo autor.

Yopán, Felipe Siedschlag
Desenvolvimento de um software de medição por imagem
/ Felipe Siedschlag Yopán ; orientação de André Roberto de
Sousa. - Florianópolis, SC, 2020.
128 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal
de Santa Catarina, Câmpus Florianópolis. Bacharelado
em Engenharia Mecatrônica. Departamento
Acadêmico de Metal Mecânica.
Inclui Referências.

1. Software de medição por imagem. 2. OpenCv. 3.
Visão computacional. I. Sousa, André Roberto de. II. Instituto
Federal de Santa Catarina. Departamento Acadêmico
de Metal Mecânica. III. Título.

DESENVOLVIMENTO DE UM SOFTWARE DE MEDIÇÃO POR IMAGEM

FELIPE SIEDSCHLAG YOPÁN

Este trabalho foi julgado adequado para obtenção do Título de Engenheiro Mecatrônico e aprovado na sua forma final pela banca examinadora do Curso de Engenharia Mecatrônica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 26 de outubro, 2020

Banca Examinadora:

André Roberto de Sousa, Prof. Dr. Eng.

Maurício Edgar Stivanello.

Raimundo Ricardo Matos da Cunha, Prof. Dr. Eng.



INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS

DECLARAÇÃO DE FINALIZAÇÃO DE TRABALHO DE CURSO

Declaro que o estudante **FELIPE SIEDSCHLAG YOPÁN**, matrícula nº 1510022244, do Curso de Engenharia Mecatrônica, defendeu o trabalho intitulado ***DESENVOLVIMENTO DE UM SOFTWARE DE MEDIÇÃO POR IMAGEM***, o qual está apto a fazer parte do banco de dados da Biblioteca Hercílio Luz do Instituto Federal de Santa Catarina, Campus Florianópolis.

Florianópolis, 12 de Novembro de 2020.

Prof. Orientador do TCC: André Roberto de Sousa

RESUMO

Essa monografia apresenta o desenvolvimento de tecnologias de visão computacional aplicadas à metrologia dimensional e as etapas do projeto de um programa de computador para medição de características dimensionais de peças através de imagens digitais como parte de um trabalho de conclusão de curso. Para cumprir os objetivos traçados, foram pesquisados princípios de aquisição e processamento de imagens digitais e da visão computacional aplicada à metrologia dimensional. Foram estudadas técnicas de referenciamento dimensional para medição utilizando imagens.

Como produto do trabalho foi criado, com a metodologia Scrum, um software de medição baseado no processamento matemático sobre os pixels das imagens digitais, escrito em linguagem Python e utilizando principalmente as bibliotecas OpenCV para processamentos de imagens e PyQt5 para criação da interface gráfica. Ao fim do trabalho, são apresentados os resultados dos testes de avaliação do sistema para verificar sua exatidão de medição.

Palavras-chave: *Software* de medição por imagem. OpenCv. Visão computacional.

ABSTRACT

This paper presents a research about computer vision applied to dimensional metrology and the stages of development of a computer software for measuring of parts using digital images as part of a course conclusion work. In order to fulfill the goals of the project, principles of image acquisition and processing were researched. In addition, some techniques for dimensional reference acquisition necessary to measurement using images were examined. The product of the project is a software for measuring based on mathematical processing over the pixels of digital images, written in Python programming language and using OpenCV, for image processing, and PyQt5, for user interface development, as main libraries. In the end of the document, the results of the tests performed are presented and the measurements are evaluated in terms of percentage error.

Keywords: Measurement software based on image. OpenCV. Mathematical processing over digital images.

LISTA DE FIGURAS

Figura 1: Posicionamento dos pixels	19
Figura 2: Representação das cores dos pixels no padrão RGB	20
Figura 3: Zoom em imagem matricial	21
Figura 4: Zoom em imagem matricial X zoom em imagem vetorial	21
Figura 5: A objetiva	23
Figura 6: Ilustração do diafragma	23
Figura 7: Obturador	24
Figura 8: Padrão de Bayer	25
Figura 9: Sensor CCD de uma câmera	26
Figura 10: O espectro da luz visível	27
Figura 11: Condução dos feixes dentro da objetiva	28
Figura 12: Interior de uma câmera digital	28
Figura 13: Ferramenta de seleção de cores disponibilizada pela Google	30
Figura 14: Dois métodos de conversão para escala de cinza	32
Figura 15: Conversão para escala de cinza seguida de binarização	32
Figura 16: Ilustração de um contorno e o vetor que o representa	33
Figura 17: Representação de um sistema de medição	35
Figura 18: Módulos do sistema de medição	35
Figura 19: Esquema interno de um projeto de perfil	38
Figura 20- Starret VB300DRO	38
Figura 21: Máquina de medir CNC	39
Figura 22: Sistema de medição com câmeras distribuídas	40
Figura 23: Menu do programa QSPAK	41
Figura 24: Interface do Qt Designer com o projeto do software de medição aberto	43
Figura 25: Funcionamento do Scrum	45
Figura 26: Representação de uma peça ao lado de uma régua	51
Figura 27: Cálculo do fator de conversão	52
Figura 28: Conjunto de blocos-padrão	53
Figura 29: Ilustração da obtenção de referência com bloco padrão	54
Figura 30: Teste do algoritmo de identificação do bloco padrão	55
Figura 31: Pixels como pontos em um plano XY	57
Figura 32: Perda de informações na conversão para valores digitais	57
Figura 33: Erro causado pela falta de paralelismo	58
Figura 34: Erro visível causado por diferença de distância entre a câmera e o objeto.	59
Figura 35 - Alinhamento entre a referência dimensional e o mensurando	60
Figura 36: Referência na própria peça	61
Figura 37: Distorção do tipo barril	62
Figura 38: Distorção do tipo almofada	62
Figura 39: Distorção do tipo complexa	63
Figura 40: Soma de Riemann sob a curva $f(x)=2+\text{sen}(x)$ no intervalo 0 a 4 com $\Delta x=0,5$	65
Figura 41: Soma de Riemann sob a curva $f(x)=2+\text{sen}(x)$ no intervalo 0 a 4 com $\Delta x=0,33$	65

Figura 42: Soma de Riemann sob a curva $f(x)=2+\text{sen}(x)$ no intervalo 0 a 4 com $\Delta x=0,15$	66
Figura 43: Círculo representando com dimensões 25x25 e 250x250 pixels	67
Figura 44: Circunferência formada a partir de três pontos	69
Figura 45: Ângulo obtido entre as retas formadas a partir de quatro pontos	71
Figura 46: Interface do programa com a primeira aba selecionada	73
Figura 47: Interface do programa com a segunda aba selecionada	75
Figura 48: Interface do programa com a terceira aba selecionada	76
Figura 49: Interface do programa com a quarta aba selecionada	77
Figura 50: Interface do programa com a quinta aba selecionada	78
Figura 51: Opções da barra de menu	79
Figura 52: Amostra do funcionamento da ferramenta de zoom automático	80
Figura 53: Fluxograma de criação de ponto por seleção de pixels	81
Figura 54: Fluxograma de criação de ponto com entrada manual das coordenadas	81
Figura 55: Caixa de diálogo alertando sobre a atribuição da referência	82
Figura 56: Passos para obtenção do fator de referência através de uma distância conhecida	82
Figura 57: Função 'definirFatorDeConversao'	83
Figura 58: Função 'obterFatorAuto'	84
Figura 59: Função 'realizarMedicao'	85
Figura 60: Função 'identificarFormas'	86
Figura 61: Função 'CalcAreaContagem'	88
Figura 62: Contorno selecionado (círculo) colorido em verde e o restante em vermelho	90
Figura 63: Cálculo da área do contorno	90
Figura 64: Função de cálculo da circunferência com seleção manual dos pontos	91
Figura 65: Distribuição dos pontos geradores da circunferência para minimização de erros	93
Figura 66: Teste do algoritmo de seleção automática dos pontos para construção da circunferência	94
Figura 67: Função de cálculo da circunferência com seleção automática dos pontos	95
Figura 68: Função para o cálculo de ângulos	96
Figura 69: Possíveis ângulos desejados entre os mesmos segmentos de reta	96
Figura 70: Imagem gerada digitalmente para os testes	98
Figura 71: Alvos de medição	99
Figura 72: Imagem gerada para testar detecção e medição de formas	99
Figura 73: Referenciamento com criação manual de pontos	100
Figura 74: Calibração através do bloco-padrão simulado	101
Figura 75: Medição da largura A do polígono 1	102
Figura 76: Medição do círculo com seleção manual de pontos	102
Figura 77: Imagem binarizada para detecção de contornos	103
Figura 78: Medição do círculo com seleção automática de pontos	103
Figura 79: Testes com linhas ligando os pontos selecionados pelo algoritmo	104
Figura 80: Medição da área do polígono 1 pelo método da contagem de pixels	104
Figura 81: Medição da área do polígono 1 com a função do OpenCV (teorema de Green)	105
Figura 82: Teste da medição de ângulo com imagem gerada digitalmente	106
Figura 83: Teste detecção de formas	106

Figura 84: Procedimento de referenciamento utilizando pontos marcados sobre uma régua	111
Figura 85: Peças utilizadas nos testes	112
Figura 86: Desenho da peça 1	112
Figura 87: Desenho da peça 2	113
Figura 88: Desenho da peça 3	113
Figura 89: Desenho da peça 4	113
Figura 90: Medição do diâmetro A na peça 1	114
Figura 91: Medição do diâmetro B da peça 1	114
Figura 92: Medição do diâmetro C da peça 1	115
Figura 93: Medida D na peça 1	115
Figura 94: Medida E da peça 1	116
Figura 95: Erro ao medir um cilindro deitado	117
Figura 96: Medida do espaçamento A na peça 2	118
Figura 97: Medida do diâmetro B na peça 2	119
Figura 98: Medição do diâmetro C na peça 2	119
Figura 99: Medição do ângulo β na peça 2	120
Figura 100: Medições A, B e C na peça 3	121
Figura 101: Medições D, E, F e G na peça 3	121
Figura 102: Medições de A e B na peça 4	123
Figura 103: Medição do ângulo β na peça 4	123

LISTA DE TABELAS

Tabela 1: Valores do teste do cálculo do fator de conversão.....	52
Tabela 2: Componentes da aba 1	73
Tabela 3: Componentes da aba 2	75
Tabela 4: Componentes da aba 3	76
Tabela 5: Componentes da aba 4	77
Tabela 6: Componentes da aba 5	79
Tabela 7: Itens da barra de menu	79
Tabela 8: Coordenadas dos pontos da Figura 65	93
Tabela 9: Valores nominais das medidas a serem testadas	99
Tabela 10: Áreas e perímetros das formas geométricas.....	100
Tabela 11: Resultados das medições com imagem gerada digitalmente e referenciamento manual.....	107
Tabela 12: Resultados das medições na detecção de formas	107
Tabela 13: Medições com imagem gerada digitalmente e referenciamento com bloco-padrão	109
Tabela 14: Resultados das medições da peça 1	117
Tabela 15: Resultados das medições na peça 2	120
Tabela 16: Resultados das medições na peça 3	122
Tabela 17: Resultados das medições na peça 4	124
Tabela 18: Tempos de medição para cenários propostos	125

SUMÁRIO

1.	INTRODUÇÃO	16
2.	OBJETIVOS	18
2.1	Objetivo Geral	18
2.2	Objetivos específicos.....	18
3.	FUNDAMENTAÇÃO TEÓRICA	19
3.1	Representação digital de Imagens	19
3.1.1	Imagens Matriciais	19
3.1.2	Imagens Vetoriais	21
3.2	Aquisição de imagens digitais.....	22
3.2.1	Componentes de câmeras digitais.....	22
a)	Objetiva	22
b)	Diafragma.....	23
c)	Obturador.....	23
d)	Filtro com padrão de Bayer	24
e)	Fotossensor.....	26
3.2.2	Funcionamento da câmera digital.....	26
3.3	Processamento de Imagens.....	29
3.3.1	Modificação dos valores de cor dos pixels	29
3.3.2	Conversão do sistema de cores	29
3.3.3	Binarização	32
3.3.4	Identificação de contornos.....	33
3.4	Fundamentos de Metrologia aplicados no contexto do trabalho.....	34

3.5	Visão Computacional aplicada na metrologia dimensional	37
3.5.1	Projektor de Perfil.....	38
3.5.2	Máquinas de Medir CNC.....	39
3.5.3	Sistemas de medição óptica distribuídos	39
3.5.4	Softwares de processamento de imagens para metrologia dimensional.....	40
3.6	Ferramentas de desenvolvimento de software utilizadas	41
3.6.1	Python.....	42
3.6.2	OpenCV	42
3.6.3	PyQt5 e Qt Designer.....	42
3.7	Metodologia.....	44
3.7.1	Metodologia Scrum para projetos de software	44
3.7.2	Kanban.....	46
4.	CONCEITO DO PROJETO:.....	47
4.1	Idealização e planejamento das funcionalidades.....	47
4.1.1	Funcionalidades globais	47
4.1.2	Funcionalidades para a obtenção da referência dimensional.....	47
4.1.3	Funcionalidades para a medição da distância entre pontos	48
4.1.4	Funcionalidades para a identificação de formas geométricas	48
4.1.5	Funcionalidades para a medição de áreas.....	48
4.1.6	Funcionalidades para a medição de circunferências e arcos	48
4.1.7	Funcionalidades para a medição de ângulos.....	49
4.2	Requisitos metrológicos e operacionais	49
5.	DESENVOLVIMENTO DO PROJETO.....	50

5.1	Obtenção da referência dimensional	50
5.1.1	Referenciamento com base em instrumentos ou peças medidas	50
5.1.2	Referenciamento utilizando bloco padrão	52
5.2	Fontes de erro investigadas	56
5.2.1	Resolução insuficiente	56
5.2.2	Paralelismo entre os planos da câmera e do objeto medido	58
5.2.3	Peça e a referência em distâncias diferentes da câmera	59
5.2.4	Distorções	61
5.3	Processamentos matemáticos implementados	63
5.3.1	Distância entre pontos:	64
5.3.2	Cálculo de áreas pela contagem de pixels	64
5.3.3	Construção de retas a partir de dois pontos	67
5.3.4	Determinação de ponto médio de um segmento de reta	68
5.3.5	Determinação de distância entre um ponto e uma reta	68
5.3.6	Construção geométrica de circunferências utilizando três pontos	68
5.3.7	Determinação do ângulo entre duas retas formadas por quatro pontos	70
5.3.8	Conversão de comprimento em pixel para unidade de comprimento real	71
5.3.9	Conversão de área em pixel para unidade de área real	71
5.4	Implementação das funcionalidades	72
5.4.1	Visão geral da interface gráfica	72
5.4.2	Criação de pontos	80
a)	Selecionando na imagem	80
b)	Criação com coordenadas informadas pelo usuário	81

5.4.3	Obtenção da Referência Dimensional	82
a)	Utilizando uma distância conhecida.....	82
b)	Usando um bloco padrão	83
5.4.4	Medição da distância entre Pontos	84
5.4.5	Identificação de formas geométricas simples e medição de suas características	85
5.4.6	Medição de áreas	87
5.4.7	Medição de raios e do centro de circunferências.....	90
5.4.8	Medição de ângulos	95
6.	VALIDAÇÃO EXPERIMENTAL	97
6.1	Testes com imagens geradas digitalmente.....	97
6.1.1	Procedimentos para a realização das medições	100
6.1.2	Medições com fator de conversão obtido a partir da distância entre pontos....	107
6.1.3	Medições com fator de conversão obtido através do bloco-padrão	109
6.2	Testes com fotografias de peças reais.....	110
6.2.1	Procedimento padrão utilizado para o referenciamento dimensional.....	110
6.2.2	Peças utilizadas nos testes	111
6.2.3	Medições na peça 1.....	114
6.2.4	Medições na peça 2.....	118
6.2.5	Medições na peça 3.....	121
6.2.6	Medições na peça 4.....	123
6.2.7	Teste do tempo de medição	124
7.	CONCLUSÕES	126
	REFERÊNCIAS	127

1. INTRODUÇÃO

O rápido avanço da tecnologia na indústria e as necessidades do mercado trazem consigo demandas de maior velocidade de produção ao mesmo tempo em que crescem as exigências quanto à qualidade, tornando cada vez mais importante o papel da inspeção automatizada. Para suprir essas demandas faz-se necessária a criação de novos métodos de inspeção que garantam tanto a agilidade quanto a confiabilidade dos processos. Nesse cenário, a visão computacional surge como alternativa, uma vez que consegue aliar o desempenho crescente do processamento computacional com a alta capacidade de adaptação que traz consigo, principalmente com os avanços na área da inteligência artificial.

O presente trabalho tem por objetivo estudar ferramentas de visão computacional de maneira voltada à metrologia dimensional, através de pesquisas dos temas envolvidos nessa área e do desenvolvimento de um programa capaz de realizar medições através de imagens digitais baseado nessas ferramentas.

Na fundamentação teórica são tratados assuntos relacionados a aquisição de imagens digitais a partir do estudo de conceitos básicos da fotografia e funcionamento de câmeras digitais, seguidos de tópicos sobre processamento de imagens. Ainda na parte de fundamentação teórica é tratado sobre fundamentos de sistemas de medição, alguns exemplos de tecnologias existentes de visão computacional aplicada à metrologia já presentes no mercado e das ferramentas de desenvolvimento de *software* disponíveis e escolhidas para o projeto. Finalizando a parte de fundamentação, é dada uma visão geral sobre a metodologia Scrum para projetos de software, usada no projeto em conjunto com o Kanban, que é um método de gestão de tarefas muito popular.

Em seguida, é apresentado o conceito do projeto, com levantamento das funcionalidades e requisitos esperados. No tópico de desenvolvimento são apresentados os métodos de referenciamento dimensional, as fontes de erro inerentes ao processo de medição aplicado, os fundamentos matemáticos utilizados como base para a implementação das funcionalidades e, principalmente, os algoritmos criados para a implementação dessas funcionalidades no programa, onde também são exibidas as telas da interface gráfica criada e os procedimentos de uso do mesmo.

Por fim, são mostradas capturas de tela e dados relacionados à validação experimental, com testes feitos primeiramente utilizando imagens criadas digitalmente com medidas nominais bem controladas e também utilizando imagens de peças reais, cujas medições são comparadas aos valores obtidos utilizando instrumentos convencionais de medição. Os resultados dos testes são avaliados em

termos de erro percentual.

2. OBJETIVOS

2.1 Objetivo Geral

Este trabalho de conclusão de curso tem por objetivo desenvolver um programa que permita a medição de características dimensionais de objetos através de imagens digitais processadas computacionalmente.

2.2 Objetivos específicos

A fim de cumprir o objetivo do projeto, planejou-se:

- Pesquisar bibliografias sobre metrologia dimensional;
- Pesquisar bibliografias sobre visão computacional aplicada na medição;
- Dominar ferramentas de desenvolvimento de software para visão computacional;
- Pesquisar e identificar necessidades de um software de medição por imagens;
- Desenvolver o software de medição por imagens orientados às necessidades identificadas;
- Avaliar as principais fontes de erros de medição com uso de imagens;
- Realizar testes de verificação da exatidão das medições feitas pelo programa, comparando com medições feitas através de instrumentos convencionais do laboratório de metrologia do IFSC;

3. FUNDAMENTAÇÃO TEÓRICA

3.1 Representação digital de Imagens

Uma imagem digital é uma representação de uma imagem bidimensional na forma binária, podendo ser processada em um sistema computacional ou mantida em alguma mídia de armazenamento. As imagens digitais existem em duas formas: imagens matriciais, também chamadas de imagens de rastreio (*raster* ou *bitmap*); e as imagens vetoriais (*vector image*). No decorrer deste trabalho, o termo ‘imagem’ será usado para se referir à imagem digital.

3.1.1 Imagens Matriciais

Como o nome sugere, uma imagem matricial representa a imagem bidimensional na forma de uma matriz com certo número de linhas e colunas preenchidas por unidades básicas chamadas de *pixels*. Cada *pixel* possui uma posição em uma linha e uma coluna da matriz e uma informação referente à sua cor. A imagem abaixo ilustra uma imagem matricial de 16 (4x4) pixels exemplificando como são designadas as posições de cada pixel em imagens desse tipo.

Figura 1: Posicionamento dos pixels

P(0,0)	P(0,1)	P(0,2)	P(0,3)
P(1,0)	P(1,1)	P(1,2)	P(1,3)
P(2,0)	P(2,1)	P(2,2)	P(2,3)
P(3,0)	P(3,1)	P(3,2)	P(3,3)

Fonte: Elaborada pelo autor.

A cor de um pixel é comumente representada como uma composição de cores RGB (*red*, *green*, *blue*), em que a intensidade de cada componente é dada por um número de 8 bits, variando de 0 a 255 e cuja combinação possibilita a obtenção de 16.777.216 representações de cores distintas; mas também é comum que a imagem esteja em escala de cinza, caso em que a cor de cada pixel é um tom de cinza também variando entre 0 e 255. Imagens em escala de cinza são muito úteis em casos em que se precise poupar processamento e/ou armazenamento e em aplicações onde as cores não tenham relevância. A seguir, na Figura 2, é apresentada uma imagem ilustrando um possível preenchimento

de cores em formato RGB para os pixels da imagem apresentada anteriormente (Figura 1), nela também é possível observar alguns exemplos de como se comporta a cor resultante com a variação dos valores componentes.

Figura 2: Representação das cores dos pixels no padrão RGB

C(255,0,0)	C(0,0,0)	C(85,85,85)	C(255,255,255)
C(255,0,127)	C(127,0,0)	C(255,255,0)	C(127,127,127)
C(127,0,255)	C(0,0,127)	C(0,127,0)	C(0,0,0)
C(0,0,255)	C(0,127,255)	C(0,255,127)	C(0,255,0)

Fonte: Elaborada pelo autor.

Alguns aspectos podem ser observados em relação à composição das cores:

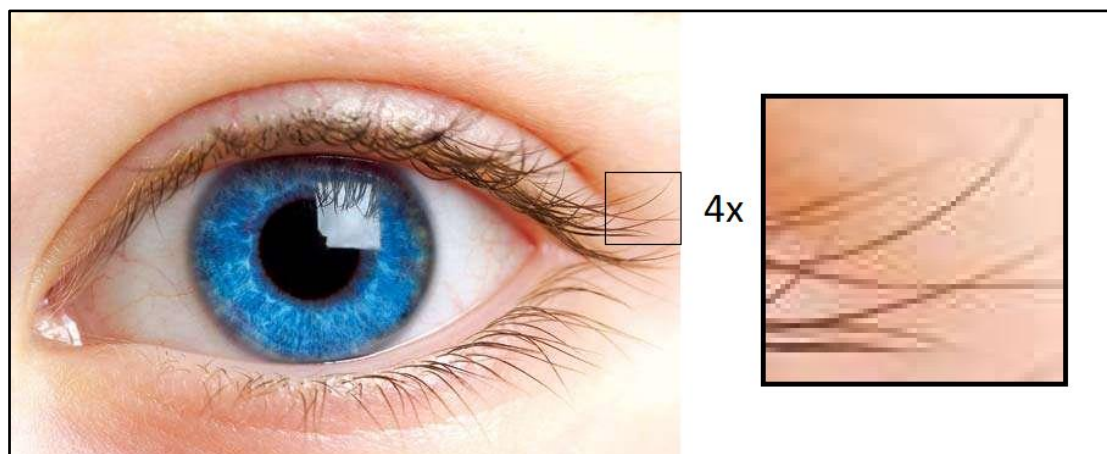
- Com todas as componentes em 0, obtemos a cor preta;
- Com todas as componentes em 255, obtemos a cor branca;
- A cor resultante se comporta tal como ocorre na natureza, como no exemplo em que o valor [255,255,0], vermelho com verde, resulta no amarelo.
- Quando todas as componentes tem o mesmo valor, obtemos um tom de cinza.

Em inglês, um termo comum para denominar imagens matriciais é a palavra *bitmap*, significando literalmente ‘mapa de bits’.

Uma característica negativa das imagens matriciais é a perda de qualidade ocorrida no seu redimensionamento. Quando uma imagem matricial é redimensionada, por exemplo, para o dobro de sua resolução original, não são criadas novas informações para os pixels a mais que foram preenchidos. O que ocorre é que a informação de cada pixel será espalhada para 4 *pixels* (a nova imagem tem o dobro de *pixels* nas duas dimensões), tendo como efeito a degradação de definição. Na Figura 3

pode-se observar esse efeito quando aplicada uma ampliação de 4 vezes em uma região da imagem.

Figura 3: Zoom em imagem matricial



Fonte: Adaptada de: escola.britannica.com.br, 2020.

3.1.2 Imagens Vetoriais

Imagens vetoriais são formadas por expressões matemáticas que representam as geometrias nelas contidas. De forma que os dados armazenados em memória contêm instruções que descrevem como as figuras devem ser geradas, como retas, círculos e arcos, e não dados correspondentes aos pixels e cores. No momento de apresentar essa imagem na tela, o computador usará as descrições matemáticas para gerar uma imagem matricial que a represente na resolução que a aplicação exigir. Como consequência dessa característica vem a grande vantagem das imagens vetoriais em relação às matriciais, o fato da qualidade delas ser mantida independente de redimensionamentos.

A Figura 4 exibe uma comparação do efeito de ampliação em uma imagem vetorial comparado à mesma ampliação em imagem matricial.

Figura 4: Zoom em imagem matricial X zoom em imagem vetorial



Fonte - www.digitalflare.co.uk, 2020.

Apesar dessa vantagem de imagens vetoriais, o uso delas é limitado a aplicações específicas, como design gráfico, marketing e logos de empresas, por exemplo.

Por serem o formato padrão da fotografia, as imagens matriciais são mais adequadas às finalidades de medição e inspeção. Neste trabalho, todas as imagens utilizadas são do tipo matricial.

3.2 Aquisição de imagens digitais

A aquisição de imagens de forma geral é um campo de estudos bastante amplo, portanto, detalhar precisamente todos os processos e etapas envolvidos desde a captura da imagem na lente de uma câmera até seu armazenamento em uma mídia fugiria ao escopo deste trabalho. De tal forma que serão tratadas aqui apenas de maneira resumida as etapas e componentes envolvidos na aquisição de imagens com câmeras digitais.

3.2.1 Componentes de câmeras digitais

O objetivo de uma câmera é converter uma cena real tridimensional em uma representação em duas dimensões daquela cena num dado instante ou intervalo de tempo. Para isso, a câmera usa como princípio de funcionamento a captura da luz incidente sobre ela. Câmeras analógicas registram essa informação de luz gravando-a em filmes e as câmeras digitais utilizam filtros de cor e fotossensores para converter a informação analógica de intensidade da luz em forma digital binária.

Seus principais componentes são:

a) Objetiva

A parte da câmera por onde entram os feixes de luz do ambiente é chamada de objetiva, ela geralmente é composta por um conjunto de lentes que direcionam os feixes de luz que atravessarão o restante dos componentes internos da câmera, até que cheguem aos fotossensores.

Este componente utiliza os efeitos de refração da luz para conduzir os feixes luminosos de maneira a produzir os efeitos desejados. As objetivas podem ser compostas por diversas lentes dispostas em série, uma atrás da outra, as quais podem movimentar-se para causar ampliações, reduções e ajustes de foco. Uma lente que causa um efeito específico e que pode ser citada como exemplo pela sua popularidade, é a do tipo ‘olho de peixe’, a qual possui grande convexidade, causando um efeito de distorção na fotografia resultante à medida que se aproximam as bordas da imagem. Para o tema deste trabalho, as lentes ideais são aquelas em que haja a menor distorção possível em toda a área da imagem e com foco uniforme ao longo do objeto alvo, uma vez que se deseja que ela reproduza o mais fielmente possível a cena fotografada. Esse tipo de objetiva é designado pelo termo ‘objetiva normal’, e tem como característica reproduzir a perspectiva mais próxima do que seria natural ao olho humano.

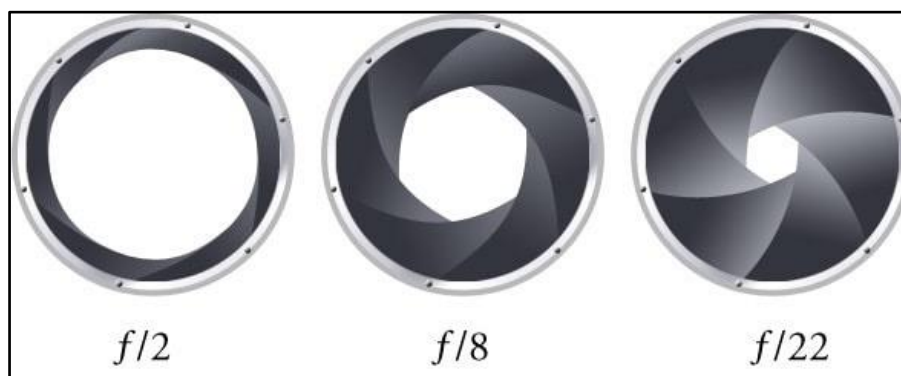
Figura 5: A objetiva

Fonte: focusfoto.com.br, 2020.

b) Diafragma

É um componente relacionado com as lentes, estando geralmente dentro da objetiva, e responsável por controlar a quantidade de luz que irá atravessá-la. Funciona como uma janela com controle de abertura, quanto mais aberto o diafragma, mais luz atravessará as lentes. O efeito da abertura do diafragma na imagem resultante é observado na claridade da imagem. Se o diafragma estiver aberto demais numa fotografia em um ambiente muito iluminado, ocorrerá o chamado estouro de luz, deixando a foto esbranquiçada e com definição degradada. Em caso de uma foto noturna, o diafragma mais aberto auxilia na captura de mais detalhes da cena, pois permitirá a passagem da maior quantidade possível de luz aos fotossensores.

O grau de abertura do diafragma é representado por f/valor , onde f é a distância focal e quanto maior o valor, mais fechado está o diafragma, conforme ilustrado na Figura 6:

Figura 6: Ilustração do diafragma

Fonte: <https://blog.emania.com.br>, 2020.

c) Obturador

Também pode-se dizer que esse componente funciona como uma janela. Mas, diferente do

diafragma, que possui um grau de abertura, o obturador possui apenas os estados aberto e fechado. O obturador também influencia na iluminação da imagem, mas faz isso controlando o tempo, e não à intensidade. Mais precisamente, este componente controla o tempo durante o qual o fotossensor estará exposto à luz.

Na fotografia, o tempo de abertura do obturador é chamado de tempo de exposição e tem grande influência na imagem resultante. Durante todo o tempo em que o obturador está aberto, o fotossensor captura a luz incidente sobre ele. Um maior tempo de exposição será benéfico no sentido de permitir a obtenção de uma imagem mais clara, porém terá efeitos negativos caso haja movimento na cena durante esse intervalo. O que ocorre é que, para cada ponto da cena fotografada deve corresponder um pixel na imagem resultante, porém a luz da cena fotografada entra na câmera de maneira contínua, e se durante o tempo de abertura do obturador os pontos da cena incidirem sobre unidades diferentes do fotossensor, ocorrerá uma sobreposição de informações, tendo borrões como resultado. Por outro lado, um tempo de exposição curto consequentemente exigirá maior iluminação na cena fotografada.

Fisicamente, o obturador é um componente mecânico ou eletromecânico com uma chapa móvel que se movimenta de modo a permitir ou não a passagem de luz. Um exemplo de obturador é apresentado a seguir (Figura 7):

Figura 7: Obturador



Fonte: <https://blog.emania.com.br>, 2020.

O efeito do obturador também pode ser obtido eletronicamente apenas controlando o tempo de leitura do sensor óptico.

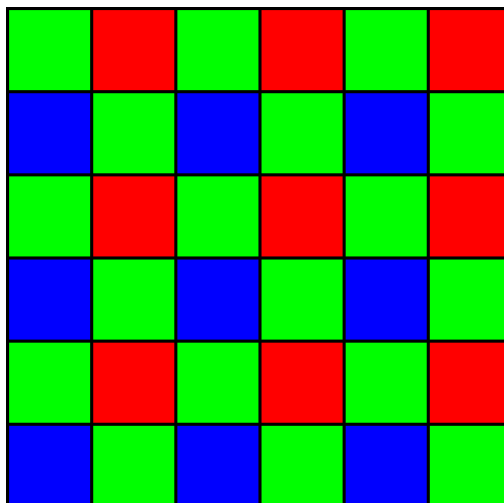
d) Filtro com padrão de Bayer

Como será explicado a seguir, o fotossensor tem a capacidade de acumular carga elétrica proporcionalmente à intensidade de luz incidente sobre ele; porém, isso não é suficiente para identificar

a cor dessa luz. Por esse motivo, fez-se necessário criar métodos de filtragem de luz externos ao sensor, sendo esta a finalidade do filtro de Bayer, o mais usado em câmeras digitais.

O padrão de Bayer, é uma matriz composta por filtros de cor intercalado entre as cores vermelho, verde e azul, como representada na Figura 8:

Figura 8: Padrão de Bayer



Fonte: Elaborada pelo autor.

Cada quadrado da matriz fica posicionado diretamente em frente a um dos pontos do fotosensor, permitindo apenas a passagem da luz correspondente à sua cor. Dessa forma, cada ponto do sensor recebe a informação da intensidade luminosa apenas de uma dessas três cores primárias, fazendo com que cada pixel da imagem pura armazene apenas informação sobre a intensidade de uma dessas cores.

Para obter as cores reais da imagem que será armazenada, para cada pixel é realizado um cálculo de interpolação levando em consideração o valor de cor do pixel atual e dos outros oito a sua volta. O resultado desse cálculo são três valores: um correspondente a intensidade de vermelho, outro a de verde e o último a de azul, valores esses correspondentes a composição da cor resultante. Esse processamento é executado como processo interno da câmera e a imagem final é armazenada já com a composição de cores de cada pixel.

Existem outros métodos de filtragem de luz, como o Foveon X3, um sensor que é capaz de capturar a intensidade de cada uma das três cores do padrão RGB em todos os pontos do sensor, utilizando filtros sobrepostos. Porém, câmeras com sensores desse tipo costumam ser muito mais caras do que as que usam o padrão de Bayer, o que desencoraja seu uso.

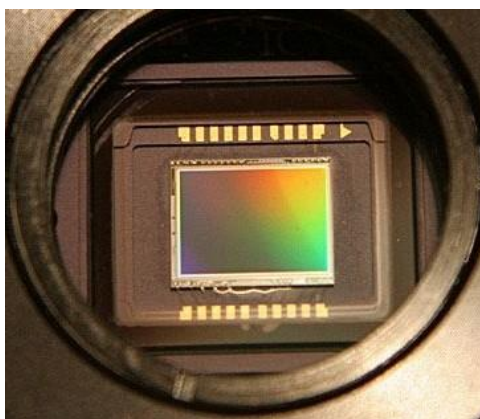
e) Fotossensor

É o componente pelo qual são recebidos os sinais luminosos relativos à cena capturada; na verdade, são compostos por milhões de fotossensores dispostos em forma matricial formando um anteparo na forma de uma chapa retangular. Esse pequeno sensor capta a luz incidente sobre ele na forma de uma carga elétrica proporcional à intensidade dessa luz; cada sensor desses ocupa uma pequena área do anteparo, e a essa área é dado o nome de ‘*photosite*’, ao qual também é relacionado um pixel da imagem resultante. Se considerarmos duas chapas desses sensores de mesmas dimensões, aquela em que os *photosites* tiverem menor área - e conseqüentemente maior densidade de pixels - será a que deve apresentar melhor definição.

Para não causar confusão, apesar do sensor ser um composto com milhões de fotossensores individuais, é mais prático dar o nome de sensor a todo o conjunto do circuito integrado que abrange os fotossensores, transistores e outros componentes envolvidos no processo de transdução do sinal analógico da luz em sinal digital, e que compõem um circuito integrado.

Existem dois tipos principais de sensores utilizados em câmeras. O tipo CCD (*charge-coupled device*) e o CMOS (*complementary metal-oxide-semiconductor*). Cada tipo possui vantagens e desvantagens em relação ao outro e suas diferenças estão principalmente na forma como é feita a varredura dos sinais adquiridos.

Figura 9: Sensor CCD de uma câmera



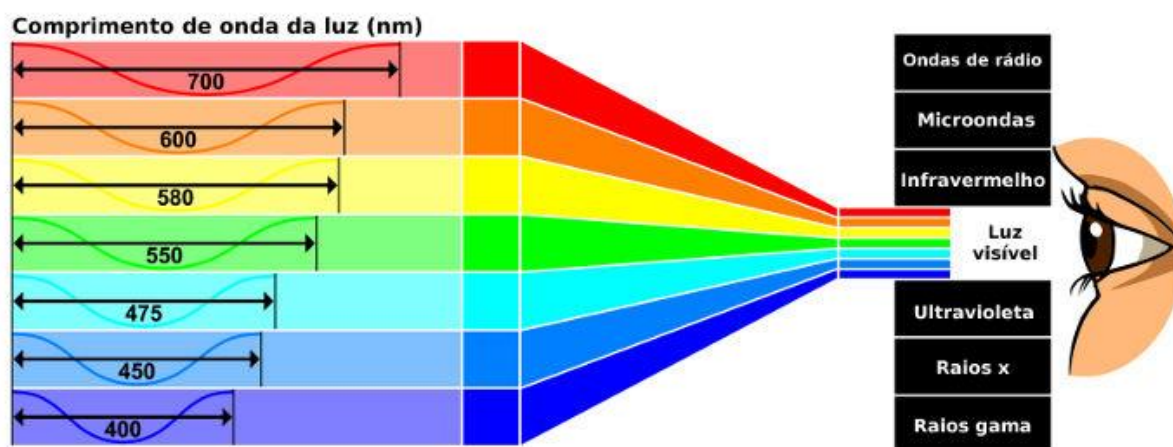
Fonte: www.brasilmergulho.com, 2020.

3.2.2 Funcionamento da câmera digital

Tal como os olhos, as câmeras funcionam registrando a luz incidente sobre ela. A luz é uma onda eletromagnética, cujas principais características são sua frequência, inversamente relacionada ao comprimento de onda, e intensidade. A luz visível aos humanos está presente em uma faixa de

frequência com comprimentos de onda correspondentes variando entre 400 e 700 nm, dentro do espectro das ondas eletromagnéticas.

Figura 10: O espectro da luz visível

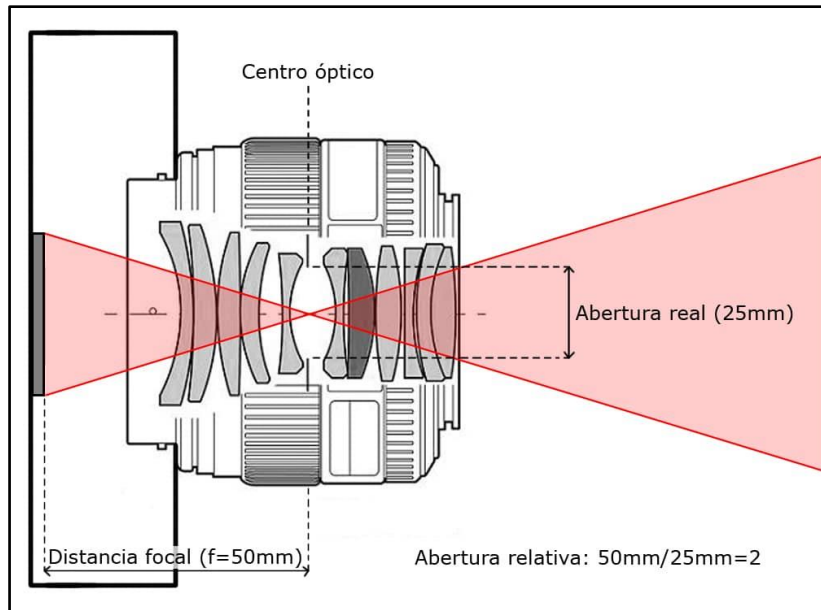


Fonte: <https://brasilecola.uol.com.br>, 2020.

A luz se propaga no espaço com uma certa velocidade dependente do meio pelo qual passa. Ela pode ser representada como um vetor que parte da fonte luminosa em uma dada direção. Ao encontrar objetos, a luz pode atravessá-los, ser absorvida, refletida ou desviada por eles, com base nos efeitos de refração e reflexão das ondas. De maneira simplificada, a cor aparente de um objeto é o resultado das frequências de onda que foram refletidas por ele, que são absorvidas pelos nervos ópticos do olho e processadas pelo cérebro. Então, as imagens são formadas pela composição de todos os feixes de luz que, ao refletirem nos objetos da cena, incidem sobre os olhos.

No caso da câmera, a objetiva recebe os feixes de luz incidentes sobre ela e os transmite através das lentes. As diferentes lentes presentes na objetiva usam da refração da luz para conduzir os feixes da maneira desejada. Ao passar pelo diafragma alguns feixes podem ter sua passagem interrompida ou não, dependendo da abertura selecionada para o mesmo, tendo como efeito um corte nas bordas da cena capturada.

Figura 11: Condução dos feixes dentro da objetiva

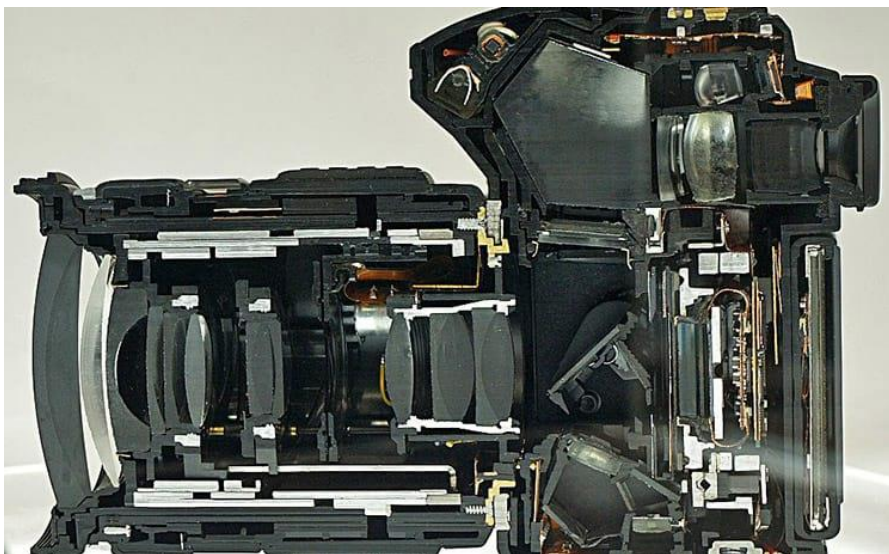


Fonte: <https://blog.emania.com.br>, 2020.

O obturador fica no estado normalmente fechado, no qual interrompe a passagem da luz. Quando a câmera é disparada, o obturador se abre por um intervalo de tempo, permitindo que a luz atinja o filtro com padrão de Bayer. Cada ponto do filtro, permite apenas a passagem da luz com comprimentos de onda correspondentes a uma das três cores usadas como base: o vermelho, verde ou azul.

Quando os raios de luz atingem o módulo fotossensor, já estão separados nas respectivas frequências de interesse, e tem a capacidade de carregar eletricamente os pontos sensitivos com energia proporcional à sua intensidade durante o período de exposição determinado pelo obturador.

Figura 12: Interior de uma câmera digital



Fonte: www.oficinadanet.com.br, 2020.

Por fim, os sensores são descarregados e os sinais analógicos na forma de carga elétrica são discretizados em valores digitais, os quais são utilizados no cálculo da cor de cada pixel. O resultado final é uma matriz de linhas e colunas de pixels, com suas respectivas cores, a qual será armazenada em memória.

Sobre esta matriz são realizadas operações matemáticas diversas, dependendo do objetivo da análise. É possível ocultar ou realçar detalhes, detectar elementos geométricos, dentre outros processamentos digitais de imagens, como será descrito no próximo item deste capítulo.

3.3 Processamento de Imagens

Ao se trabalhar com imagens, existem processamentos que podem ser executados sobre os dados de forma a facilitar a posterior obtenção de informações úteis. Como as imagens são armazenadas na memória dos computadores como dados binários, o acesso e modificação dos valores de seus dados via software é relativamente simples, havendo bibliotecas em diversas linguagens que permitem tal controle. Nesse tópico serão citadas de maneira genérica algumas das operações básicas sobre imagens mais relevantes para o presente trabalho.

3.3.1 Modificação dos valores de cor dos pixels

Modificar os valores de pixels costuma ser tão simples quanto modificar o valor de qualquer variável em uma linguagem de programação. Utilizando como exemplo a linguagem utilizada nesse projeto, os valores das cores de cada *pixel* são acessíveis através do objeto no qual a imagem foi carregada na memória, selecionando-os através da posição do *pixel*. Uma vez que se tenha acesso aos valores, é possível modificá-los livremente, atribuindo valores entre 0-255 para cada uma das cores. A forma específica de se fazer isso será mostrada mais adiante.

3.3.2 Conversão do sistema de cores

Apesar de a forma mais comum de se designar as cores em sistemas eletrônicos seja através do padrão RGB, existem também outros padrões, que utilizam lógicas diferentes de composição da cor; alguns dos principais padrões são listados abaixo, acompanhados de uma breve descrição:

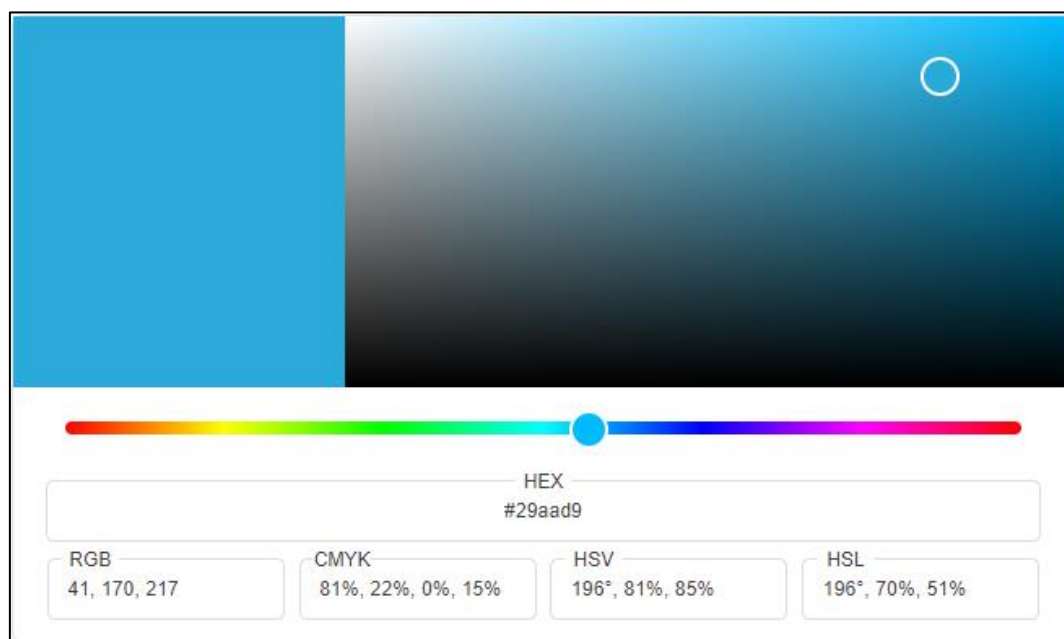
- RGB (*Red*(vermelho), *Green*(verde), *Blue*(azul)) – baseada na natureza aditiva das componentes de cor da luz. São armazenados valores correspondentes às intensidades de vermelho, verde e azul e possui ainda variações como o RGBA, que adiciona um

dado chamado de ‘Alfa’, que corresponde à transparência da cor definida.

- CMYK (*Cyan*(Ciano), *Magenta*, *Yellow*(Amarelo), *Black*(Preto)) – padrão criado para representar a composição de cores utilizando pigmentos, os quais apresentam natureza subtrativa em relação à composição, devido a absorção de luz. É muito utilizado na área gráfica.
- HSV (*Hue* (matiz), *Saturation* (Saturação), *Value* (Valor)) – onde: a matiz será o valor que determina a tonalidade da cor numa escala desde o vermelho até o violeta. A saturação determina o grau de pureza dessa tonalidade, onde quanto mais baixo o grau de pureza mais acinzentada se torna a tonalidade e o valor determina o brilho da cor.

Como existem aplicações onde um padrão de cores é preferível ao outro, a conversão entre esses padrões é uma operação muito comum no processamento de imagens. A Figura 13 exibe um exemplo com esses diferentes padrões representando uma mesma cor selecionada através de uma ferramenta de seleção de cores disponibilizada pela Google.

Figura 13: Ferramenta de seleção de cores disponibilizada pela Google



Fonte: www.google.com

Dentro desse projeto, não haverá necessidade de executar conversões entre esses diferentes padrões de cor, já que as cores são de pouca importância quando se trata de medição, mas há uma transformação que será utilizada em grande parte das funcionalidades e que também pode ser considerada uma conversão, que é a transformação de imagem colorida em imagem em escala ou tons de cinza, variando do preto puro ao branco puro. A grande vantagem de se trabalhar com imagens em tons de cinza é a menor exigência de poder computacional, tornando as operações mais rápidas; isso porque para cada pixel passa a ser necessário armazenar apenas um dado correspondente à cor.

Tomando como exemplo uma conversão de RGB para tons de cinza, essa transformação ocorre da seguinte maneira: a imagem é percorrida pixel a pixel; de cada pixel são lidos os valores das componentes RGB; os três valores passam por uma função matemática que retorna como resultado um valor único que será a nova cor, em escala de cinza, associada ao pixel. A função matemática escolhida para operar sobre os valores de RGB terá grande influência no resultado da conversão. Intuitivamente, pode-se imaginar que a média dos valores de RGB seria uma boa forma de computar o valor de cinza e, de fato, esse é um método utilizado, ficando a operação da seguinte maneira:

$$Y = \frac{R + G + B}{3} \quad (1)$$

Onde:

- Y – Valor final em escala de cinza
- R – Intensidade de vermelho
- G – Intensidade de verde
- B – Intensidade de azul

O cálculo do tom de cinza utilizando a média funciona, o problema é que utilizando a média as componentes das cores estão contribuindo de maneira igual no resultado final, o que não corresponde de maneira correta ao valor natural aproximado da cor original em escala de cinza. Para calcular de forma correta, foi criada uma função que leva em consideração a sensibilidade do olho humano às cores primitivas, dando pesos diferentes a cada uma delas. Portanto, a função correta a ser utilizada na conversão é:

$$Y = 0,299 \times R + 0,587 \times G + 0,114 \times B \quad (2)$$

O valor de Y é então armazenado, normalmente como um dado de 8 bits. A Figura 14 exibe um exemplo comparativo entre as conversões utilizando a função média e a função que atribui pesos aos valores das cores primitivas.

Figura 14: Dois métodos de conversão para escala de cinza



Fonte: Adaptada de: www.tutorialspoint.com, 2020.

3.3.3 Binarização

A binarização é outro processamento sobre as cores dos pixels que será muito utilizado no projeto. O objetivo dessa conversão é que a todos os pixels da imagem seja atribuída ou a cor preta ou a cor branca, por isso o uso termo ‘binarizar’, em função de alguma regra. Esse tipo de processo ajuda a evidenciar diferenças entre partes da imagem e será utilizado em todas as funcionalidades que necessitam de obtenção de contornos e no cálculo de áreas por contagem de pixels, que dependerá diretamente de que a imagem seja binária.

Para executar a binarização o processo é semelhante ao utilizado na conversão entre RGB para escala de cinza. A diferença é a operação utilizada para calcular o resultado final da cor. Em uma imagem já em escala de cinza e com cor de 8 bits, cada pixel conterà um valor de intensidade de cinza compreendido entre 0 e 255. O valor atual do pixel é lido e submetido à uma comparação com um valor de limiar, pertencente a mesma faixa de valores, se o valor da cor atual do pixel for inferior ao valor do limiar o novo valor atribuído ao pixel será 0, se seu valor for superior ao do limiar, o novo valor será 255. O resultado será uma imagem em preto e branco.

Figura 15: Conversão para escala de cinza seguida de binarização



Fonte: Elaborada pelo autor.

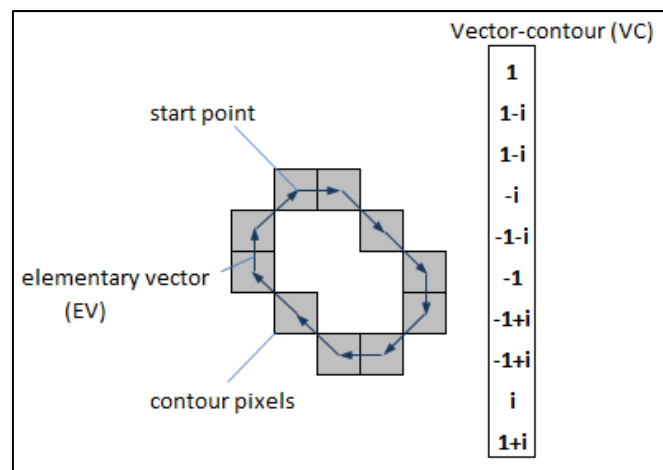
A Figura 15 apresenta um exemplo do processo de binarização a partir da foto de um tabuleiro de xadrez utilizando 160 como valor de limiar. Caso em que pixels com valor inferior a 160, foram modificados para 0, o preto absoluto, e valores superiores foram modificados para 255, o branco absoluto. É possível notar como o processo evidencia o contraste do que aparece na imagem.

Vale mencionar que existem métodos de binarização mais complexos, como a binarização adaptativa, que calcula diferentes valores de limiar para cada região da imagem, considerando as condições de luminosidade nessas regiões; nesse caso, o algoritmo determina o valor do limiar para cada pixel considerando os valores dos pixels numa região em torno dele. Isso tem efeitos muito importantes para algumas aplicações, mas não entra no escopo desse projeto.

3.3.4 Identificação de contornos

Um contorno pode ser explicado como uma curva unindo pontos contínuos, com a mesma cor, ao longo de uma borda ou fronteira. Para determinar um contorno em uma imagem, é necessário o uso de um algoritmo que percorra cada pixel avaliando sua cor e a cor dos pixels ao seu redor, seguindo por uma borda conforme encontre pixels da mesma cor até que o contorno se encerre (contorno aberto) ou reencontre seu ponto de origem (contorno fechado).

Figura 16: Ilustração de um contorno e o vetor que o representa



Fonte: www.codeproject.com, 2020.

Muitas das funcionalidades criadas durante o projeto utilizam o recurso de identificação de contornos. Na biblioteca utilizada no projeto, o OpenCV, o algoritmo aplicado é o de Suzuki & Be (1985), que não será detalhado aqui. Como a análise feita pelo algoritmo sobre os pixels leva em consideração a cor, os melhores resultados são obtidos com o uso de imagens binarizadas, sendo que

a documentação do OpenCV informa que apenas imagens binarizadas são aceitas. O algoritmo é otimizado para encontrar contornos brancos em um fundo preto, por isso o programa aqui descrito foi criado possibilitando a binarização inversa, a fim de que o usuário possa tratar a imagem até que chegue ao estado desejado, binarizada com fundo preto e objeto em branco.

Como resultado, a função de busca de contornos retorna uma lista de contornos, em que cada posição contém um dos contornos identificados na imagem, com as posições de todos os pixels pertencentes a esse contorno. Além disso, a biblioteca possui diversos recursos adicionais relacionados aos contornos identificados, como vértices, área, perímetro, momentos (permitem o cálculo do centro de massa), entre outros.

3.4 Fundamentos de Metrologia aplicados no contexto do trabalho

Dado que este trabalho envolve o desenvolvimento de um software de visão computacional aplicada à medição, foi necessário o estudo de aspectos fundamentais e aplicados de metrologia.

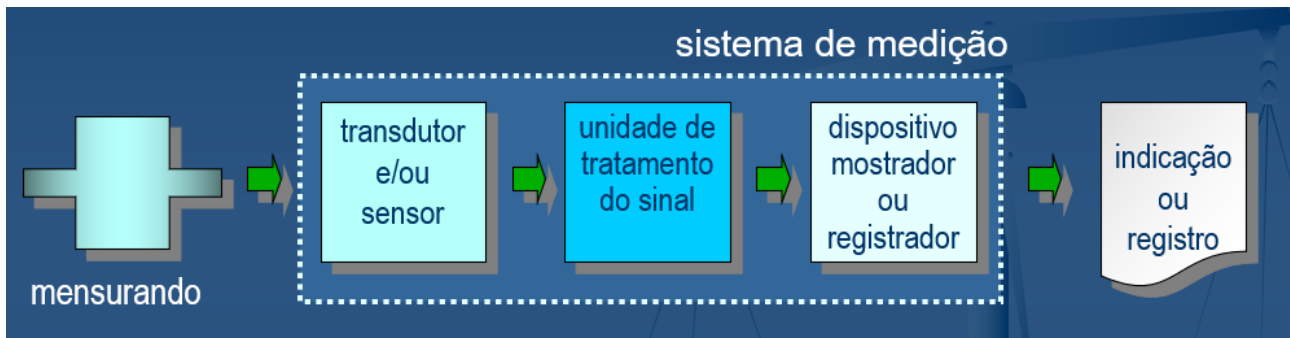
Um sistema de medição pode ser definido como um instrumento ou equipamento capaz de realizar medições ao interagir com um dado mensurando. Genericamente, ele pode ser dividido em três módulos: o sensor/transdutor, a unidade de tratamento de sinais e o dispositivo mostrador ou registrador.

O transdutor é a parte do sistema que interage diretamente com o mensurando, gerando algum tipo de sinal proporcional à grandeza que está sendo medida, que pode ser descrito por uma função matemática bem definida, geralmente linear. Um exemplo presente neste trabalho é o fotossensor, o qual captura a luz nele incidente gerando um sinal elétrico proporcional à intensidade da luz.

A unidade de tratamento de sinal é responsável por executar algum tipo de processamento no sinal recebido do transdutor e obter um sinal que represente a medição da grandeza medida. Isso pode ocorrer na forma de cálculos matemáticos, tratamento de sinais elétricos (geralmente envolvendo amplificação do sinal) ou mesmo por meio de fenômenos mecânicos. Um exemplo conveniente seria o processamento dos sinais elétricos obtidos pelos fotossensores de uma câmera digital a fim de gerar a fotografia. Alguns sistemas de medição mais simples não possuem este módulo.

O dispositivo mostrador torna o resultado da medição disponível ao usuário de alguma forma, seja marcando um valor em uma escala, exibindo-o em uma tela ou registrando em algum dispositivo de memória. Como exemplos é possível citar desde o cursor de um paquímetro, que permite a leitura da medição de acordo com sua posição na escala até a interface de um software de medição ou o armazenamento dos resultados em memória, em sistemas mais complexos.

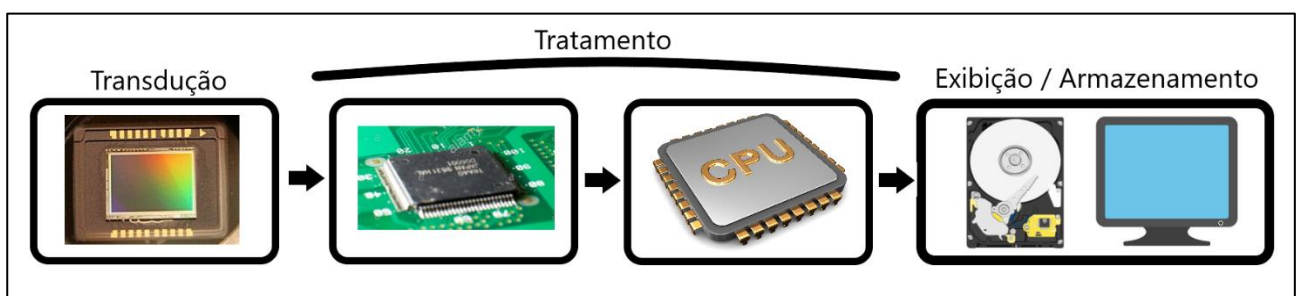
Figura 17: Representação de um sistema de medição



Fonte - Fundamentos da Metrologia Científica e Industrial - Capítulo 4

No presente trabalho estes módulos estão presentes conforme mostra a representação na Figura 18. O transdutor é o sensor de imagem que converte a intensidade luminosa em um sinal elétrico, que é processado eletronicamente nos circuitos de condicionamento de sinais presentes e matematicamente pelos softwares de medição, até que o valor do mensurando seja obtido, armazenado e indicado na tela do computador.

Figura 18: Módulos do sistema de medição



Fonte: Elaborada pelo autor.

Ao término do processo de medição, o sistema apresentará um resultado, geralmente na forma de um número. Mas esse valor numérico não é a representação exata da grandeza medida, e sim valor aproximado, que pode estar mais ou menos distante do valor real. O quão distante do valor real será aceitável, depende da aplicação para a qual se destina a medição. Portanto, tão importante quanto saber o resultado, é saber o quanto se pode confiar nesse resultado. De tal forma que uma medição só está completa, quando acompanhada da incerteza a ela associada.

Sistemas de medição possuem uma série de parâmetros metrológicos que definem, não só o grau de confiança que se pode ter em seus resultados, como muitas outras características. A seguir serão definidos alguns desses parâmetros, selecionados devido à sua relevância neste trabalho.

- **Faixa de medição:** Compreende o intervalo entre os valores mínimo ao máximo que o sistema está preparado para medir, garantindo o nível de incerteza especificado. Esse

intervalo é menor ou no máximo igual à faixa de indicação.

- Resolução: É a menor variação da grandeza do mensurando que o sistema é capaz de perceber e indicar. Em instrumentos com mostrador digital, a resolução corresponde ao incremento digital.
- Erro sistemático: O erro sistemático é uma estimativa do erro que tende a persistir através de diversas medições. Idealmente, seria a diferença entre a média de infinitas medições efetuadas sobre o mesmo mensurando, sob as mesmas condições, e o resultado de referência. Na prática, utiliza-se a tendência para estimar o erro sistemático. A tendência é calculada como a diferença entre a média de um número finito de medições e o valor de referência. Em muitos casos, é possível utilizar a tendência para aplicar correções nas medições, isso é feito somando o valor da tendência com sinal invertido ao resultado preliminar.
- Repetibilidade: Representa uma faixa de valores dentro da qual tende a se situar o erro aleatório quando realizadas diversas medições nas mesmas condições. Essa faixa é definida estatisticamente em função de um grau de confiabilidade escolhido. O valor comumente utilizado para confiabilidade é de 95%, podendo variar dependendo das necessidades da aplicação. Isso significa que ao ser efetuada uma medição, pode-se ter 95% de certeza de que o erro aleatório estará dentro da faixa de valores da repetitividade.
- Erro Máximo: É o módulo máximo estimado para o erro de medição do sistema em toda a sua faixa de medição, levando em conta tanto erros sistemáticos como aleatórios. Significa que o resultado obtido não pode estar mais distante do valor real do mensurando do que esse módulo.

Uma etapa indispensável no desenvolvimento de sistemas de medição é a sua calibração, que consiste em determinar a correspondência dos valores indicados pelo sistema quando comparado com uma dimensão de referência, estabelecida com incerteza baixa o suficiente para poder ser utilizada nesta calibração. A calibração pode ser realizada de modo direto ou indireto.

Na calibração direta o sistema de medição interage com uma medida materializada de referência que tenha seu valor verdadeiro conhecido com incerteza preferencialmente dez vezes menor do que a do instrumento a ser calibrado. O valor medido pelo sistema de medição é comparado com o valor conhecido da medida materializada e os ajustes de calibração são feitos até que a medição resulte em um valor dentro da incerteza esperada para o sistema.

A calibração indireta é geralmente utilizada para grandezas que não permitem o uso de medidas materializadas, como a velocidade de um automóvel, por exemplo. Na calibração indireta o mensurando é gerado por um sistema auxiliar e medido simultaneamente pelo sistema de medição a ser calibrado e por um sistema de medição padrão, cujo erro seja preferencialmente menor do que um décimo do erro apresentado pelo sistema a ser calibrado. Os valores medidos pelos dois sistemas são comparados e o sistema a ser calibrado é ajustado até que apresente resultados satisfatórios.

Neste trabalho, a calibração do sistema foi realizada de modo indireto, como será descrito no capítulo de validação experimental do software de medição.

3.5 Visão Computacional aplicada na metrologia dimensional

Grande parte das informações que o ser humano recebe do mundo à sua volta vem por meio do sentido da visão. É através dela que humanos e outros animais são capazes de perceber obstáculos em seu caminho, ter noção de distâncias, interpretar cores, reagir a ameaças visíveis e tantas outras coisas relacionadas à sua interação com o mundo externo.

No contexto tecnológico, pode-se dizer que a visão computacional é o campo de estudo que tem por objetivo trazer a visão e seus benefícios ao mundo das máquinas. Fazem parte disso tanto o desenvolvimento de *hardware* aplicado a captura e processamento de imagens, quanto de *softwares* capazes de utilizar o conteúdo dessas imagens para extrair as informações utilizadas nas diversas aplicações.

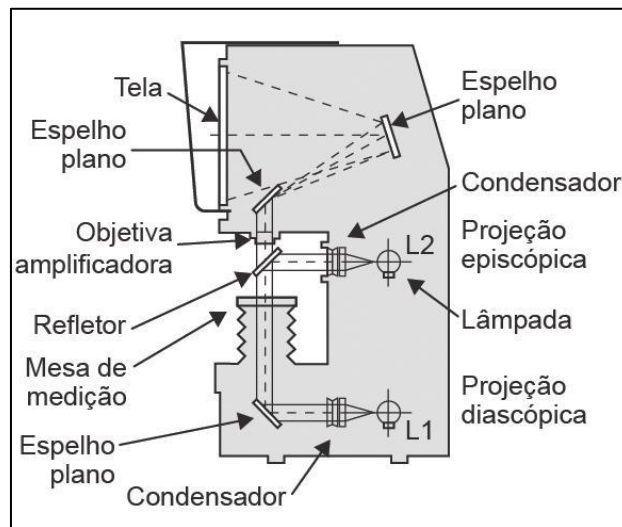
Existem muitas áreas onde já são aplicados sistemas com visão computacional, das quais pode-se citar: segurança, medicina, robótica, controle de qualidade, esportes, jogos eletrônicos, comunicações e a metrologia. Alguns exemplos de aplicações específicas que podem ser citados são: os sistemas de identificação e contagem de objetos em bagagens e de reconhecimento facial na segurança de aeroportos; uso em veículos autônomos, que processam constantemente imagens do que está ao seu redor para rápida tomada de decisão e os atualmente muito populares códigos no padrão QR(*Quick Response*), onde é gerada uma imagem que será decodificada com a câmera e o software de um celular, permitindo funcionalidades diversas.

Dentro da metrologia dimensional existem algumas aplicações já consolidadas envolvendo óptica e imagens, como exemplos serão abordados aqui brevemente o projetor de perfis e o software de medição por visão QSPAK, desenvolvido pela empresa japonesa Mitutoyo.

3.5.1 Projetor de Perfil

O projetor de perfis é um sistema de medição óptico baseado no uso de lentes. O projetor de perfis vertical (Figura 19 e Figura 20) possui uma mesa de vidro sobre a qual podem ser colocadas peças pequenas, e utiliza uma fonte de luz e um sistema de lentes para projetar o perfil da peça com uma ampliação em uma tela com um cursor. A mesa é posicionada através de dois controles giratórios que controlam seu movimento nos eixos horizontais e a medição é geralmente feita através de dois micrômetros, um em cada eixo. Nos equipamentos mais modernos, existe eletrônica e softwares embarcados que permitem processamentos mais complexos sobre os dados adquiridos.

Figura 19: Esquema interno de um projeto de perfil



Fonte: AKKARI, A. C. S. - 2017

Projetores de perfil são muito úteis para medição de peças muito pequenas e de formatos complexos que não poderiam ser medidas através de instrumentos convencionais de toque, como por exemplo as engrenagens de um relógio de pulso.

Figura 20- Starret VB300DRO



Fonte: www.starrett.com.br, 2020.

3.5.2 Máquinas de Medir CNC

Os projetores de perfil citados no item 3.5.1 são soluções adequadas para a medição de peças em pequenas séries, mas, no contexto da produção seriada há soluções mais adequadas como as máquinas de medir CNC ópticas. Estes equipamentos possuem uma mesa onde a peça a medir é apoiada, e uma câmera se desloca de modo programado para adquirir imagens em regiões de interesse previamente definidas pelo programa de medição.

As imagens são adquiridas e processadas, permitindo uma inspeção dimensional da peça muito rápida e com boa exatidão. Os sistemas possuem alta capacidade de ampliação das imagens e recursos para otimizar a iluminação de modo a obter a melhor definição possível de bordas dos elementos geométricos a medir. A Figura 21 ilustra um exemplo destes equipamentos.

Figura 21: Máquina de medir CNC



Fonte: Mitutoyo

3.5.3 Sistemas de medição óptica distribuídos

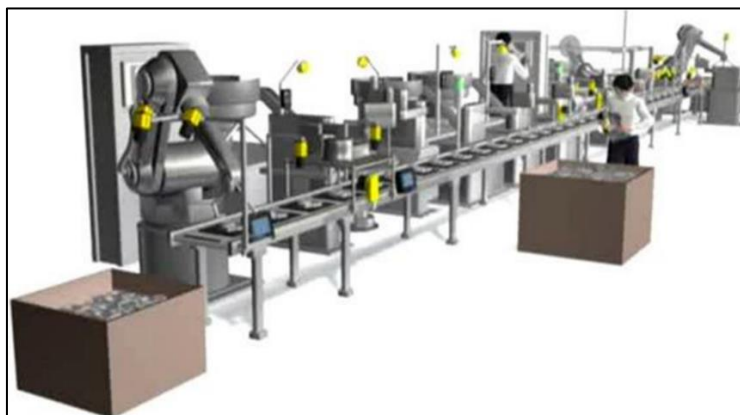
Os equipamentos descritos anteriormente possuem em comum a característica de integrarem todos os módulos do sistema de medição em uma só máquina.

Há também no mercado uma solução de medição óptica composta por câmeras distribuídas no processo produtivo, integradas diretamente na manufatura e realizando a inspeção dimensional e também a inspeção atributiva (verificação da presença/ausência de componentes e elementos na peça).

Esta solução perde em exatidão em relação às máquinas de medir, mas ganha em flexibilidade operacional e em custo, permitindo maior customização da solução de medição para a demanda de

inspeção. A Figura 22 ilustra o emprego de câmeras distribuídas na produção para a inspeção dimensional e atributiva de produtos.

Figura 22: Sistema de medição com câmeras distribuídas



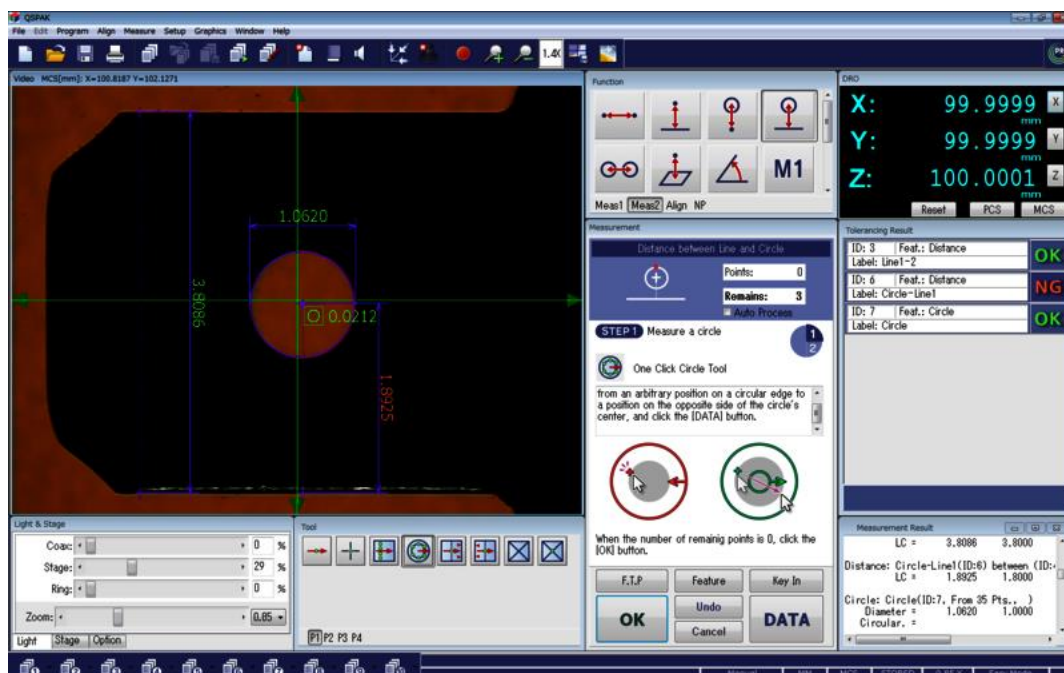
Fonte: Cognex, 2020

3.5.4 Softwares de processamento de imagens para metrologia dimensional

Independentemente do tipo de solução de medição óptica, todos os sistemas de visão computacional empregados na área de metrologia dimensional necessitam de um software de medição para fins de gerenciamento e programação da medição, aquisição e processamento das imagens e indicação dos resultados para os usuários.

É possível o desenvolvimento de soluções customizadas para cada necessidade de medição e existem soluções de emprego amplo para processamento de imagens e determinação das dimensões das peças. Um desses exemplos é o software QSPAK da empresa Japonesa Mitutoyo, que permite a medição de diversos recursos de peças em tempo real através de processamento de imagens. Ele é preparado para trabalhar em conjunto com máquinas de medição e sistemas de câmeras.

Figura 23: Menu do programa QSPAK



Fonte: <https://shop.mitutoyo.eu>, 2020

Alguns dos recursos presentes são:

- Detecção e medição de formas geométricas;
- Detecção de bordas;
- Uso de modelos ideais das peças para análise comparativa;
- Integração com máquinas CNC e de medição por coordenadas;
- Integração com algumas máquinas da empresa, permitindo controlar parâmetro como a iluminação, foco, etc;

O QSPAK é apenas um exemplo dentre várias soluções disponibilizadas por diversas empresas no mercado, com as mais diversas capacidades.

3.6 Ferramentas de desenvolvimento de software utilizadas

Como etapa inicial de projetos na área de programação é importante que se reserve um tempo para avaliar quais linguagens e ferramentas serão as mais adequadas para a resolução do problema apresentado.

Avaliando as opções de ferramentas disponíveis para trabalhar com processamento sobre imagens, considerando o requisito de desenvolver uma interface gráfica e levando também em conta a experiência e habilidades do autor, optou-se pelo uso da linguagem Python com a biblioteca OpenCV para sistemas de visão e a biblioteca PyQt5 para o desenvolvimento da interface gráfica.

3.6.1 Python

O Python é uma linguagem de programação de alto nível e alta versatilidade. É uma linguagem de fácil aprendizado, permitindo o uso dos paradigmas de orientação a objeto e de programação estruturada. É também uma linguagem multiplataforma e com uma grande comunidade de usuários, o que enriquece muito a quantidade de ferramentas e bibliotecas já criadas para as mais diversas aplicações. Outras vantagens são sua sintaxe intuitiva e de fácil leitura, a abundância de documentação e sua modularidade, facilitando a reutilização e compartilhamento de código.

A linguagem tem sido utilizada por profissionais de diversas áreas, sendo atualmente muito popular no universo da ciência de dados, no desenvolvimento de inteligência artificial e no aprendizado de máquina.

Nesse trabalho, o Python serviu de base para todo o desenvolvimento, sendo suficiente para o cumprimento de todos os requisitos definidos para o projeto, desde o tratamento de imagens, passando pelas operações matemáticas, até a interface gráfica com o usuário. A versão do Python utilizada é a 3.8.0. Para as operações matemáticas o módulo ‘math’, nativo da linguagem, foi usado na maioria dos cálculos e o módulo ‘sympy’ foi usado para resolução de sistemas de equações.

3.6.2 OpenCV

É uma biblioteca de software para visão computacional e aprendizado de máquina de código aberto e escrita em C/C++. A biblioteca contém diversas otimizações para o tratamento de imagens, com algoritmos de uso comum na área já prontos para uso. Uma vez que o processamento de imagens é muito dispendioso computacionalmente, o OpenCV apresenta grande vantagem por muitos de seus algoritmos serem otimizados para trabalhar na unidade de processamento gráfico, caso haja uma no computador.

Dentro do Python, o OpenCV está presente na forma do módulo `opencv-python`, que possui ótima documentação e é de fácil instalação através do pip, o gerenciador de pacotes nativo da linguagem. A versão do módulo OpenCV utilizado no projeto é a 4.1.2. A instalação do módulo Numpy, também feita via pip, é um requisito adicional para o funcionamento da biblioteca `opencv-python`.

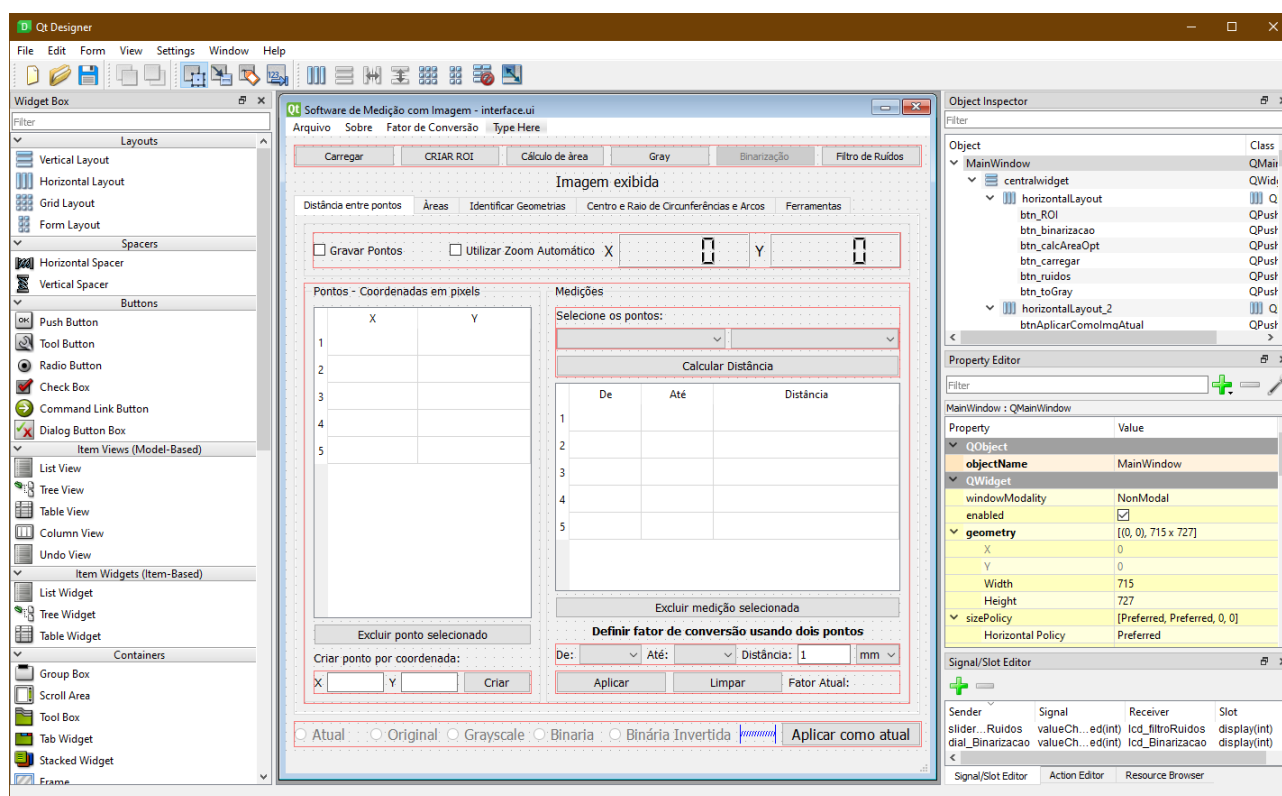
3.6.3 PyQt5 e Qt Designer

Para o desenvolvimento da interface gráfica do programa apresentado nesse trabalho, a ferramenta escolhida foi a biblioteca PyQt5, que é baseada na biblioteca Qt v5, um *framework* multiplata-

forma para desenvolvimento de interfaces gráficas escrito em C++. O PyQt5 é um módulo que permite a chamadas das funções do Qt v5 dentro da sintaxe do Python, apresentando alto desempenho e também é instalado facilmente através do gerenciador pip.

O Qt possui uma ferramenta muito útil para o desenvolvimento ágil chamada de Qt Designer. O Qt Designer é um programa que permite a criação da interface gráfica de maneira visual, através da seleção de componentes gráficos em um menu, com posicionamento feito com o cursor do mouse e diversas opções de customização. A Figura 24 apresenta uma captura de tela do Qt Designer sendo utilizado durante o projeto.

Figura 24: Interface do Qt Designer com o projeto do software de medição aberto



Fonte: Elaborada pelo autor.

Como resultado, o Qt Designer exporta a interface criada como um arquivo no formato ‘.ui’, que contém uma descrição do que foi gerado na sintaxe XML. Para gerar o código Python referente a interface criada, utiliza-se o comando “pyuic5”, uma ferramenta instalada junto com o PyQt5. O comando gera um módulo Python contendo o código necessário para renderizar a interface criada no Qt Designer.

Importando o módulo Python gerado, é possível criar um objeto que referencia a interface e possibilita interagir com seus componentes via código, permitindo leitura e modificação de valores, tratamento de eventos e tudo mais quanto for necessário.

A documentação de referência para o PyQt5 pode ser encontrada no seguinte endereço: <https://www.riverbankcomputing.com/static/Docs/PyQt5/>.

3.7 Metodologia

Nos meios acadêmicos e científicos, afirmações precisam ser fundamentadas em fatos, os quais precisam estar sustentados pela lógica e por experimentos que os confirmem. A metodologia científica é um conjunto de ferramentas que orientam o desenvolvimento do conhecimento científico, com o objetivo de garantir a confiabilidade daquilo que é descoberto. O registro dos métodos é importante para outros possam replicar os experimentos, a fim de confirmar ou mesmo refutar as conclusões prévias.

“A metodologia da pesquisa num planejamento deve ser entendida como o conjunto detalhado sequencial de métodos e técnicas científicas a serem executados ao longo da pesquisa, de tal modo que se consiga atingir os objetivos inicialmente propostos e, ao mesmo tempo, atender aos critérios de menor custo, maior rapidez, maior eficácia e mais confiabilidade de informação.” (BARRETO; HONORATO, 1998).

Existem diversos modelos de metodologia disponíveis, cada qual voltado mais especificamente para um tipo de trabalho. Considerando que o presente trata de desenvolvimento de *software*, optou-se pelo *Scrum*, uma metodologia criada com esse objetivo.

3.7.1 Metodologia Scrum para projetos de software

a) Agentes participantes no Scrum

São três os agentes que fazem parte da dinâmica de trabalho no Scrum. O primeiro é o *Product Owner* (dono do produto), que é responsável por especificar as características desejadas para o produto, através de uma lista chamada de *Product Backlog*, e avaliar o andamento do projeto, podendo propor alterações a fim de que o produto final saia de acordo com o desejado. Nesse trabalho, tanto o autor quanto o orientador cumprem esse papel, participando das definições do projeto.

O segundo é o *Scrum Master* (mestre de Scrum), um membro da equipe de desenvolvimento que é responsável por garantir que as práticas do Scrum sejam mantidas pela equipe. Ele atua como facilitador nas reuniões e ajuda a remover obstáculos que estejam impedindo o progresso das tarefas.

Por último temos o *Scrum Team* (equipe de Scrum), que é a equipe de desenvolvimento, podendo ser composta por profissionais de áreas diversas, onde cada membro recebe atribuições para

que contribua no cumprimento das tarefas. No referido projeto, não há uma equipe de desenvolvimento, portanto o autor é quem desempenha esse papel.

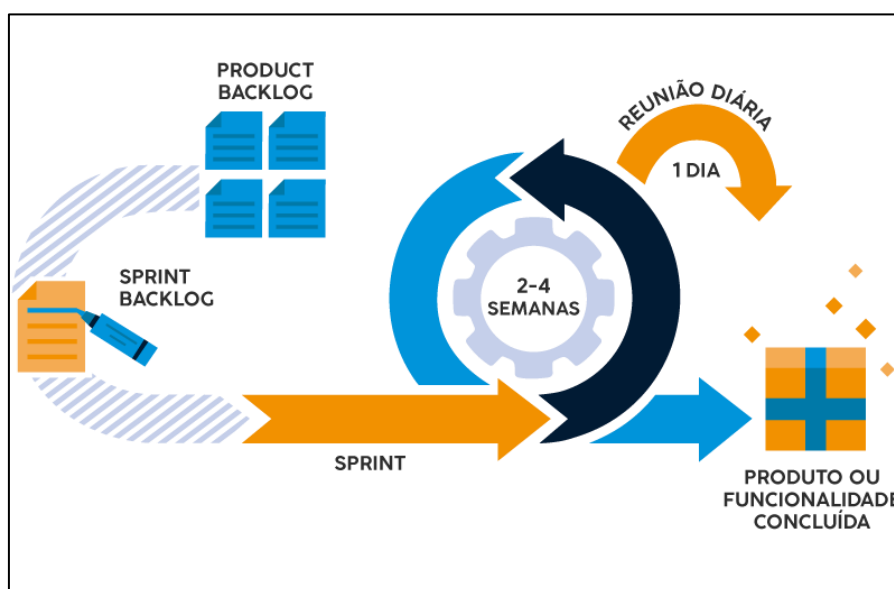
b) Funcionamento da metodologia

Em um projeto gerenciado via Scrum o tempo é dividido em períodos chamados de *Sprints*, com duração definida no início do projeto. Antes de cada Sprint os participantes realizam uma reunião de planejamento onde são escolhidas tarefas do Product Backlog a serem cumpridas durante seu período.

No início de cada dia são realizadas reuniões curtas, chamadas de Daily Scrum, onde cada membro da equipe relata seu progresso do dia anterior, menciona seus planos para o dia atual e tem a chance de relatar dificuldades e receber sugestões.

Ao final da Sprint são realizadas duas reuniões. A primeira é a de revisão, onde os resultados são apresentados ao Product Owner para avaliação. A segunda é a reunião de retrospectiva, onde são levantadas as ocorrências positivas e negativas da Sprint e são feitos planos para evitar que as negativas se repitam em Sprints futuras.

Figura 25: Funcionamento do Scrum



Fonte: www.tecnicon.com.br, 2020.

Nesse projeto, a duração escolhida para as Sprints foi de uma semana, com planejamento feito pelo autor ao início de cada semana e reuniões de acompanhamento com o professor orientador.

3.7.2 Kanban

Como ferramenta auxiliar foi utilizado o Kanban, um método visual de gestão de tarefas criado dentro da Toyota. A ideia do Kanban é criar cartões que representam as tarefas e posicioná-los em colunas presentes em um quadro, cada qual representando um possível estado da tarefa. Nesse projeto, foi utilizada uma ferramenta digital de Kanban disponível na internet e de uso gratuito, o KanbanFlow, com quadros de tarefas divididos nas colunas: *'To do'* (a fazer), *'In progress'* (em progresso) e *'Done'* (feita).

4. CONCEITO DO PROJETO:

4.1 Idealização e planejamento das funcionalidades

Levando em conta os requisitos definidos para o projeto, o tempo disponível para o desenvolvimento e o conjunto de ferramentas de software selecionadas, fez-se um levantamento das funcionalidades a serem desenvolvidas a fim de que os objetivos do projeto fossem cumpridos.

Nesse tópico essas funcionalidades serão apresentadas divididas de acordo com sua contribuição nos diferentes objetivos.

4.1.1 Funcionalidades globais

- Capacidade de abrir arquivos de imagem de diversos formatos;
- Seleção de pixels na imagem com o ponteiro do mouse para criação de pontos;
- Indicação na interface gráfica das coordenadas do ponteiro do mouse sobre a imagem durante a criação de pontos;
- Listagem dos pontos criados na forma de uma tabela com numeração e as coordenadas dos pontos em X e Y;
- Opção de exibição de um zoom dinâmico sobre uma região em torno da posição do ponteiro na imagem a fim de facilitar a visualização do ponto a ser selecionado;
- Criação de região de interesse;
- Indicação visual dos pontos criados desenhada na imagem;
- Criação manual de pontos informando as coordenadas desejadas;
- Navegação entre as versões da imagem (original, em escala da cinza, binarizada) através de botões de seleção;
- Conversão da imagem original para uma imagem em escala de cinza;
- Binarização de imagens em escala de cinza com limiar selecionado por componente de seleção giratório, com opção de binarização invertida;
- Ferramenta de filtro de ruídos com seleção do tamanho do *kernel* com seletor do tipo *slider*;

4.1.2 Funcionalidades para a obtenção da referência dimensional

- Seleção dos pontos de referência através de menu *combo box* para o cálculo do fator de conversão baseado na distância entre dois pontos;
- Algoritmo para obtenção automática do fator de conversão usando bloco padrão acionado via evento de clique de um botão;

- Seleção da unidade de medida de comprimento via botão, tendo como opções: mm, cm, m e km;

4.1.3 Funcionalidades para a medição da distância entre pontos

- Seleção de dois pontos, através de menu *combo box* para o cálculo da distância;
- Cálculo da distância acionado por evento de clique em um botão;
- Exibição das medições criadas na interface através de uma tabela;
- Desenho de retas na imagem representando as medições de comprimento realizadas;
- Possibilidade de exclusão de medições da tabela;

4.1.4 Funcionalidades para a identificação de formas geométricas

- Algoritmo de identificação automática de algumas formas geométricas comuns por contagem de vértices do contorno, acionado por um botão na interface;
- Obtenção das áreas das formas identificadas;
- Obtenção do perímetro das formas identificadas;
- Obtenção do centro de massa das formas identificadas;
- Exibição da lista de formas identificadas em uma tabela, contendo as informações de forma, área, perímetro, centro de massa e número de vértices;

4.1.5 Funcionalidades para a medição de áreas

- Algoritmo de cálculo da área por contagem de pixels acionado por botão.
- Como no caso do cálculo por contagem de pixels só pode haver um objeto na imagem, ficando, portanto, o fundo da imagem em preto ou branco e o objeto na cor restante, dependendo da forma como o usuário decidiu binarizar a imagem, a interface deve exibir tanto a área relacionada aos pixels brancos quanto aos pretos, cabendo ao usuário identificar a área correta pela cor;
- Identificação de contornos ativada por botão para o cálculo de área de contornos usando o recurso nativo do OpenCV;
- Seleção do contorno alvo dentre os identificados com menu *combo box*;

4.1.6 Funcionalidades para a medição de circunferências e arcos

- Para o modo de seleção manual de pontos, seleção feita via *combo box*, com opção de exibir ou não os pontos selecionados na imagem após a seleção;

- Algoritmo de cálculo do centro e raio da circunferência acionado por clique de um botão em ambos os métodos;
- Para o modo de seleção automática dos pontos, algoritmo de detecção de circunferências e seleção de três pontos dispersos pelo contorno, acionado por botão;
- Valores de raio, diâmetro e das coordenadas do centro encontrados exibidos na interface para ambos os métodos;

4.1.7 Funcionalidades para a medição de ângulos

- Seleção dos quatro pontos via menus *combo box*;
- Acionamento do algoritmo de cálculo via botão;
- Exibição do resultando em graus na interface;

4.2 Requisitos metrológicos e operacionais

Como este trabalho propõe o desenvolvimento de um software como parte de um sistema de medição, há que se definir requisitos metrológicos que expressem características desejáveis para o sistema.

- a) Faixa de medição: Como a medição ocorre a partir de imagens, seria possível definir uma faixa de medição bastante ampla. No entanto, a aquisição de imagens de objetos de grandes dimensões irá degradar a resolução da imagem, visto que a relação pixel/mm irá ser reduzida. Deste modo é proposto que o sistema de medição seja utilizado para a medição de peças com dimensões entre 0 e 200 mm.
- b) Erro máximo: Para estas dimensões planejou-se como requisito de exatidão um erro máximo (mm) de $(0,1 + L/100)$, sendo L a dimensão medida, em mm.
- c) Tempo de medição: Característica importante e indicativa da praticidade operacional do sistema, o tempo de medição total foi definido em 1 minuto, incluindo o referenciamento com o padrão dimensional, a aquisição e processamento das imagens da peça e a determinação do resultado de medição.

5. DESENVOLVIMENTO DO PROJETO

Feito o levantamento inicial das funcionalidades desejadas, procedeu-se ao desenvolvimento do projeto. Inicialmente o desenvolvimento foi dividido em etapas semanais, onde para cada semana foram designadas algumas funcionalidades a serem desenvolvidas a fim de que se cumprisse com os prazos estabelecidos. Na prática, devido à contratempos diversos, o desenvolvimento não transcorreu com a linearidade esperada, o que implicou em atrasos que fizeram com que algumas das funcionalidades planejadas acabassem por não ser implementadas.

Esse capítulo trará explicações sobre o funcionamento do software e os fluxogramas das principais partes com as devidas explicações. Devido à grande quantidade de código escrito, não será possível abordar todas as funcionalidades de maneira detalhada; portanto, o capítulo tratará de maneira completa apenas da implementação das funcionalidades principais, com muitas das funções auxiliares sendo apenas citadas, principalmente aquelas relativas à manipulação da interface gráfica que, apesar de terem tomado grande parte do tempo de desenvolvimento, não são o foco deste trabalho.

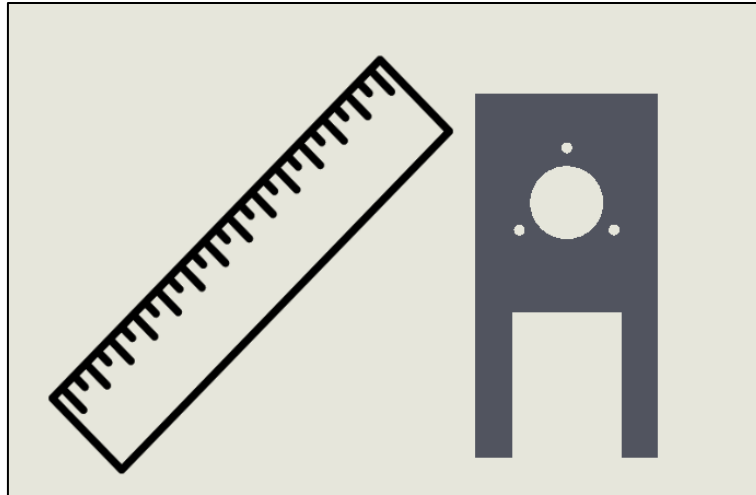
5.1 Obtenção da referência dimensional

Independentemente do método utilizado para medição com uso de imagens, alguma forma de referenciamento sempre será necessária. O método de referenciamento escolhido para esse projeto depende de uma referência que deve estar contida em cada imagem utilizada; referência na qual serão baseadas todas as medições. Devido a isso, uma das partes mais importantes do projeto é a forma como se obtém essa referência. Nesta tarefa, o objetivo será obter um fator de conversão que relacione matematicamente uma quantidade de pixels a um valor de distância.

5.1.1 Referenciamento com base em instrumentos ou peças medidas

Esse método de obtenção da referência depende diretamente do usuário. Primeiramente, é necessário que a fotografia seja obtida já contendo a medida de referência que se deseja utilizar; para isso, o usuário deverá posicionar o objeto com a face a ser medida exposta na direção da câmera. Ao lado do objeto deve ser posicionado o objeto com a medida de referência. O objeto de referência pode ser um instrumento de medição que exiba uma escala visível na fotografia, como uma régua ou um paquímetro, ou um objeto qualquer com uma medida conhecida com o maior grau de precisão possível. Deve-se tomar o cuidado de posicionar o mensurando e a referência de forma que as faces da referência e da região alvo da medição estejam alinhadas, idealmente no mesmo plano.

Figura 26: Representação de uma peça ao lado de uma régua



Fonte: Elaborada pelo autor.

Utilizar como referência uma distância contida na própria peça também é possível. Por exemplo, caso se deseje medir um perfil complexo, pode-se medir uma aresta desse perfil e usá-la como referência para medição de todas as outras características.

Uma vez com a foto obtida, o processo de obtenção do fator de conversão consiste de selecionar os dois pontos que definem a distância de referência na imagem e calcular a distância em pixels entre eles. De posse dos valores da distância real de sua representação em pixels, calcula-se o fator de conversão utilizando a seguinte equação:

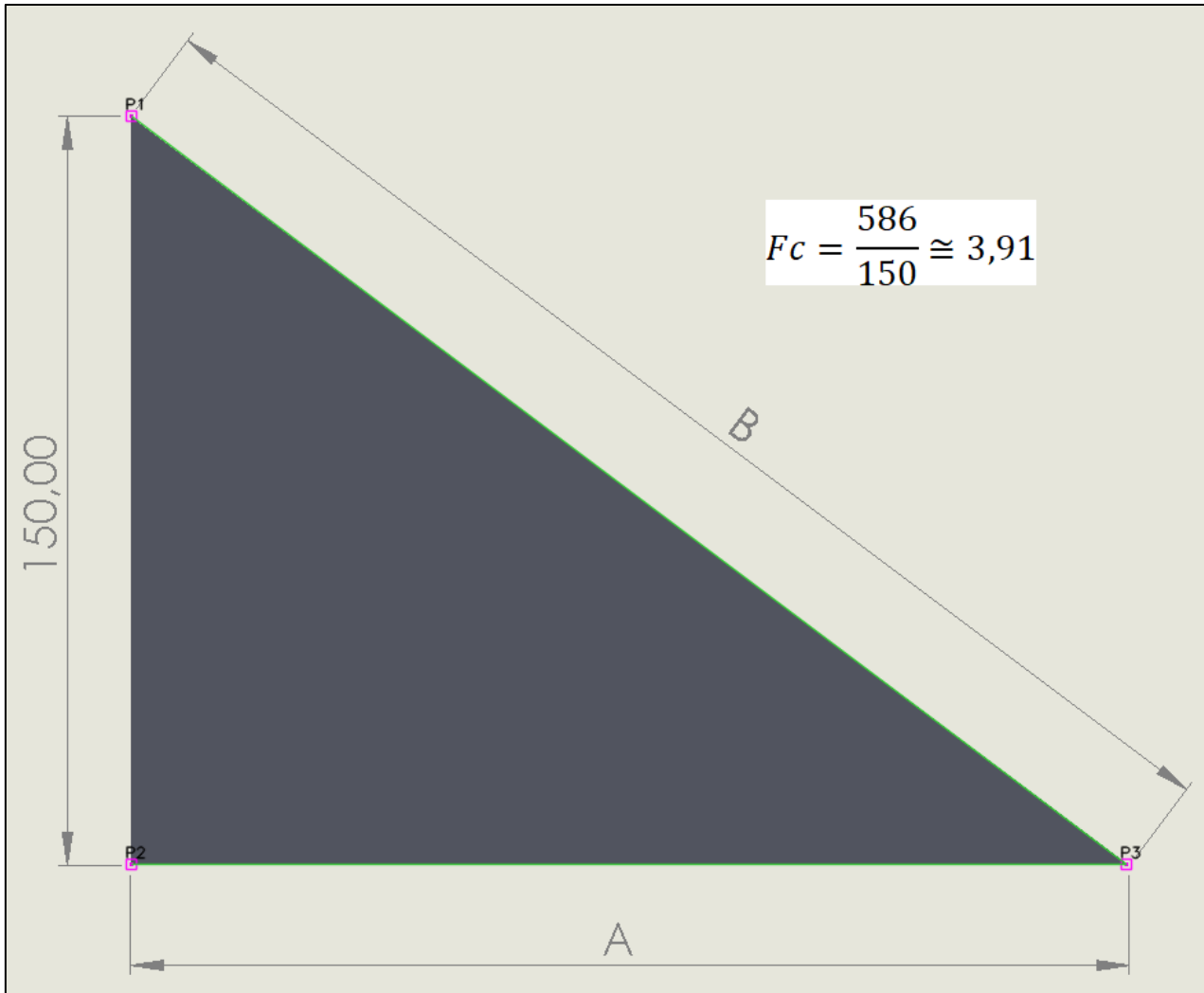
$$F_c = \frac{distPx}{distReal} \quad (3)$$

Onde:

- F_c – fator de conversão, em px/unidade de comprimento;
- $distPx$ – distância em pixels calculada entre os pontos marcados sobre a referência;
- $distReal$ – distância real em unidade de comprimento;

O resultado significa a quantidade de pixels que equivale a uma unidade de comprimento. Como exemplo, o triângulo retângulo mostrado na Figura 27 foi criado com lados 150mm e 200mm, resultando em 250mm de hipotenusa. Utilizando como base a distância conhecida de 150mm se obteve o fator de conversão de 3,91 px/mm calculado a partir dos pontos P1 e P2. Com base nesse fator, foram obtidas as distâncias A e B, através da equação 3, conforme apresentado na Tabela 1.

Figura 27: Cálculo do fator de conversão



Fonte: Elaborada pelo autor.

Tabela 1: Valores do teste do cálculo do fator de conversão

	Distância (px)	Referência (mm)	Medição (mm)
A	778	200,00	199,40
B	976	250,00	249,52

5.1.2 Referenciamento utilizando bloco padrão

Blocos-padrão são medidas materializadas prismáticas, construídas com a máxima exatidão possível de modo a ser empregados como referência na área de metrologia dimensional para operações de medição e de calibração de instrumentos. Podem ser encontrados em aço ou cerâmica e a sua exatidão depende da incerteza com que a sua dimensão é determinada, sendo tipicamente abaixo de 0,001 mm o afastamento da sua dimensão real em relação à sua dimensão nominal.

Por ter alta exatidão e ser de geometria simples e fácil de ser medida em sistemas ópticos, é bastante utilizado para a determinação da relação pixel/mm.

Figura 28: Conjunto de blocos-padrão



Fonte: Mitutoyo.

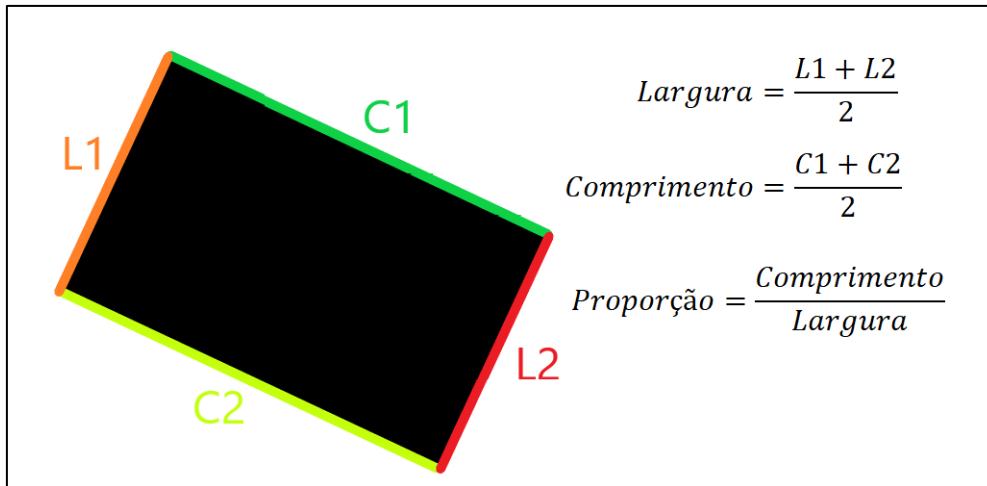
Neste trabalho o emprego de blocos-padrão foi planejado para automatizar o processo de obtenção do fator de conversão e evitar a influência do usuário na tarefa, dispensando a necessidade de selecionar pontos manualmente. Desse modo, o plano foi utilizar um bloco padrão como objeto de referência, identificá-lo na imagem e usar uma de suas medidas no cálculo do fator de conversão.

Devido à limitação de acesso ao laboratório da instituição durante o período de desenvolvimento do projeto, que impossibilitou o uso de um bloco padrão real para realização de testes, esse método foi implementado com base em um bloco padrão simulado através de um retângulo adicionado na imagem criada digitalmente. O processo pode ser dividido em três partes: identificação do bloco padrão na foto, medição da largura do bloco em pixel a ser usada como valor de referência e o cálculo do fator de conversão.

Para a identificação do bloco-padrão na imagem foram empregados a conversão para escala de cinza e posterior binarização, em seguida é utilizado o processo para identificação de formas, onde são obtidos todos os quadriláteros presentes na imagem. Após a obtenção dos quadriláteros, é necessário identificar qual deles é o bloco padrão. Para isso, definiu-se uma proporção entre o comprimento (maior dimensão) e a largura (menor dimensão) do retângulo que o representa, proporção essa com valor escolhido de 1,5, obtido pela divisão do comprimento pela largura. Esse valor é atribuído a uma variável dentro do programa, podendo ser alterado conforme a proporção do bloco padrão utilizado.

De cada quadrilátero na imagem são retirados quatro pontos de vértice; em seguida, são calculadas as distâncias entre um vértice e outro, resultando em quatro valores de distância, dois correspondendo ao comprimento e os outros dois à largura do quadrilátero (Figura 29), as médias desses valores são usadas no cálculo da taxa de proporção desse quadrilátero.

Figura 29: Ilustração da obtenção de referência com bloco padrão



Fonte: Elaborada pelo autor.

A proporção calculada é comparada com a proporção ideal do bloco padrão. Esse processo ocorre para cada quadrilátero até que o correto, aquele que apresenta proporção calculada dentro de uma faixa de valores de 1% em torno da proporção de referência, seja identificado.

Feita a identificação do bloco padrão, procede-se ao cálculo da largura, que é intuitivamente a distância entre as retas C1 e C2 (Figura 29). Utilizando novamente os quatros pontos de vértice obtidos do contorno, são identificadas as equações dessas duas retas. Como nunca serão retas perfeitamente paralelas, matematicamente a distância entre elas é zero, pois acabam se encontrando em algum ponto. Por isso, optou-se por calcular a largura através da média das distâncias entre pontos pertencentes a uma das retas e a outra reta.

Conhecendo as equações que descrevem as retas, escolhe-se uma para a qual são gerados diversos pontos a ela pertencentes e compreendidos dentro da faixa de valores entre os vértices usados para defini-la. A distância entre cada um desses pontos e a reta do lado oposto é calculada usando a equação:

$$d = \frac{|a \cdot X_0 + b \cdot Y_0 + c|}{\sqrt{(a^2 + b^2)}} \quad (4)$$

Onde:

- d – Distância;
- X_0, Y_0 – Coordenadas do ponto;
- a, b, c – Termos da equação da reta na forma: $ax+by+c=0$;

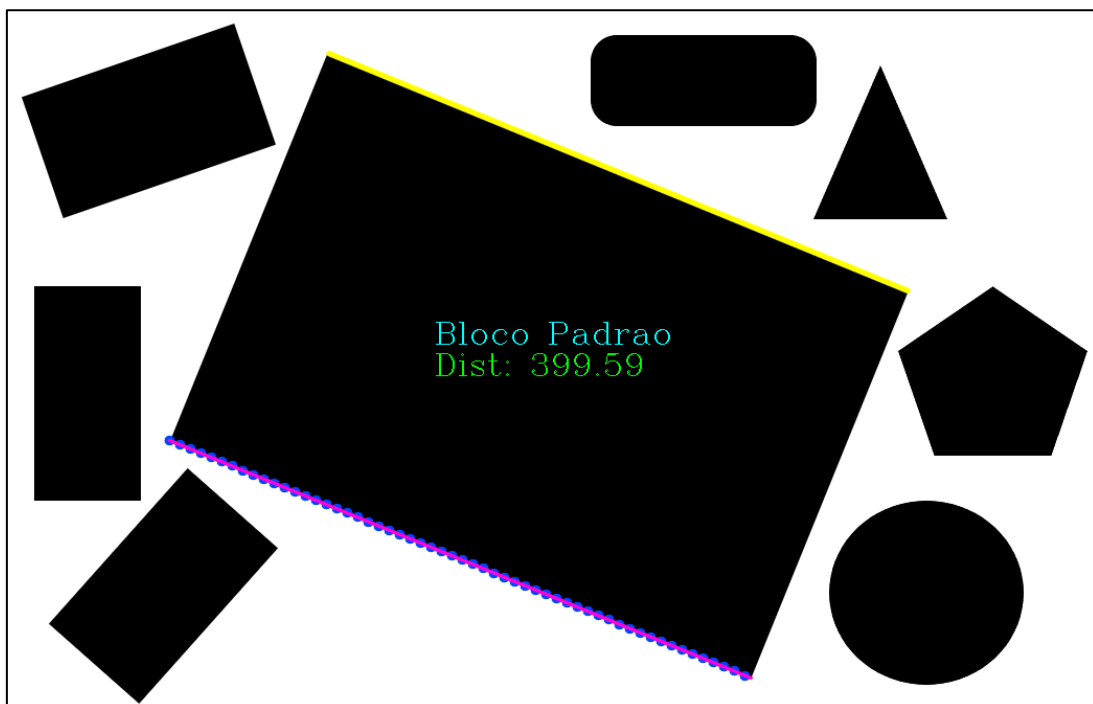
Calculadas as distâncias, o resultado final da largura é a média desses valores:

$$l = \frac{d_1 + d_2 + d_3 + \dots + d_n}{n} \quad (5)$$

Por fim, obtido o valor da largura em pixels do bloco padrão na imagem, e tendo o valor conhecido da medida gravado em uma variável, o programa calcula o fator de conversão utilizando a equação 3. Ao usar esse método, todo o processo ocorre automaticamente, bastando que o usuário pressione um botão na interface.

A Figura 30 mostra um teste do algoritmo feito com uma imagem gerada em computador, onde o retângulo que simula o bloco-padrão tem as dimensões 600x400 pixels e foram adicionados outros retângulos e formas dentre as quais o algoritmo seleciona a correta. Na imagem, as duas retas utilizadas como base do cálculo são indicadas em amarelo e rosa, os pontos gerados automaticamente dentro da reta rosa aparecem em azul e a distância da largura calculada com a média das distâncias entre os pontos e a reta amarela aparece no centro do bloco padrão com o valor de 399,59 px, muito próximo ao valor de referência de 400 px.

Figura 30: Teste do algoritmo de identificação do bloco padrão



Fonte: Elaborada pelo autor.

5.2 Fontes de erro investigadas

Para qualquer método de medição existem certas condições que favorecerão e outras que desfavorecerão a confiabilidade dos resultados. A fim de otimizar o processo de medição através de imagens, foram analisados os fatores com maior potencial de gerar erros e como evitar que eles ocorram ou, ao menos, minimizá-los.

5.2.1 Resolução insuficiente

Independentemente do método utilizado para medição com uso de imagens, alguma forma de referenciamento sempre será necessária. O método de referenciamento escolhido para esse projeto depende de uma referência que deve estar contida em cada imagem utilizada, referência na qual serão baseadas todas as medições. Devido a isso, uma das partes mais importantes do projeto, é a forma como se obtém essa referência. Nesta tarefa, o objetivo será obter um fator de conversão que relacione matematicamente uma quantidade de pixels a um valor de distância.

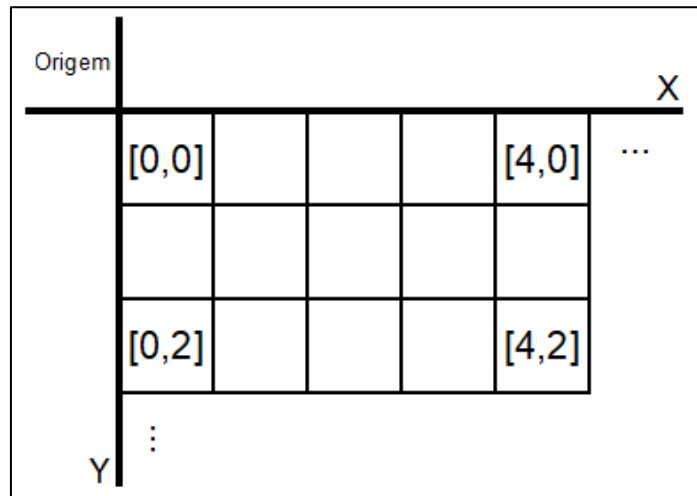
Antes de explicar os métodos de obtenção de referência desenvolvidos é preciso introduzir as premissas nas quais eles se baseiam.

Partindo do princípio de que uma fotografia digital é composta por linhas e colunas de pixels, dispostas na forma de dois eixos ortogonais, podemos dizer que ela é análoga ao conceito matemático de plano. Da mesma forma, pode-se relacionar os pixels à pontos contidos no plano da imagem. Contudo, diferente do ponto em seu conceito matemático, que é infinitesimal, não possuindo comprimento, área ou volume; o pixel possui altura e largura e, portanto, também uma área. Mesmo com essas diferenças, por ser a unidade mínima e indivisível que compõe a imagem digital, a analogia do pixel como ponto é válida e permite a aplicação das relações matemáticas da geometria plana no cálculo dimensional em imagens, sendo a base para o cumprimento dos objetivos do *software*.

Por convenção, a forma como se designa a posição dos pixels na imagem é feita da seguinte maneira: nas linhas a posição é incrementada da esquerda para a direita e nas colunas, de cima para baixo. O primeiro índice de cada linha e coluna sempre será o zero. Adequando a lógica de utilização da imagem como um plano XY à essa convenção, temos que a origem do plano fica no canto superior esquerdo da imagem, sendo ele delimitado em X por zero e o valor correspondente ao total de colunas na imagem, e em Y por zero e o número de linhas na imagem. A Figura 31, a seguir, ilustra essa

estrutura.

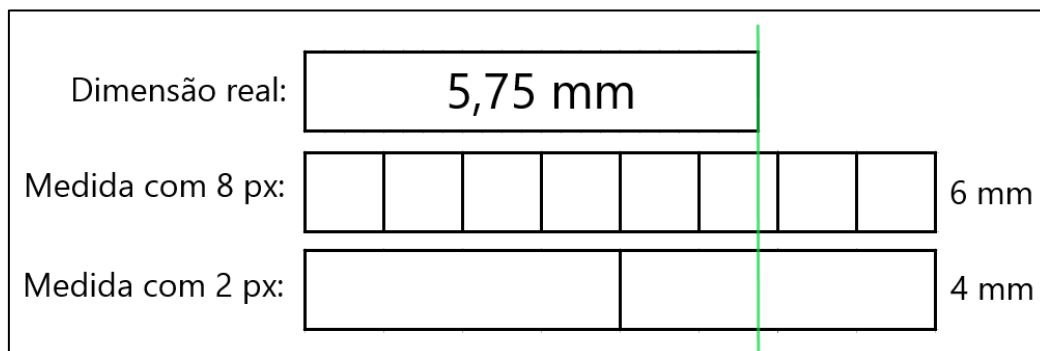
Figura 31: Pixels como pontos em um plano XY



Fonte: Elaborada pelo autor.

Pelo fato de o pixel não ser exatamente como um ponto, existem algumas limitações no seu uso para medição. Por ser indivisível e não infinitesimal, ocorre com ele o mesmo problema que existe em qualquer aplicação que envolva a conversão de sinais analógicos em digitais. O que ocorre é que não é possível contar ou registrar frações de pixels, portanto, em uma medição cujo valor real estaria em algum ponto no meio de um pixel ocorre perda de informação, com o resultado sendo representado por um valor correspondente a divisão entre pixels mais próxima. A Figura 32 foi produzida para ilustrar esse conceito.

Figura 32: Perda de informações na conversão para valores digitais



Fonte: Elaborada pelo autor.

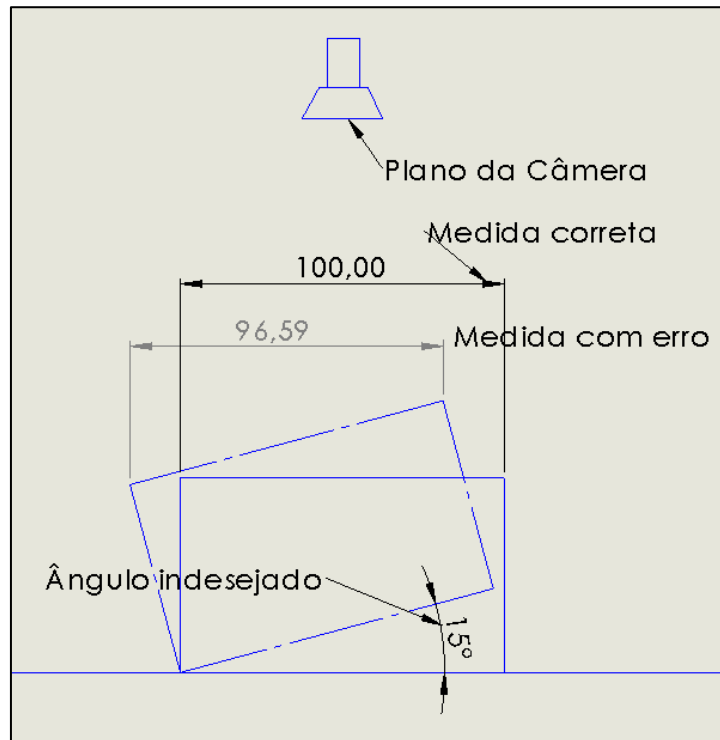
No exemplo, aparece um possível objeto com 5,75 mm de comprimento sendo comparado com duas representações de pixels com larguras distintas representando duas imagens de resoluções diferentes. A linha verde representando o ponto onde o valor real cortaria um pixel, caso possível. Na primeira imagem, de maior resolução, considerando uma relação pixel/mm de 1, o resultado da medição obtida seria de 6 mm. Já na segunda imagem, com uma resolução quatro vezes menor e relação

pixel/mm de apenas 0,25, seria obtido um resultado de 4 mm. Esses seriam os valores mais próximos que cada uma das imagens seria capaz de fornecer. Com esse exemplo conclui-se ainda que quanto maior a densidade de pixels representando um objeto em uma fotografia, maior tenderá a ser a fidelidade da medição desse objeto.

5.2.2 Paralelismo entre os planos da câmera e do objeto medido

Um das condições que pode ter impactos significativos na medição com imagens é o paralelismo entre o plano formado pela câmera, se imaginarmos o seu sensor como uma chapa que grava um corte de duas dimensões do que está à sua frente, e o plano do mensurando. Quando há um ângulo entre esses planos, a imagem resultante contém um erro inerente, o qual é proporcional a esse ângulo. A Figura 33, a seguir, ilustra como isso acontece.

Figura 33: Erro causado pela falta de paralelismo



Fonte: Elaborado pelo autor.

A imagem exibe uma peça com um comprimento de 100 unidades. Quando a peça está com o plano da dimensão alvo paralelo ao plano da câmera, a medida percebida pelo sensor da câmera representa corretamente a medida real do objeto; caso haja um ângulo – no exemplo, de 15° – qualquer entre esses planos, o corte em duas dimensões capturado pela câmera conterá uma versão enganosa dessa dimensão. No caso do exemplo, como o valor do ângulo indesejado é conhecido, é possível computar a dimensão incorreta que seria obtida através da seguinte expressão:

$$Medida = Valor\ real \times \cos \alpha$$

$$100,00 \times \cos 15^\circ = 96,59\ unidades$$

Em casos em que seja inevitável trabalhar com ângulo entre os planos, e esse ângulo seja conhecido, não haverá problemas, uma vez que o resultado final pode ser calculado levando-o em consideração. O problema desse tipo de erro é que ele surge naturalmente quando se tira qualquer fotografia, sendo mais acentuado quando a foto é tirada sem o devido cuidado. No entanto, para ângulos muito pequenos esse erro torna-se desprezível, principalmente se comparado aos causados por outras fontes de erro que serão citadas.

5.2.3 Peça e a referência em distâncias diferentes da câmera

As dimensões aparentes de um objeto presente em uma foto variam com a distância em que se encontram da câmera. Se dois objetos de mesmas dimensões forem posicionados a diferentes distâncias da câmera, suas dimensões aparentes variarão consideravelmente. Por isso, para que se obtenham os melhores resultados, o ideal é que a face que possui a dimensão de referência do bloco-padrão (ou outro corpo de referência) esteja no mesmo plano que a face alvo do corpo que será medido. Caso contrário, o fator de conversão calculado com base nas dimensões da referência não corresponderá de forma precisa às dimensões reais, como mostra a Figura 34.

Figura 34: Erro visível causado por diferença de distância entre a câmera e o objeto.

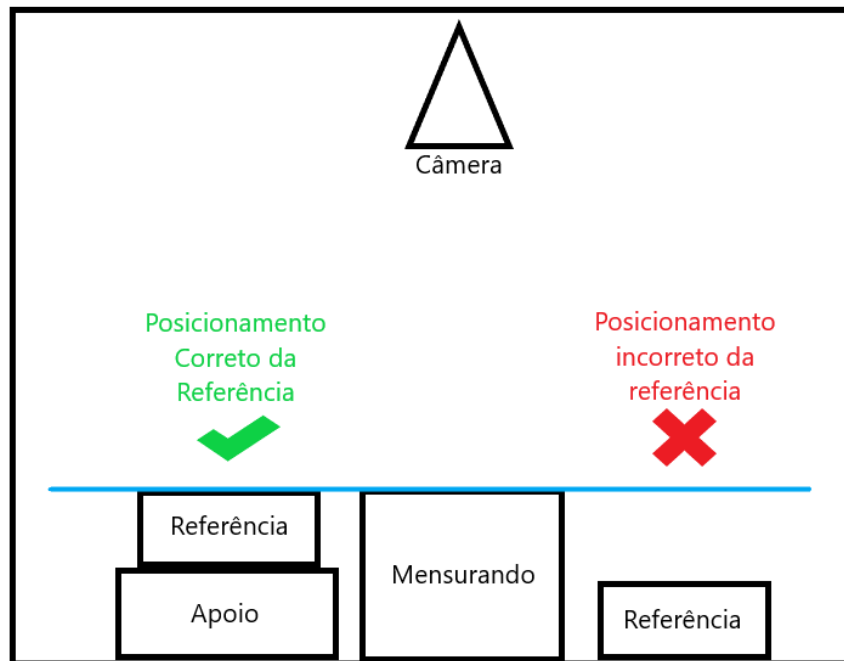


Fonte: Elaborada pelo autor.

A imagem da Figura 34 contém uma comparação entre a vista frontal do desenho técnico e uma foto real de uma das peças utilizadas nos testes de validação do projeto e ilustra muito claramente o quão significativo pode ser o erro causado pela não coplanaridade entre as faces da referência e do mensurando. Como pode ser observado na vista frontal, o ressalto tem a mesma largura do corpo principal da peça. Já na foto (sinalizado em vermelho) parece menor do que é; isso ocorre justamente por ele estar mais distante da câmera.

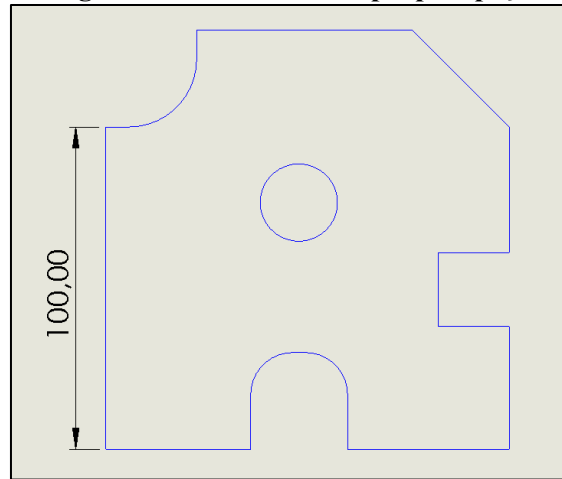
É interessante observar que a distância entre a face principal e a face do ressalto - ou seja, a diferença da distância entre os planos e a câmera - mostradas é de apenas cerca de 10 mm, o que indica que mesmo desalinhamentos muito pequenos entre os planos do objeto de referência e do mensurando podem resultar num erro de medição relevante. Portanto, é necessário que se tome um cuidado especial com o posicionamento do corpo de referência em relação à superfície da peça que se deseja medir. Sendo que, idealmente, essas superfícies devem pertencer ao mesmo plano geométrico.

Figura 35 - Alinhamento entre a referência dimensional e o mensurando



Fonte: Elaborada pelo autor.

Para evitar essa dificuldade, também é possível tomar como referência a medida de uma dimensão conhecida no próprio corpo do mensurando e que esteja no mesmo plano das dimensões que se deseja obter. Por exemplo, supondo que se deseje medir todas as dimensões de uma chapa com o formato apresentado na Figura 36. Uma possibilidade seria utilizar um instrumento de medição convencional para obter a medida de uma das arestas, conforme exemplificado na imagem, e usá-la como referência para que se pudesse obter todas as outras utilizando o software de medição.

Figura 36: Referência na própria peça

Fonte: Elaborada pelo autor.

Outra possibilidade de solução para este problema é o emprego de lentes telecêntricas, que possuem a capacidade de manter a relação pixel/mm numa determinada faixa de distâncias da câmera para a peça. No entanto, devido a seu alto custo, esta solução foi descartada neste trabalho.

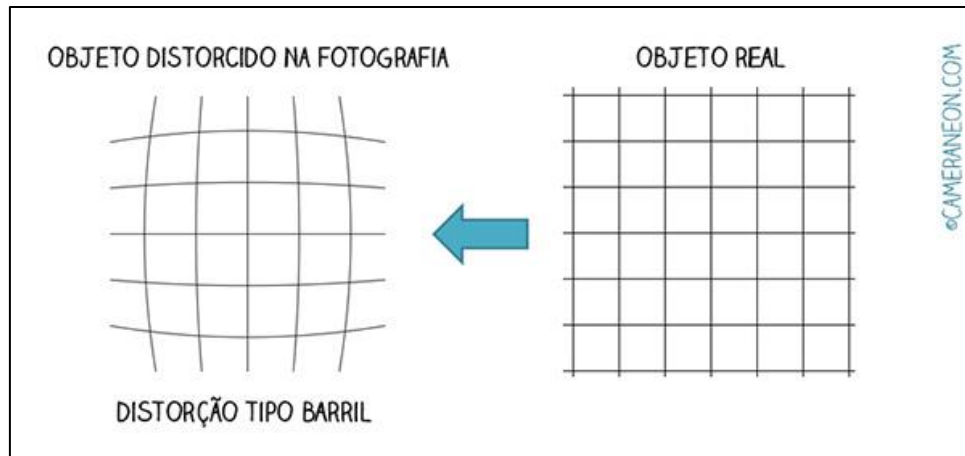
5.2.4 Distorções

Outro ponto que merece atenção ao se utilizar imagens na medição é relacionado às distorções que surgem ao longo da imagem, geralmente causados por efeitos intencionais ou imperfeições nas lentes. Essas distorções são chamadas de distorções radiais e resultam em uma representação falha da cena fotografada, tendo o efeito de magnificações irregulares ao longo da imagem, justamente o oposto do desejado para a medição. Assim como a falta de paralelismo câmera/mensurando e o desalinhamento referência/peça, também é impossível eliminar absolutamente os erros causados por distorções, pois mesmo as lentes fabricadas para evitá-las ao máximo não são perfeitas nessa tarefa.

Genericamente, as distorções radiais são classificadas em três tipos: Barril, Almofada e Complexa.

Na distorção do tipo barril ocorre uma magnificação variável partindo do centro da cena e diminuindo em direção às bordas da imagem:

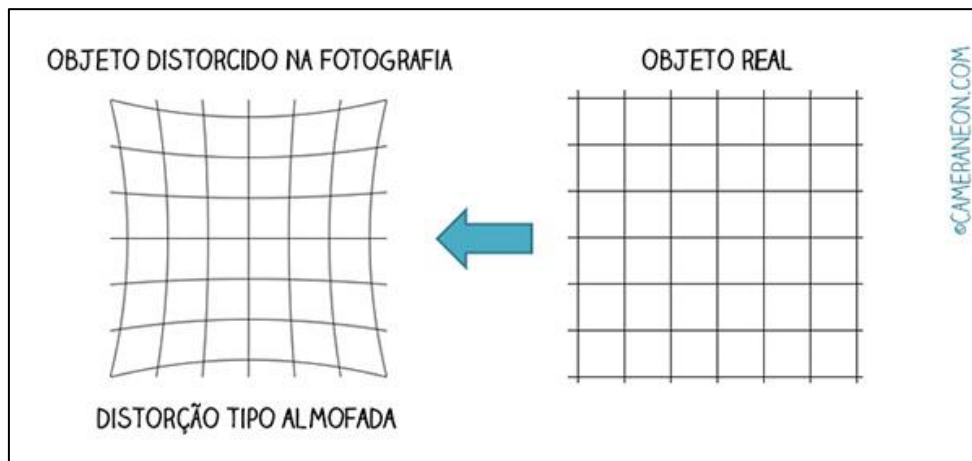
Figura 37: Distorção do tipo barril



Fonte: cameraneon.com, 2020.

Na distorção do tipo almofada, ocorre o oposto, com as bordas magnificadas em relação ao centro:

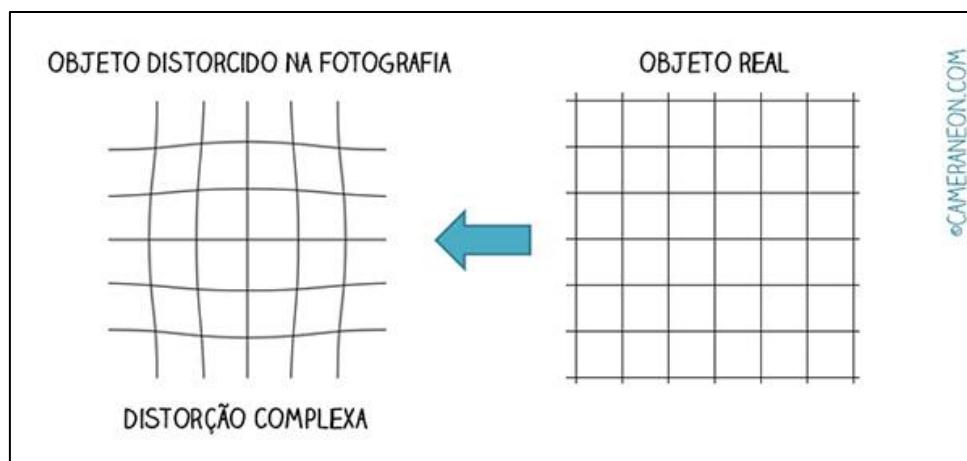
Figura 38: Distorção do tipo almofada



Fonte: cameraneon.com, 2020.

Já a distorção do tipo complexa envolve os dois tipos de irregularidades, com o centro da imagem apresentando distorção do tipo barril e as bordas do tipo almofada.

Figura 39: Distorção do tipo complexa



Fonte: cameraneon.com, 2020.

O grande problema desse tipo de erro é que ele influencia diretamente no fator de conversão entre a medida em pixels e em unidade de comprimento. Pelo fato de o fator de conversão ser calculado com base em uma medida de referência contida na imagem, o erro causado pelas distorções ataca direta e justamente essa que é a base da qual dependem todas as medições. Ainda mais, esse erro é cumulativo, pois numa imagem distorcida tanto a referência quanto cada medida feita sofrem influências de não linearidades.

Para evitar ao máximo esse tipo de erro, é ideal o uso de objetivas ‘normais’, que são conjuntos de lentes que buscam reproduzir o que seriam as condições normais da visão humana (considerando apenas um olho). Mesmo utilizando lentes apropriadas, ainda existe uma distância focal ótima em que serão obtidos os melhores resultados.

Na prática, com o uso de fotografias obtidas através de câmera de celular, ou câmeras de computador, o melhor que se pode fazer é a calibração da câmera com uso de configurações manuais. No caso de câmeras fixadas em um dispositivo mecânico de boa precisão, um parâmetro que pode ser controlado pelo usuário é a distância entre a câmera e o objeto alvo, tendo sido observado que ocorrem maiores distorções em fotos tiradas há distâncias muito curtas, causadas pelos ajustes automáticos de foco e pós processamento do aparelho.

5.3 Processamentos matemáticos implementados

Conforme mencionado no tópico dedicado à obtenção da referência dimensional, consideramos para fins práticos os pixels da imagem como pontos de dimensão unitária em um plano, de forma que a maioria das relações matemáticas válidas para tratar pontos num plano, são também válidas para trabalhar com imagens. No presente tópico serão apresentados os fundamentos matemáticos utilizados no desenvolvimento do projeto para a determinação das dimensões planejadas.

5.3.1 Distância entre pontos:

A equação utilizada para calcular a distância entre dois pixels é exatamente a mesma que se usa para distância entre dois pontos. Portanto, a distância entre os pixels A e B é calculada por:

$$d = \sqrt{(X_b - X_a)^2 + (Y_b - Y_a)^2} \quad (6)$$

Onde:

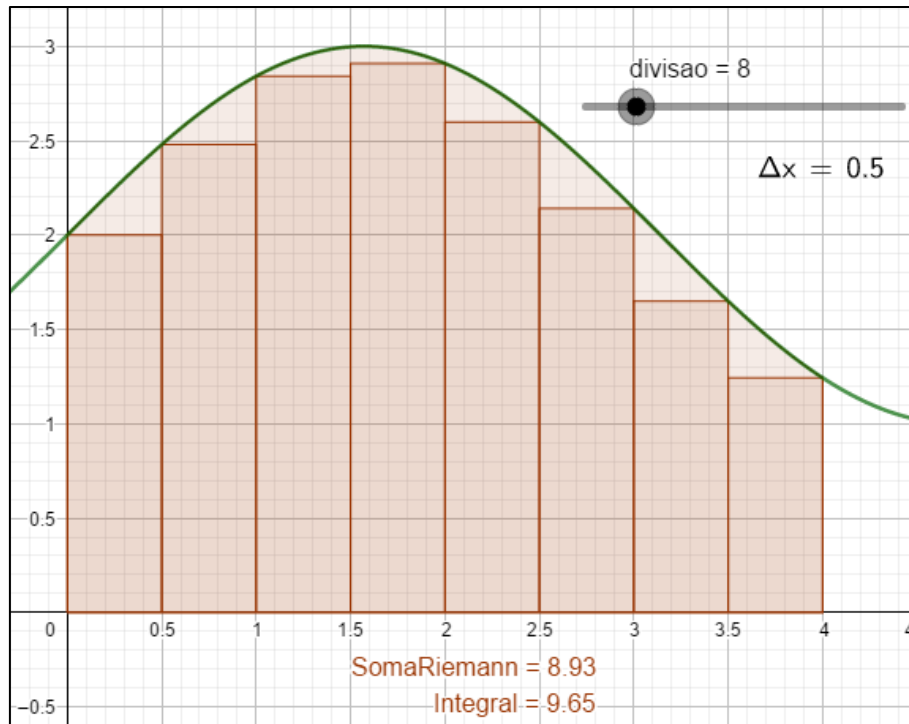
- X_a, Y_a – Coordenadas do pixel A;
- X_b, Y_b – Coordenadas do pixel B;

5.3.2 Cálculo de áreas pela contagem de pixels

Para explicar o método que será utilizado para o cálculo de área de figuras quaisquer, podemos tomar como base o conceito das integrais. Uma de suas definições, é que a integral de uma função determina a área sob a curva da mesma no plano cartesiano, isso é comprovado intuitivamente através da soma de Riemann (George B. Jr.; Finney, Ross L. (1996)). A soma de Riemann pode ser representada como a soma das áreas de retângulos de base Δx e altura tocando a curva $f(x)$ entre um intervalo A e B, onde quanto menor o Δx , mais aproximada fica a soma da área sob a curva nesse intervalo. Uma integral é uma soma de Riemann onde Δx tende a zero, portanto o número de retângulos abaixo da curva tende a infinito, através do que se obtém a melhor aproximação possível da área.

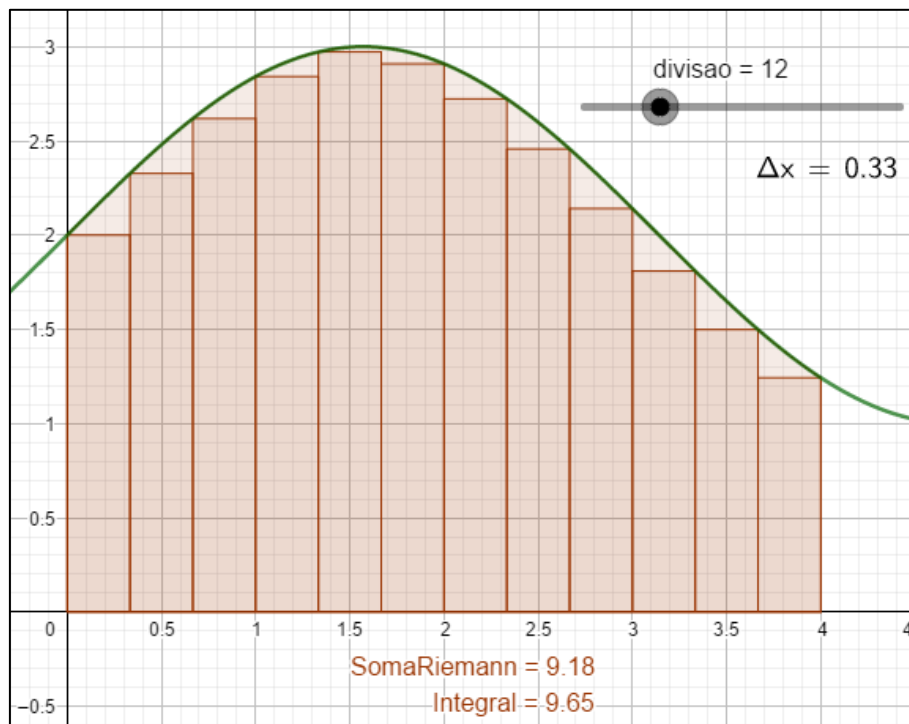
Para ilustrar o conceito da soma de Riemann, utilizando o *software* Geogebra, tomamos como exemplo a função $f(x) = 2 + \text{sen}(x)$. Apresentamos abaixo da curva a integral da função no intervalo de 0 a 4 e a soma de Riemann, no mesmo intervalo. Para fins de comparação, são apresentadas a seguir imagens exibindo essas representações para três variações de Δx na soma de Riemann.

Figura 40: Soma de Riemann sob a curva $f(x)=2+\text{sen}(x)$ no intervalo 0 a 4 com $\Delta x=0,5$



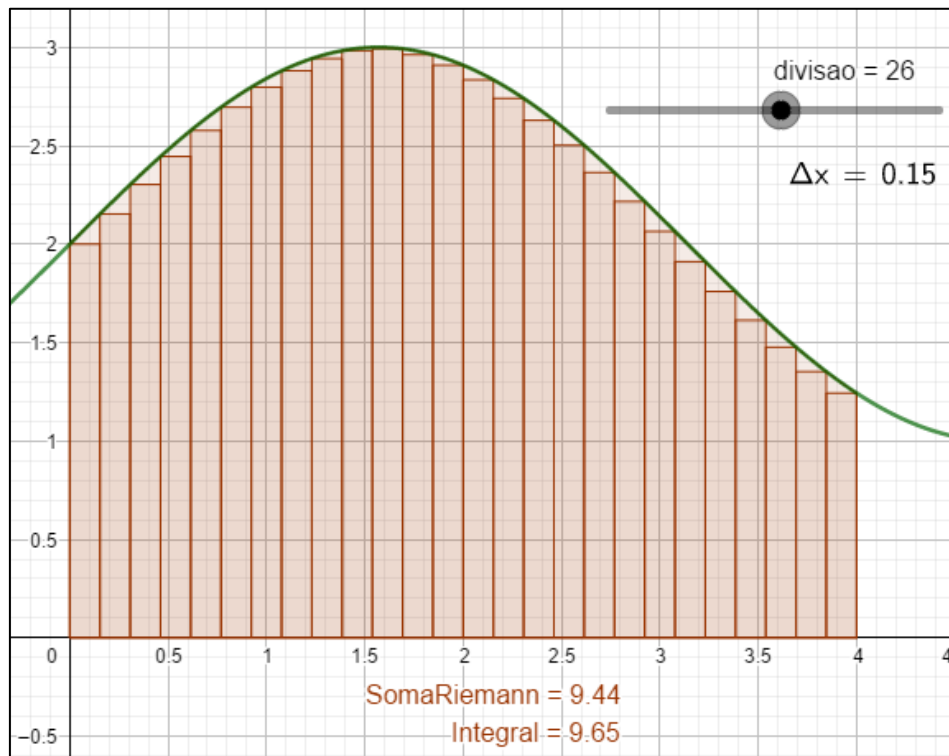
Fonte: Elaborada pelo autor.

Figura 41: Soma de Riemann sob a curva $f(x)=2+\text{sen}(x)$ no intervalo 0 a 4 com $\Delta x=0,33$



Fonte: Elaborada pelo autor.

Figura 42: Soma de Riemann sob a curva $f(x)=2+\text{sen}(x)$ no intervalo 0 a 4 com $\Delta x=0,15$



Fonte: Elaborada pelo autor.

Os gráficos deixam claras as afirmações citadas anteriormente. Fica visível que quanto menor o Δx , mais parecida a soma dos retângulos fica em relação a integral da função. Isso é também comprovado numericamente, pois observa-se que o valor da soma de Riemann vai se aproximando de valor da integral conforme Δx é diminuído.

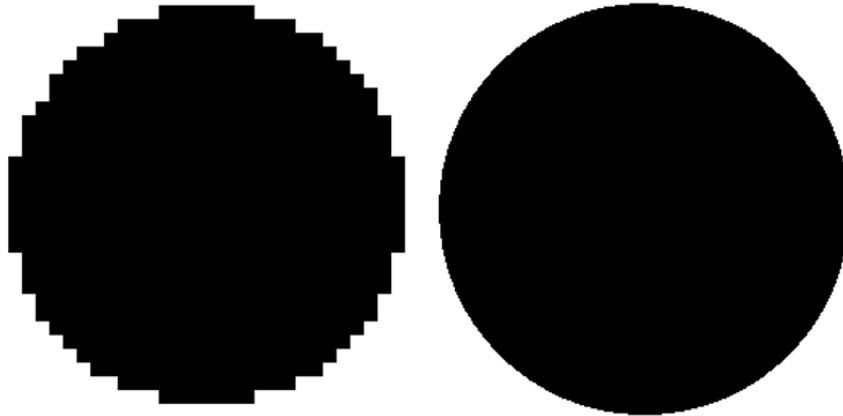
Esse raciocínio permite estabelecer uma comparação entre o conceito mencionado de integral como área sob a curva de uma função e o cálculo de áreas em imagens. Como já mencionado, o pixel pode ser considerado um quadrado unitário e, portanto, tal como na soma de Riemann, pode-se dizer que áreas de contornos de formas quaisquer em uma imagem digital são a soma das áreas de todos os pixels internos a esse contorno.

Outra implicação natural é que, tal como menores valores de Δx tornam a área da soma de Riemann mais próxima do valor real da área abaixo da curva. Caso duas imagens do mesmo objeto sejam usadas para calcular a área do mesmo por contagem de pixels, aquela que possuir maior resolução, ou seja, maior densidade de pixels, deve ser a que apresentará o resultado mais próximo do real.

Para ilustrar esse conceito, foi criada uma imagem (Figura 43) com dois círculos simbolizando

o que seria o mesmo círculo representado com densidades de pixels diferentes. O primeiro foi criado com a resolução de 25x25 pixels e o segundo com 250x250 pixels. É fácil perceber que o segundo círculo é uma representação muito mais fiel do que seria um círculo real e seria muito melhor usar esse para calcular a área do círculo.

Figura 43: Círculo representando com dimensões 25x25 e 250x250 pixels



Fonte: Elaborada pelo autor.

5.3.3 Construção de retas a partir de dois pontos

A determinação da equação de retas é utilizada em duas importantes funcionalidades do programa, na obtenção automática da referência com bloco padrão e no cálculo de circunferências geradas por três pontos.

A equação geral da reta tem a seguinte forma:

$$ax + by + c = 0 \quad (7)$$

O método utilizado nesse projeto para obtenção da equação geral da reta foi desenvolvido a partir do método do determinante. Para que pontos P_1 e P_2 pertençam a mesma reta, o resultado da determinante abaixo deve ser zero.

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x & y & 1 \end{vmatrix} = 0 \quad (8)$$

Desenvolvendo essa determinante de maneira genérica, pode-se obter relações que facilitam a implementação do algoritmo. Assim:

$$x_1y_2 + xy_1 + yx_2 - xy_2 - yx_1 - x_2y_1 = 0$$

$$x(y_1 - y_2) + y(x_2 - x_1) + x_1y_2 - x_2y_1 = ax + by + c = 0$$

Portanto:

$$a = y_1 - y_2 \quad (9)$$

$$b = x_2 - x_1 \quad (10)$$

$$c = x_1y_2 - x_2y_1 \quad (11)$$

Com essas relações, obtém-se a equação geral da reta diretamente em função das coordenadas de dois pontos.

5.3.4 Determinação de ponto médio de um segmento de reta

A determinação do ponto médio de um segmento de reta delimitado por dois pontos é utilizada no algoritmo de cálculo de centro e raio de circunferências.

Dados os pontos P_1 e P_2 pertencentes a mesma reta, delimitando um segmento desta reta, as coordenadas do ponto médio P_m entre P_1 e P_2 são determinadas por:

$$(x_m, y_m) = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right) \quad (12)$$

5.3.5 Determinação de distância entre um ponto e uma reta

Dados o ponto $P_1(x_1, y_1)$ e a reta $r: ax+by+c=0$, a menor distância entre P_1 e r é determinada por:

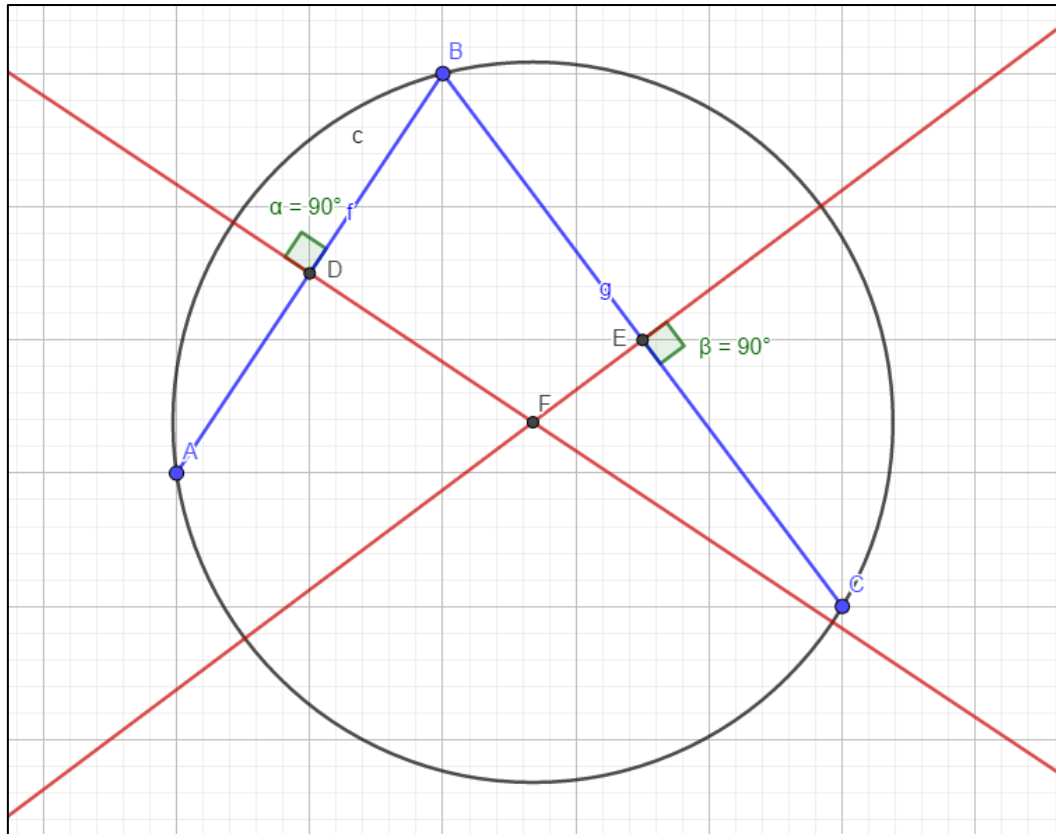
$$d = \frac{|ax_1 + by_1 + c|}{\sqrt{a^2 + b^2}} \quad (13)$$

Esse cálculo é utilizando no algoritmo de obtenção automática da referência com bloco padrão.

5.3.6 Construção geométrica de circunferências utilizando três pontos

O método implementado para obtenção de circunferências é baseado em uma regra que permite a obtenção do centro de uma circunferência utilizando retas formadas por três pontos pertencentes a ela. O software Geogebra foi utilizado para criar a ilustração da Figura 44 a fim de facilitar a compreensão desse método.

Figura 44: Circunferência formada a partir de três pontos



Fonte: Elaborada pelo autor.

Dados os pontos A, B e C pertencentes à circunferência mostrada, são calculados os pontos médios D e E entre os segmentos de reta AB e BC, respectivamente. Em seguida, cria-se uma reta perpendicular ao segmento AB e passando pelo ponto D e uma reta perpendicular ao segmento BC, passando pelo ponto E. Dadas essas condições, o centro da circunferência coincide com o ponto de intersecção das duas retas resultantes, na imagem, em vermelho. Portanto, no exemplo, o centro da circunferência está no ponto F.

Para isso, a primeira operação utilizada é a do cálculo do ponto médio entre A e B, utilizando a equação 12, em seguida é necessário encontrar o coeficiente angular do segmento AB, para isso utiliza-se a expressão:

$$m_{ab} = \frac{y_b - y_a}{x_b - x_a} \quad (14)$$

Após a determinação deste parâmetro, encontra-se o coeficiente angular de uma reta r qualquer perpendicular ao segmento AB:

$$m_r = \frac{-1}{m_{ab}} \quad (15)$$

Na sequência, é preciso especificar qual das retas com coeficiente angular m_r deve ser utilizada, o coeficiente angular é utilizado em conjunto com o ponto D para encontrar essa reta:

$$y - y_d = m_r(x - x_d)$$

$$y = m_r x - m_r x_d + y_d$$

$$-m_r x + y + m_r x_d - y_d = ax + by + c = 0$$

Portanto:

$$a = -m_r \quad (16)$$

$$b = 1 \quad (17)$$

$$c = m_r x_d - y_d \quad (18)$$

Os mesmos cálculos são feitos para o segmento BC com o ponto E, resultando em outra reta. Por fim, encontra-se o ponto de intersecção entre as duas retas obtidas através de um sistema de equações envolvendo as equações que as representam:

$$\begin{cases} ax + by + c = 0 \\ dx + ey + f = 0 \end{cases} \quad (19)$$

Onde:

$dx+ey+f=0$ - Equação da reta perpendicular ao segmento BC passando pelo ponto E;

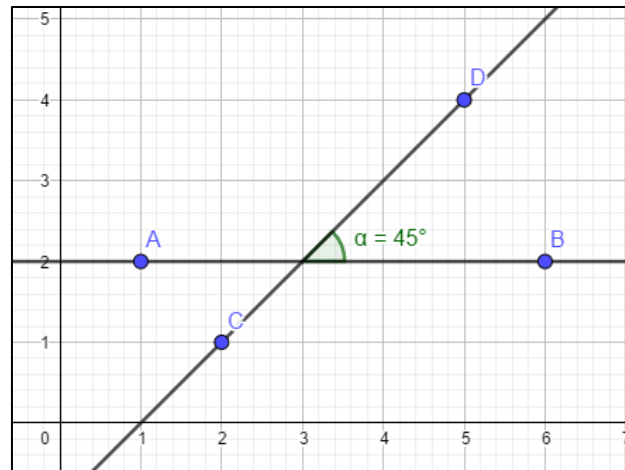
x,y – coordenadas do ponto de intersecção das retas;

Esse método permite a obtenção do centro de qualquer circunferência definida por três pontos, sendo que o para definir totalmente a circunferência após a identificação do centro, é calculada a distância entre esse centro e qualquer desses pontos. No programa, optou-se por calcular o raio usando a média das distâncias entre o centro e cada um dos três pontos.

5.3.7 Determinação do ângulo entre duas retas formadas por quatro pontos

Para implementar a funcionalidade de cálculo de ângulos no programa foi utilizada uma única equação que utiliza as coordenadas de quatro pontos para calcular o ângulo entre as duas retas formadas por esses pontos. Utilizando novamente o Geogebra para gerar uma ilustração (Figura 45):

Figura 45: Ângulo obtido entre as retas formadas a partir de quatro pontos



Fonte: Elaborada pelo autor.

Dados os pontos A, B, C e D, com os quais deseja-se obter o ângulo α entre as retas formadas por AB e CD, α é dado por:

$$\alpha = \cos^{-1} \left(\frac{(x_b - x_a) \times (x_d - x_c) + (y_b - y_a) \times (y_d - y_c)}{\sqrt{(x_b - x_a)^2 + (y_b - y_a)^2} \times \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}} \right) \quad (20)$$

A ângulo resultante é dado em radianos, então é feita a conversão para graus caso desejado.

5.3.8 Conversão de comprimento em pixel para unidade de comprimento real

Para converter o valor de medidas de comprimento que estão em pixels para a unidade definida pelo usuário é usada a seguinte equação:

$$d_R = \frac{d_{px}}{F_c} \quad (21)$$

Onde:

- d_R – distância real em unidade de comprimento
- d_{px} – distância medida em pixels
- F_c – fator de conversão

5.3.9 Conversão de área em pixel para unidade de área real

Para converter o valor da medida de área em pixels para área na unidade definida pelo usuário é usada a seguinte equação:

$$A_R = \frac{A_{px}}{F_c^2} \quad (22)$$

Onde:

- A_R – distância real em unidade de comprimento
- A_{px} – distância medida em pixels
- F_c – fator de conversão

5.4 Implementação das funcionalidades

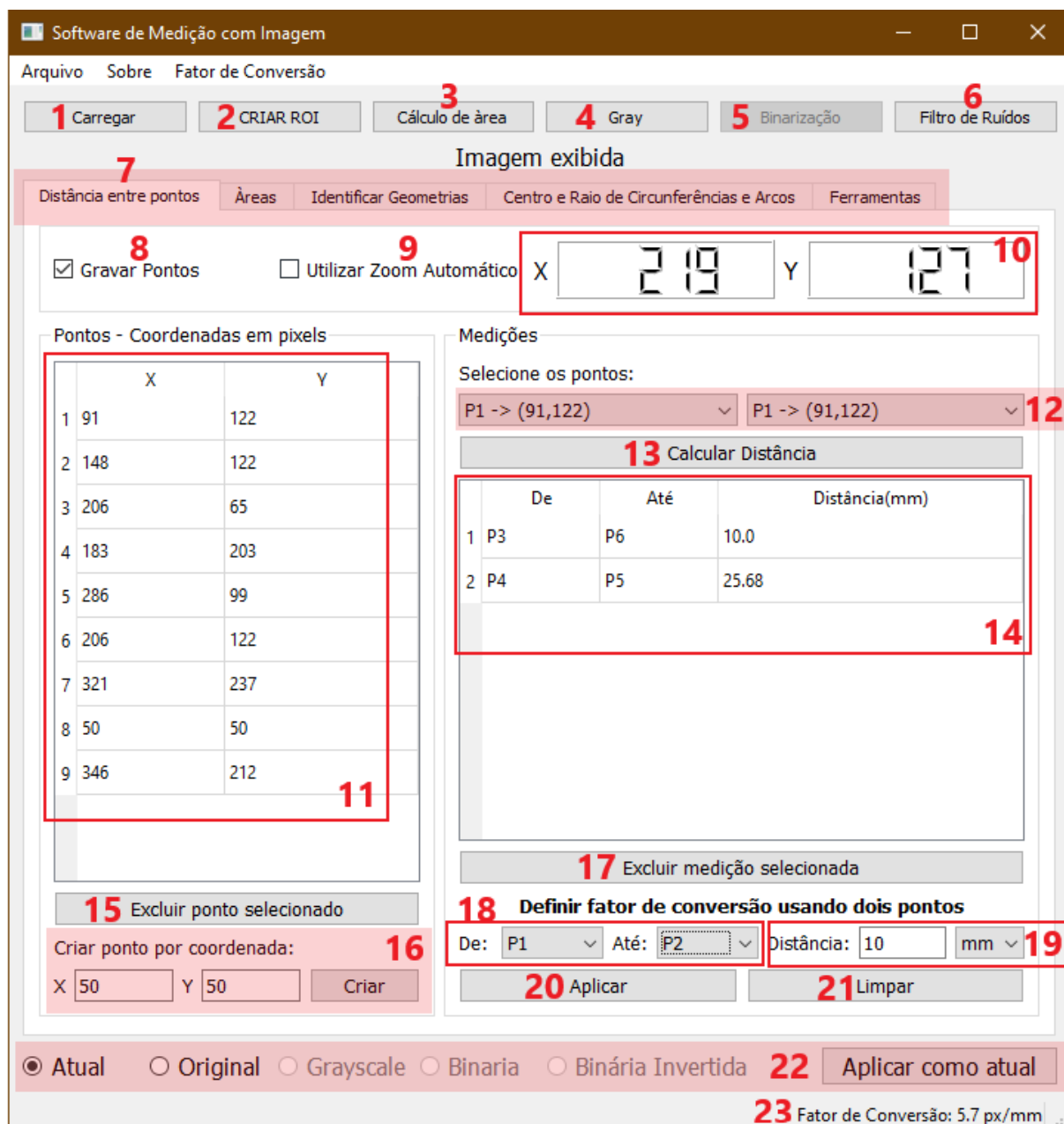
Os cálculos explicados na seção anterior foram implementados no software de medição, conforme será explicado nesta seção. Os resultados de todas as medições são arredondados para duas casas decimais antes de serem exibidos na interface.

5.4.1 Visão geral da interface gráfica

A interface gráfica foi criada de forma a incluir as funcionalidades pretendidas de forma intuitiva. Os componentes da interface serão mostrados nas figuras a seguir, cada uma seguida de uma tabela com sua lista de itens.

Aba 1 – Criação de pontos, medição de distâncias e definição do fator de conversão:

Figura 46: Interface do programa com a primeira aba selecionada



Fonte: Elaborada pelo autor.

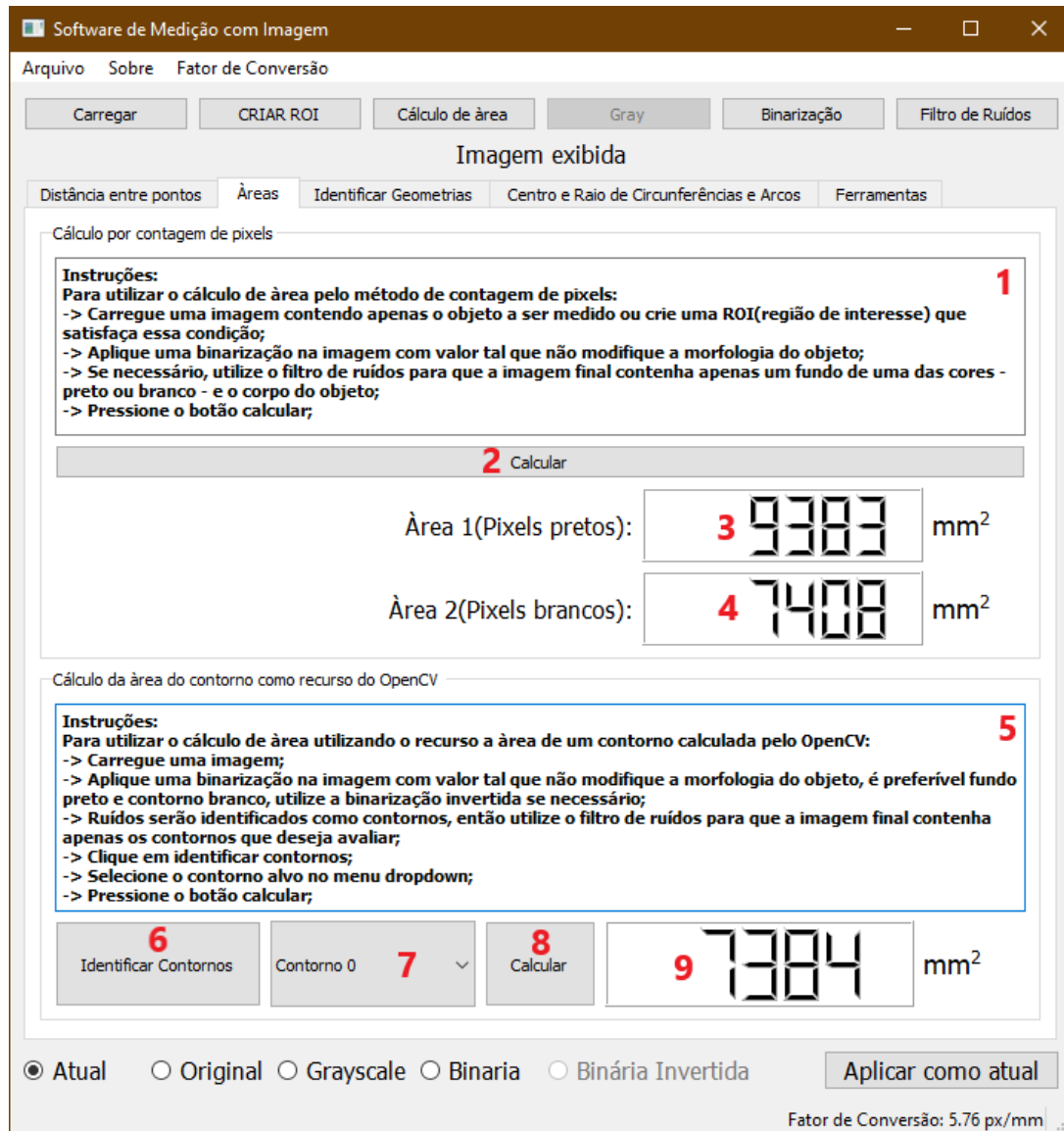
Tabela 2: Componentes da aba 1

Item	Descrição
1	Acionamento da busca e abertura de imagens do computador
2	Seleção da ferramenta de criação de regiões de interesse
3	Atalho para as funcionalidades de cálculo de área
4	Atalho para conversão da imagem para escala de cinza
5	Atalho para função de binarização
6	Atalho para a ferramenta de filtragem de ruídos
7	Seletor de abas
8	Checkbox de habilitação da criação de pontos
9	Checkbox de habilitação da ferramenta de zoom automático

10	Exibição das coordenadas atuais do ponteiro do mouse
11	Tabela de pontos criados
12	Botões combo box para seleção dos pontos para medição de distância
13	Botão para efetuar e registrar a medição da distância entre os dois pontos selecionados
14	Tabela de medições criadas
15	Botão para deletar ponto selecionado na tabela
16	Ferramenta de criação manual de pontos com a entrada das coordenadas
17	Botão para deletar medição selecionada na tabela
18	Seleção dos pontos referentes a medida de referência
19	Entradas do valor da medida de referência e da unidade
20	Botão para calcular o fator de conversão com base na referência informada
21	Limpar seleções do menu de referenciamento
22	Navegação entre as versões da imagem mantidas em memória
23	Fator de conversão atual

Aba 2 – Ferramentas para cálculo de áreas:

Figura 47: Interface do programa com a segunda aba selecionada



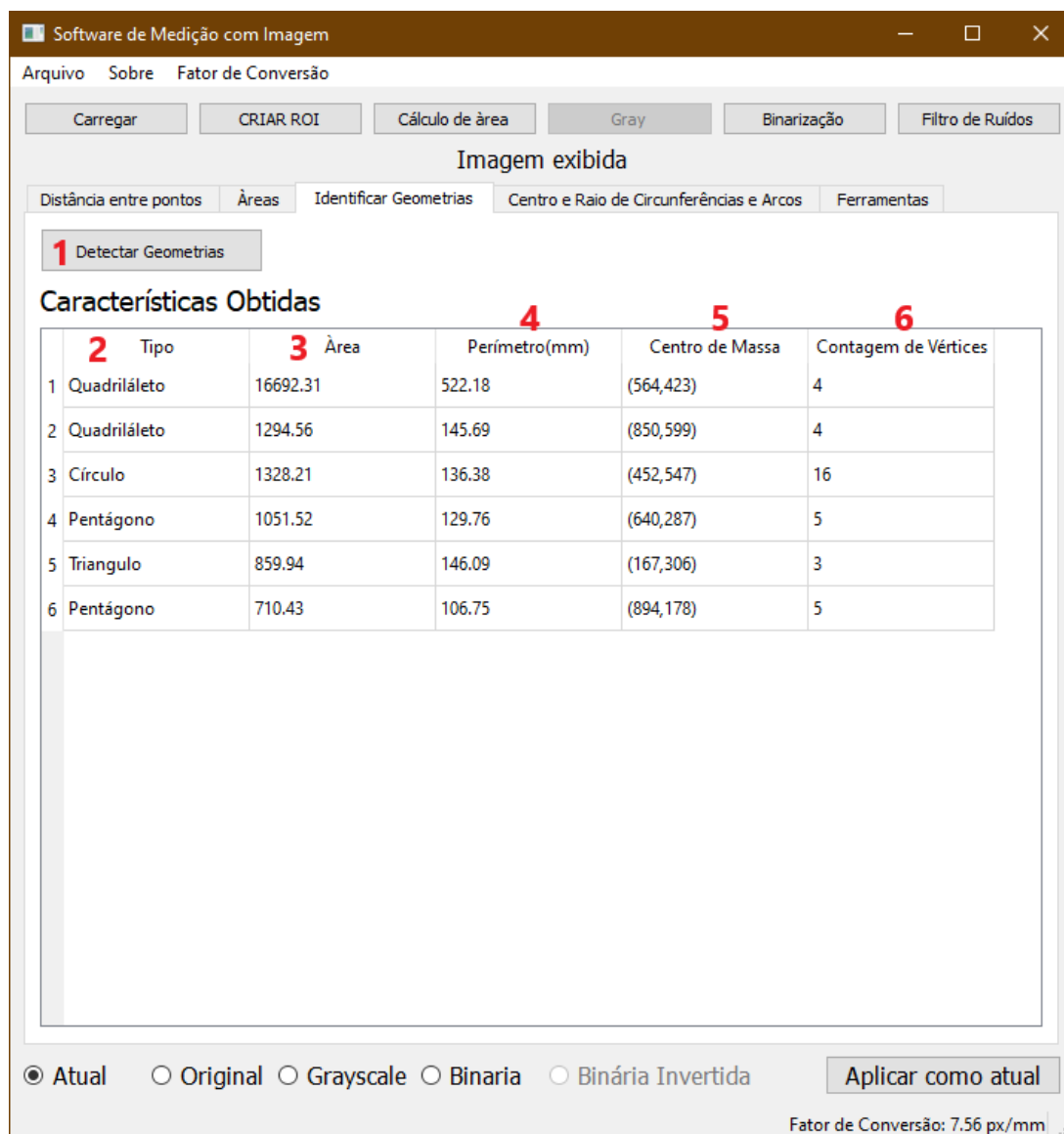
Fonte: Elaborada pelo autor

Tabela 3: Componentes da aba 2

Item	Descrição
1	Instruções para medição de área por contagem de pixels
2	Botão para realização da medição
3	Resultado da medição da área correspondente aos pixels pretos
4	Resultado da medição da área correspondente aos pixels brancos
5	Instruções para medição de área pelo recurso nativo do OpenCV
6	Botão para realizar a identificação dos contornos
7	'Combo box' que lista e permite a seleção do contorno a ser medido
8	Botão para realização da medição do contorno selecionado
9	Resultado da medição da área do contorno selecionado

Aba 3 – Ferramenta de detecção de formas geométricas:

Figura 48: Interface do programa com a terceira aba selecionada



Fonte: Elaborada pelo autor.

Tabela 4: Componentes da aba 3

Item	Descrição
1	Acionamento da detecção de geometrias
2	Coluna exibindo o tipo da forma geométrica detectada
3	Coluna exibindo a área calculada da forma geométrica detectada
4	Coluna exibindo o perímetro calculado da forma geométrica detectada
5	Coluna exibindo o centro de massa da forma geométrica detectada
6	Coluna exibindo o número de vértices detectados no contorno

Aba 4 – Ferramenta de medição de circunferências e arcos:

Figura 49: Interface do programa com a quarta aba selecionada



Fonte: Elaborada pelo autor.

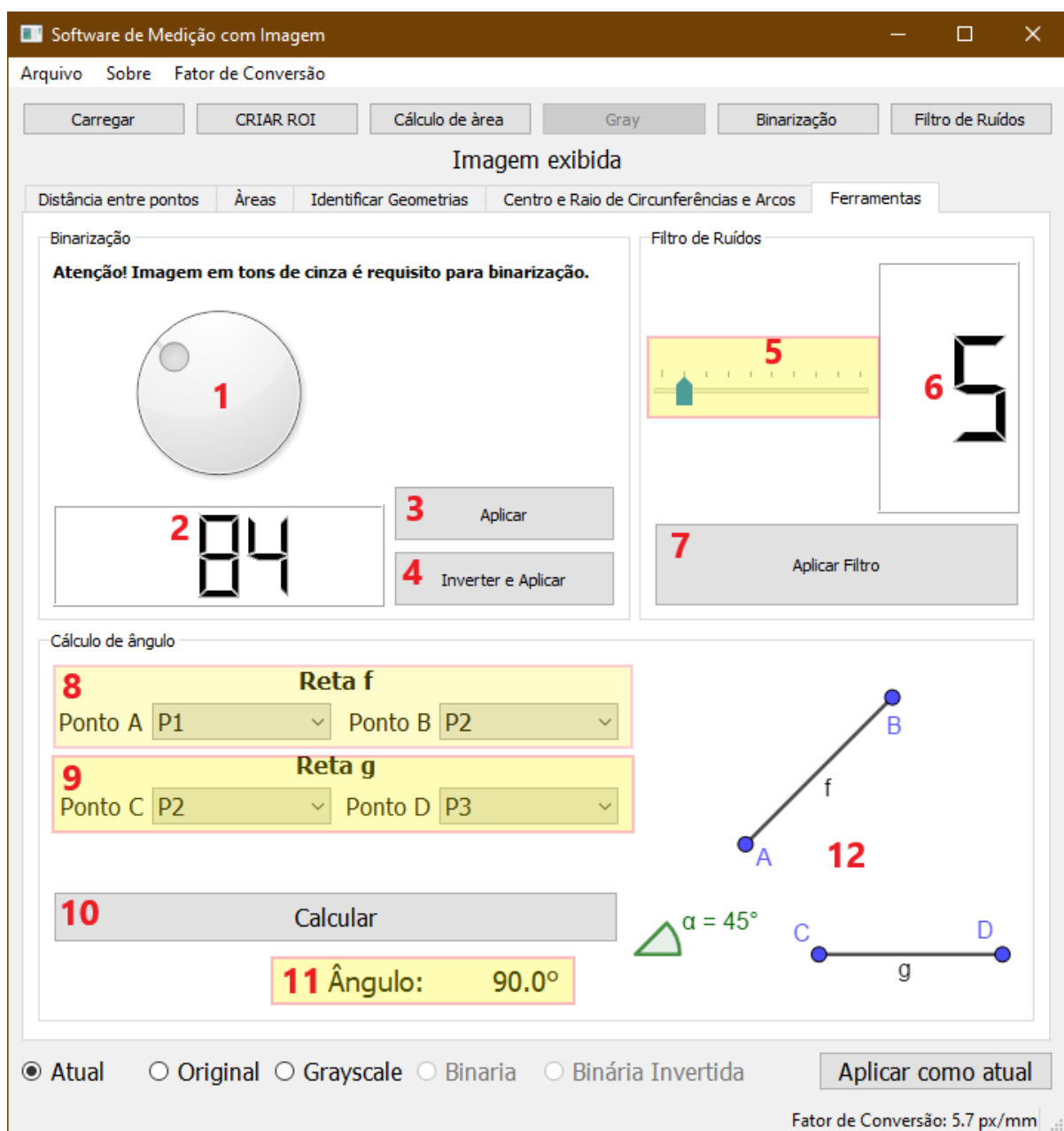
Tabela 5: Componentes da aba 4

Item	Descrição
1	Habilitar marcação da circunferência na imagem (por definitivo)
2	Habilitar marcação da circunferência na imagem (temporariamente)
3	Indicação das coordenadas dos pontos selecionados manualmente
4	Botões combo box para seleção dos pontos para medição
5	Habilitar marcação temporária dos pontos na imagem
6	Executar medição baseada nos três pontos selecionados
7	Indicação do centro da circunferência por seleção manual de pontos
8	Indicação do raio medido por seleção manual de pontos
9	Indicação do diâmetro medido por seleção manual de pontos

10	Acionar algoritmo de detecção de círculos
11	Seleção do círculo alvo dentre os presentes na imagem
12	Executar medição com seleção automática dos pontos na circunferência
13	Habilitar exibir na imagem os pontos selecionados automaticamente
14	Indicação do centro da circunferência por seleção automática dos pontos
15	Indicação do raio medido por seleção automática dos pontos
16	Indicação do diâmetro medido por seleção automática dos pontos

Aba 5 – Ferramentas de binarização, filtragem de ruídos e cálculo de ângulos

Figura 50: Interface do programa com a quinta aba selecionada



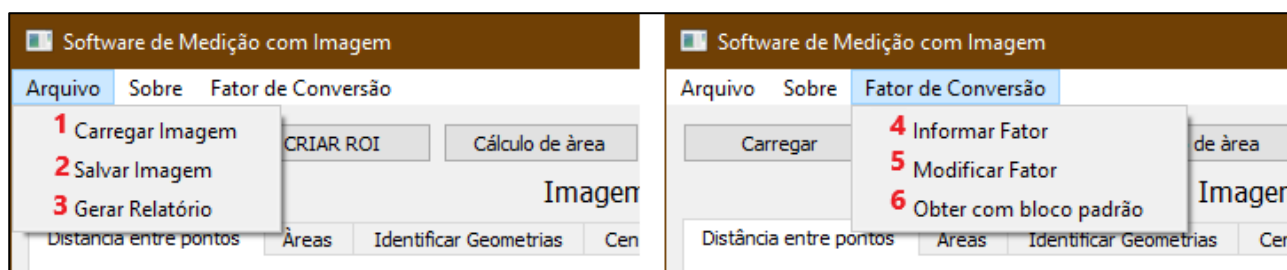
Fonte: Elaborada pelo autor.

Tabela 6: Componentes da aba 5

Item	Descrição
1	Seletor do limiar para a binarização
2	Indicação do valor de limiar
3	Aplicar binarização com limiar selecionado
4	Aplicar binarização invertida com limiar selecionado
5	Seletor de tamanho do kernel
6	Indicador do tamanho de kernel selecionado
7	Aplicar filtro com valor de kernel selecionado
8	Seleção dos pontos da primeira reta
9	Seleção dos pontos da segunda reta
10	Aciona algoritmo de cálculo do ângulo entre as retas
11	Indicação do resultado do cálculo do ângulo entre as retas
12	Ilustração de modelo para cálculo do ângulo entre as retas

Opções da barra de menu:

Figura 51: Opções da barra de menu



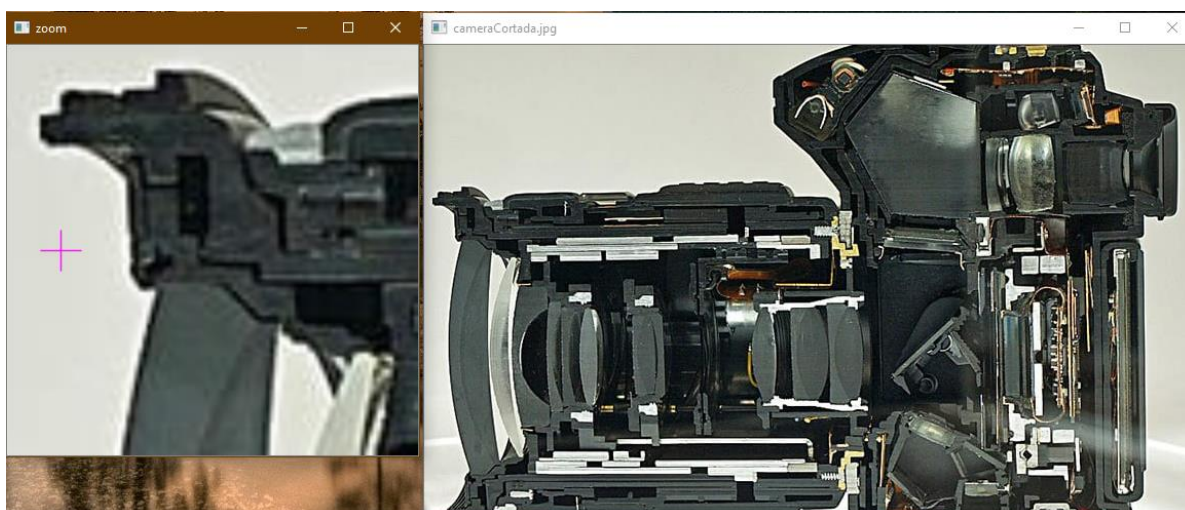
Fonte: Elaborada pelo autor.

Tabela 7: Itens da barra de menu

Item	Descrição
1	Acionamento da busca e abertura de imagens do computador
2	Salvar imagem em seu estado atual
3	Geraria um relatório das medições feitas em um arquivo PDF (não implementado)
4	Conduz para a seleção de pontos e entrada do valor de referência na aba 1
5	Modificação do fator de conversão por entrada do usuário (não implementado)
6	Aciona algoritmo de obtenção automática do fator de conversão com bloco padrão

O programa também abre uma nova janela ao carregar a imagem selecionada. Quando habilitada a ferramenta de zoom automático, é aberta ainda outra janela para mostrar a região com zoom.

Figura 52: Amostra do funcionamento da ferramenta de zoom automático



Fonte: Elaborada pelo autor.

5.4.2 Criação de pontos

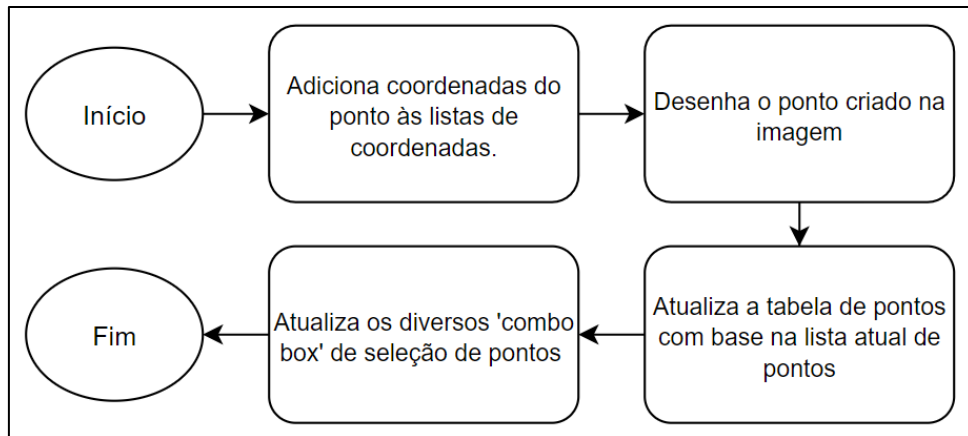
O programa já inicia com duas listas nas quais podem ser gravadas as coordenadas dos pontos em X e em Y, respectivamente. A criação de pontos consiste em acrescentar o valor da posição em X do pixel alvo em uma das listas e a posição em Y na outra lista. Foi feito dessa forma porque o componente 'combo box' do PyQt5 retorna o índice da posição selecionada, então as coordenadas do ponto podem ser facilmente lidas das listas usando esse índice. Isso também facilita o preenchimento das tabelas.

Como as listas do Python iniciam no índice 0, o índice referente a cada ponto sempre está defasado em 1 do nome do ponto, ou seja, o índice referente ao ponto P_n é $n-1$.

a) Selecionando na imagem

Uma vez habilitada a criação de pontos (Figura 46, item 8), o programa passa a rastrear a posição do mouse sobre a janela onde está aberta a imagem através de uma função acionada pelo tratamento de eventos do mouse do OpenCV, que informa o tipo de evento ocorrido e a posição do cursor no instante em que esse evento ocorreu. Quando a função identifica o evento referente a soltura do botão esquerdo do mouse, ela aciona o seguinte algoritmo:

Figura 53: Fluxograma de criação de ponto por seleção de pixels

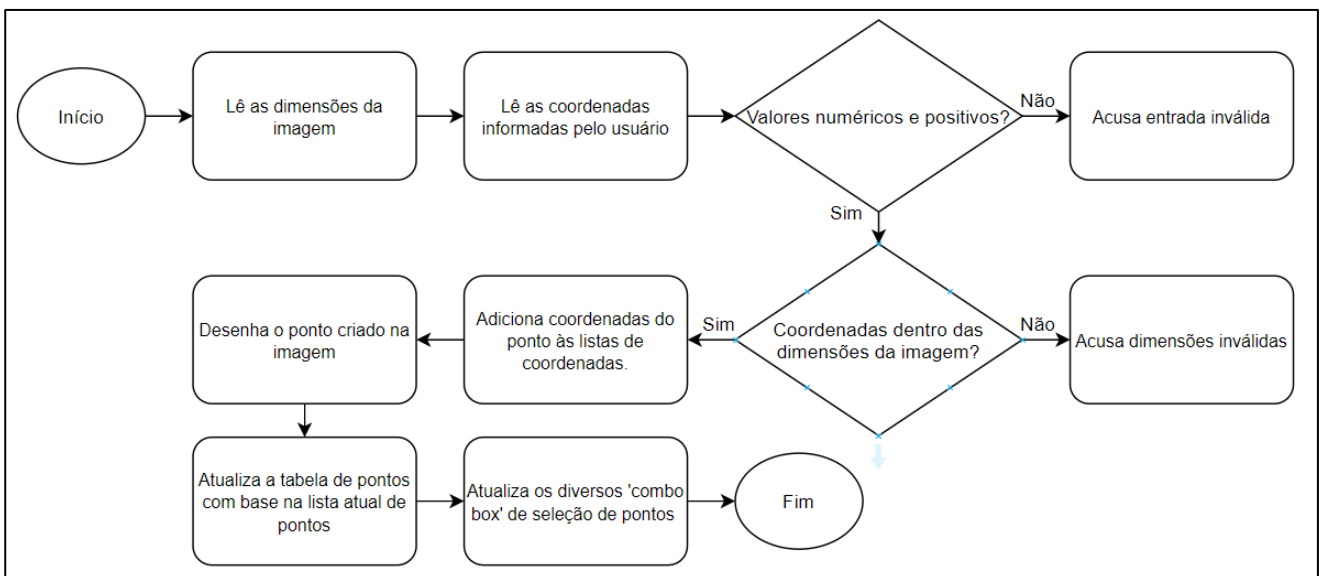


Fonte: Elaborada pelo autor.

b) Criação com coordenadas informadas pelo usuário

O usuário também pode optar por criar pontos informando as coordenadas desejadas ao programa. Para isso ele deve usar o menu identificado no item 16 da Figura 46. A função que cria um ponto através desse método opera da seguinte maneira:

Figura 54: Fluxograma de criação de ponto com entrada manual das coordenadas



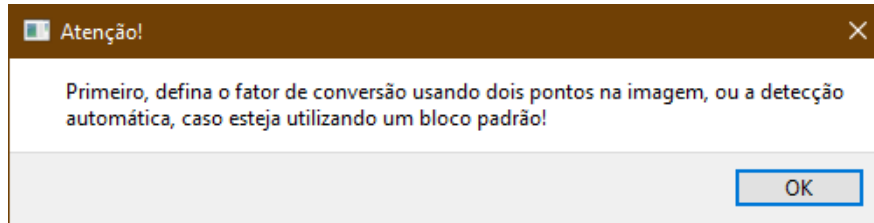
Fonte: Elaborada pelo autor.

Os pontos criados são listados na tabela de pontos (Figura 46, item 11) e podem ser excluídos através do botão que aparece abaixo da tabela (Figura 46, item 15). Ao serem excluídos, uma cópia da imagem original é recuperada, nela são desenhados apenas os pontos restantes e a imagem é atualizada na tela, de forma que a exclusão do ponto também o elimina da imagem em tela.

5.4.3 Obtenção da Referência Dimensional

A operacionalização da obtenção da referência dimensional inicia assim que uma imagem é carregada no programa, sendo o usuário alertado para que crie a referência dimensional como primeiro passo.

Figura 55: Caixa de diálogo alertando sobre a atribuição da referência



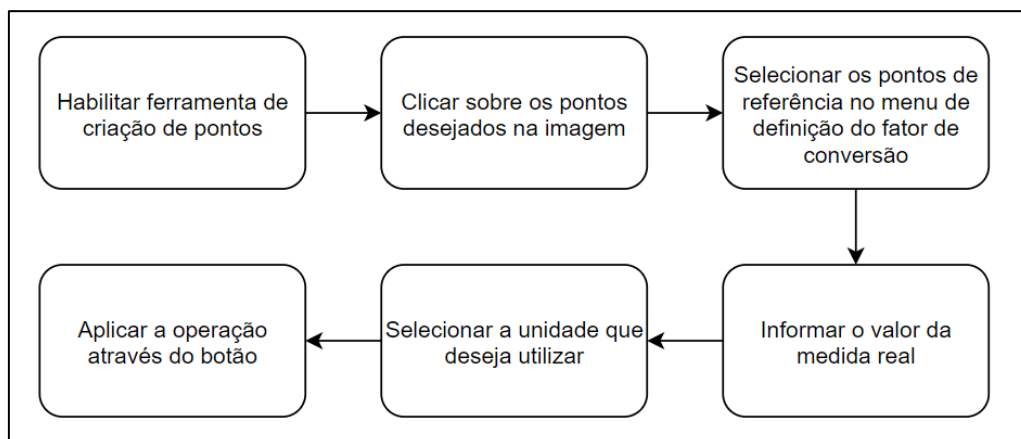
Fonte: Elaborada pelo autor.

Diante dessa necessidade, o usuário tem duas opções, dependendo do método escolhido ao criar a imagem.

a) Utilizando uma distância conhecida

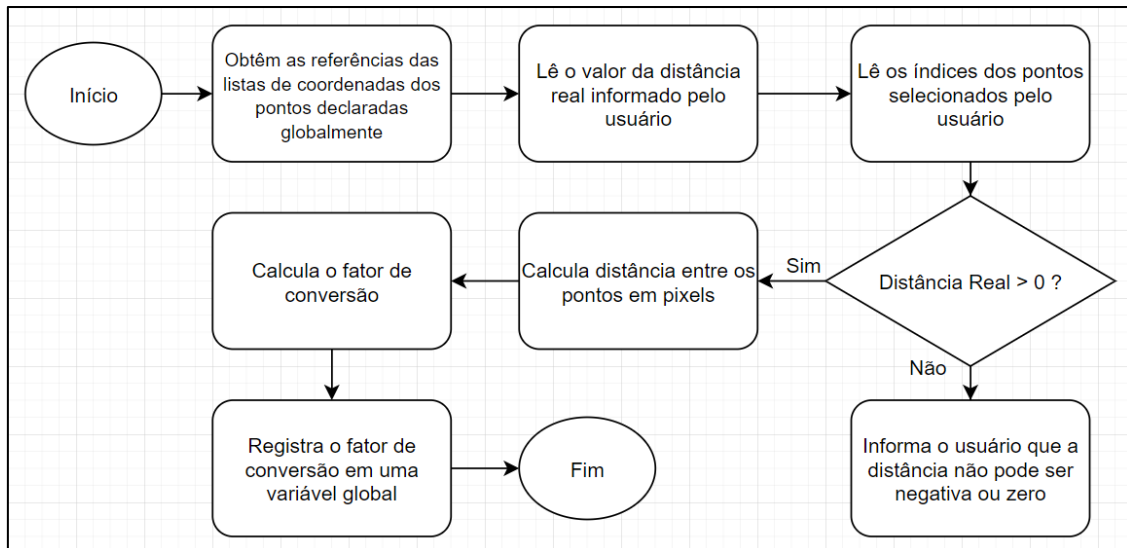
O primeiro método de aquisição da referência é baseado em uma distância entre pontos que o usuário deve selecionar na imagem. Após carregada a imagem, o usuário deve proceder conforme o fluxograma abaixo:

Figura 56: Passos para obtenção do fator de referência através de uma distância conhecida



Fonte: Elaborada pelo autor.

O algoritmo da função utilizada para essa operação funciona conforme Figura 57:

Figura 57: Função 'definirFatorDeConversao'

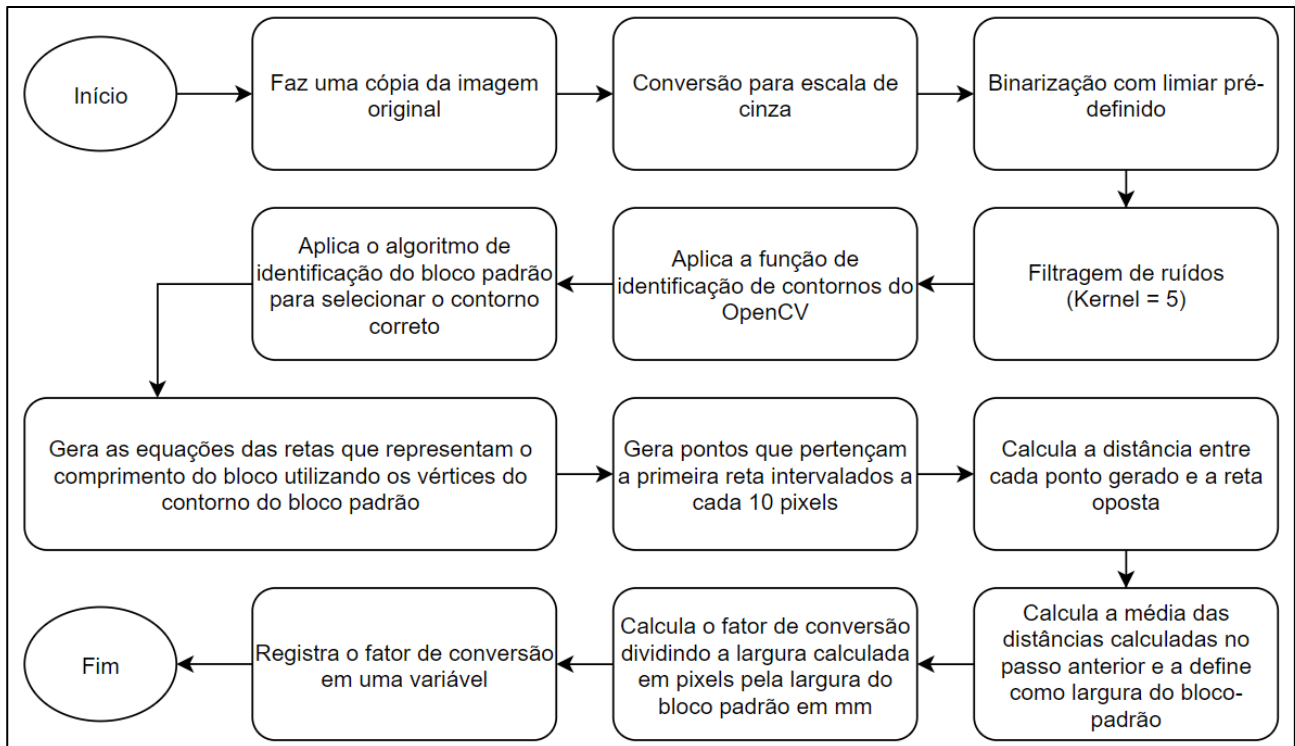
Fonte: Elaborada pelo autor.

O resultado é o valor numérico para o fator de conversão gravado em uma variável global do programa e exibido na interface, conforme exemplificado na marcação 23 da Figura 46.

b) Usando um bloco padrão

Para utilizar a funcionalidade obtenção automática da referência através de um bloco-padrão, após carregar a imagem alvo, o usuário deve clicar na opção 'Obter com bloco padrão' no menu 'Fator de conversão' na barra de menu (Figura 51, item 6).

O algoritmo base do código implementado no programa para execução desse método de referenciamento segue conforme o fluxograma da Figura 58 e é executado automaticamente, sem que o usuário precise selecionar qualquer opção adicional.

Figura 58: Função 'obterFatorAuto'

Fonte: Elaborada pelo autor.

O algoritmo de identificação do bloco-padrão funciona conforme explicado no item 5.1.2. O algoritmo que gera pontos pertencentes a primeira reta não será detalhado; basta mencionar que ele gera pontos da reta no intervalo compreendido entre os pontos de vértice do bloco-padrão, suas coordenadas X são incrementadas de 10 em 10 pixels e as coordenadas Y calculadas conforme a equação da reta. A distância entre os pontos e a reta oposta é calculada com a equação 13, a largura do bloco-padrão é determinada pela média desses valores (equação 5) e o fator de conversão conforme a equação 3.

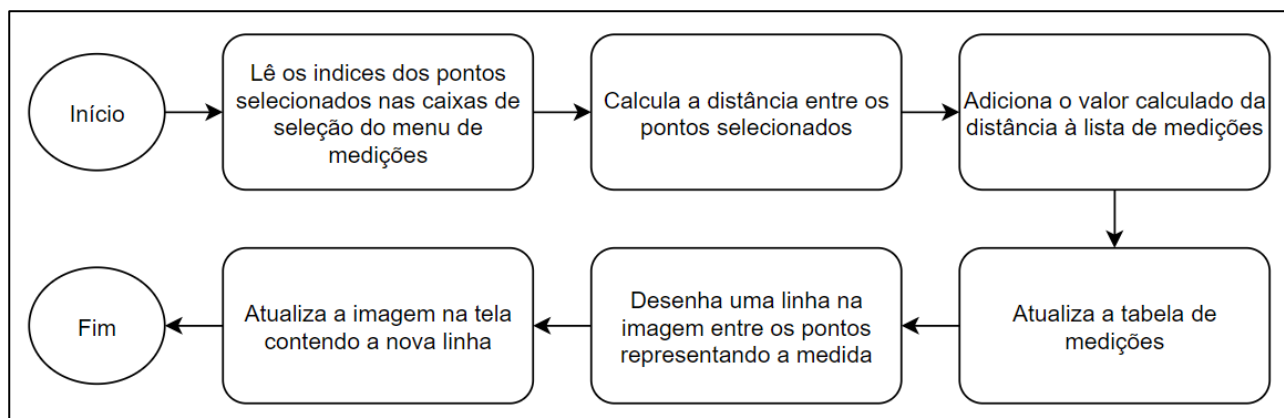
5.4.4 Medição da distância entre Pontos

A medição de distância é possível a partir do momento em que há um fator de conversão especificado. Os valores das medições ficam registrados em uma lista (Python List) multidimensional que armazena também as coordenadas dos dois pontos que a originaram. Para realizar a medição, o usuário deve:

- Habilitar a gravação de pontos (Figura 46, item 8);
- Criar pontos clicando sobre pixels na imagem;
- Selecionar os pontos correspondentes a medição desejada nas caixas de seleção do menu de medição (Figura 46, item 12);
- Clicar no botão 'Calcular Distância' no menu de medições (Figura 46, item 13);

O botão aciona a função 'realizarMedicao', cujo algoritmo é descrito a seguir:

Figura 59: Função 'realizarMedicao'



Fonte: Elaborada pelo autor.

A função que calcula a distância entre os pontos faz a leitura das coordenadas dos pontos baseada nos índices selecionados pelo usuário a executa a operação através da equação 6.

A tabela (Figura 46, item 14) onde as medições são listadas exibe o ponto de origem (coluna 1), o ponto final (coluna 2) e a distância entre eles (coluna 3).

Da mesma forma como os pontos, as medições também podem ser excluídas, através de um botão que fica embaixo da tabela (Figura 46, item 17). Ao excluir uma medição, a imagem também é atualizada, recuperando a imagem original e redesenhando apenas as linhas das medições ainda existentes.

5.4.5 Identificação de formas geométricas simples e medição de suas características

A identificação de formas serve para localizar objetos na imagem, tentando detectar sua forma dentre algumas opções (triângulo, quadrilátero, pentágono, hexágono e círculo) e listando outras informações (área, perímetro, centro de massa e número de vértices). O recurso funciona baseado no número de vértices dos contornos presentes na imagem analisada, por isso é um recurso simples e feito para funcionar bem quando as formas são bem definidas. Para utilizá-lo, o usuário deve primeiro:

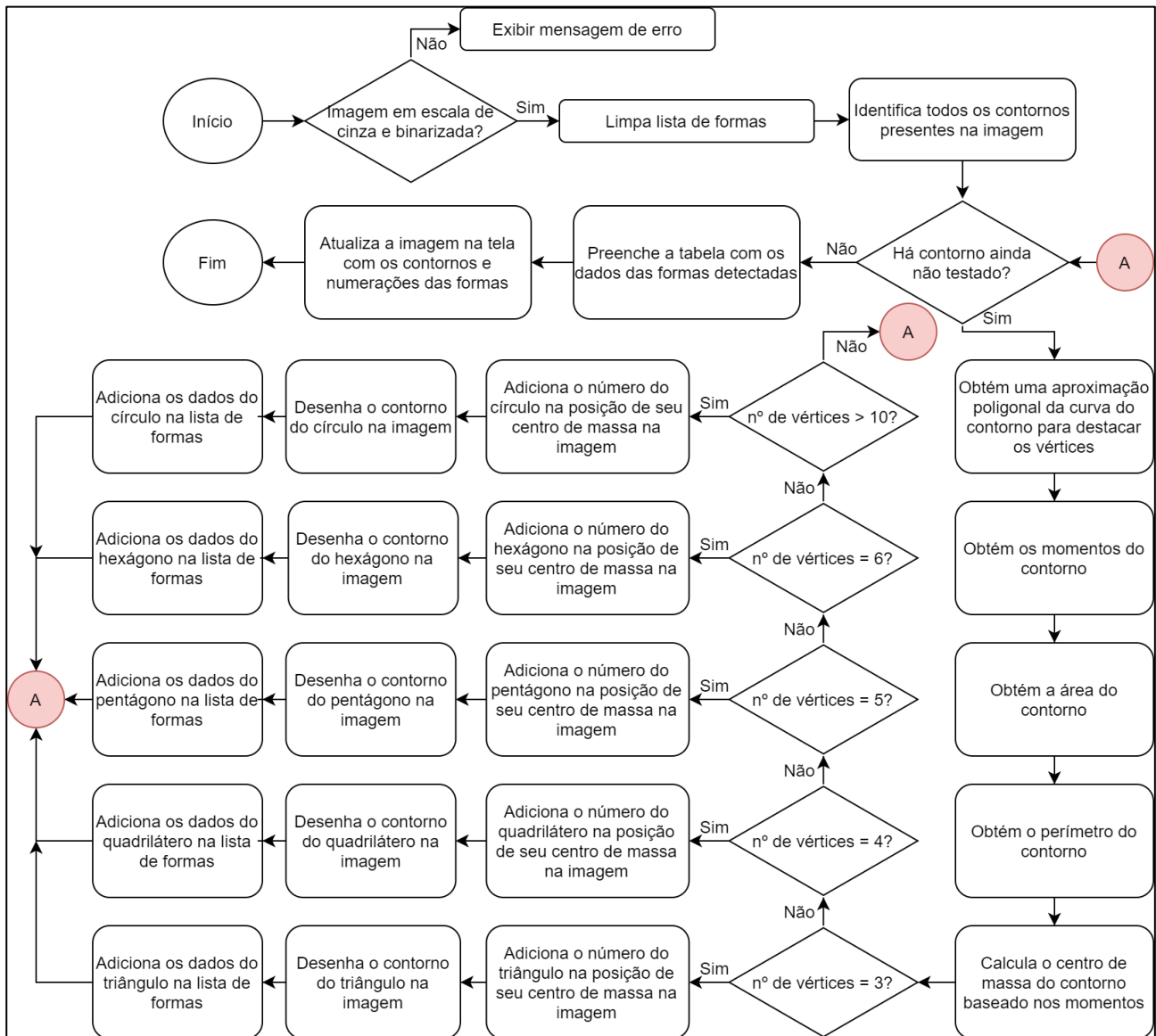
- Fazer o referenciamento dimensional, gerando o fator de conversão;
- Converter a imagem para escala de cinza;
- Binarizar a imagem de tal forma que fique com o fundo em preto e os objetos em branco;
- Usar o filtro de ruídos para limpar possíveis manchas na imagem, qualquer mancha

será identificada como contorno aparecerá na tabela de resultados caso não seja eliminada;

- Na aba 'Identificar Geometrias', clicar no botão 'Detectar Geometrias' (Figura 48, item 1);

O botão 'Detectar Geometrias' aciona a função 'identificarFormas' que funciona conforme o fluxograma a seguir:

Figura 60: Função 'identificarFormas'



Fonte: Elaborada pelo autor.

A lista de formas identificadas aparece na tabela de características obtidas (Figura 48), com cinco colunas exibindo o tipo de forma, área, perímetro, centro de massa (coordenadas do ponto em pixels) e o número de vértices do contorno.

5.4.6 Medição de áreas

a) Medição de áreas pelo método da contagem de pixels

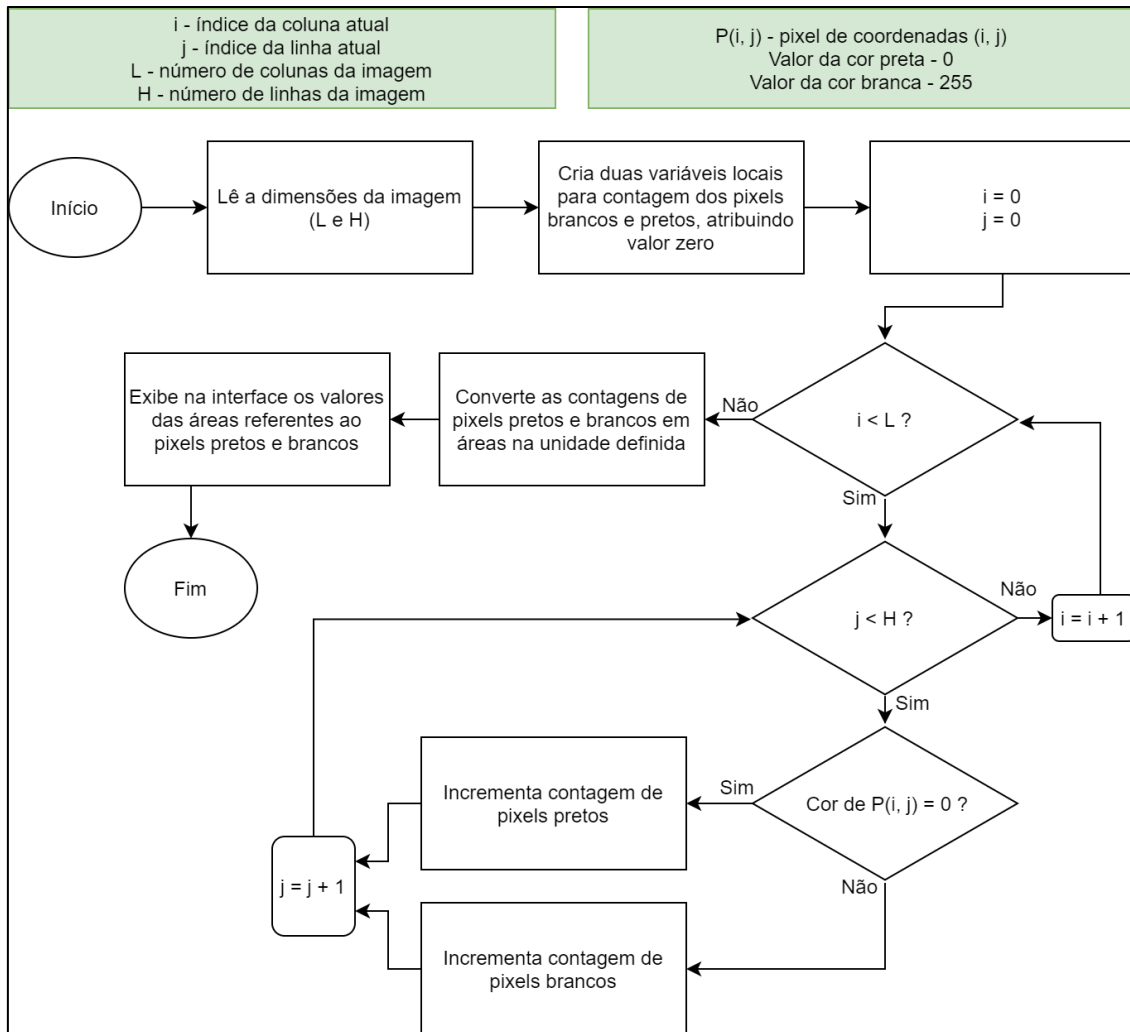
A lógica por trás desse método é baseada na soma das áreas dos pixels no interior do objeto, conforme fundamentado no item 5.3.2.

O que ocorre na prática é a contagem dos pixels brancos e pretos da imagem, sendo exibidos como resultado na interface os valores da área correspondente aos pixels de cada cor. Portanto, para que o resultado seja coerente é necessário que a imagem contenha apenas um fundo, preto ou branco e o objeto a ser medido, da cor restante. Caso haja mais de um objeto na imagem, a área resultante será a soma das áreas dos objetos. Então, para que a funcionalidade funcione corretamente, o usuário deve:

- Fazer o referenciamento dimensional, gerando o fator de conversão;
- Converter a imagem para escala de cinza;
- Binarizar a imagem de tal forma que reste apenas o fundo em preto ou branco e o objeto alvo da medição na cor restante;
- Utilizar o filtro de ruídos para eliminar possíveis manchas indesejadas.
- No grupo ‘Cálculo por contagem de pixels’ da aba ‘Áreas’, clicar no botão ‘Calcular’ (Figura 47, item 2);

O algoritmo para a contagem dos pixels por cor na imagem funciona conforme o fluxograma a seguir:

Figura 61: Função 'CalcAreaContagem'



Fonte: Elaborada pelo autor.

Em resumo, o algoritmo percorre todos os pixels da imagem avaliando se o valor da cor em escala de cinza é igual a zero, como a imagem é binarizada, cada pixel só pode ter assumido o valor 0 (preto) ou 255 (branco), então basta verificar essa condição para determinar qual variável de contagem será incrementada para cada pixel verificado. Havendo todos os pixels sido avaliados, as variáveis de contagem conterão as áreas de preto e de branco em pixels, sendo necessário apenas fazer a conversão para a unidade definida pelo usuário (equação 22) e exibir o resultado na interface.

b) Medição de áreas com o recurso nativo de contornos do OpenCV

Esse método é baseado no cálculo que o OpenCV faz automaticamente ao identificar contornos e é exatamente o mesmo usado na funcionalidade de identificação de formas geométricas. Internamente, conforme informa sua documentação, o OpenCV utiliza o teorema de Green para cálculo da área utilizando uma integral dupla, o que é invisível ao programador, sendo necessário apenas fazer a leitura do valor através da chamada de uma função.

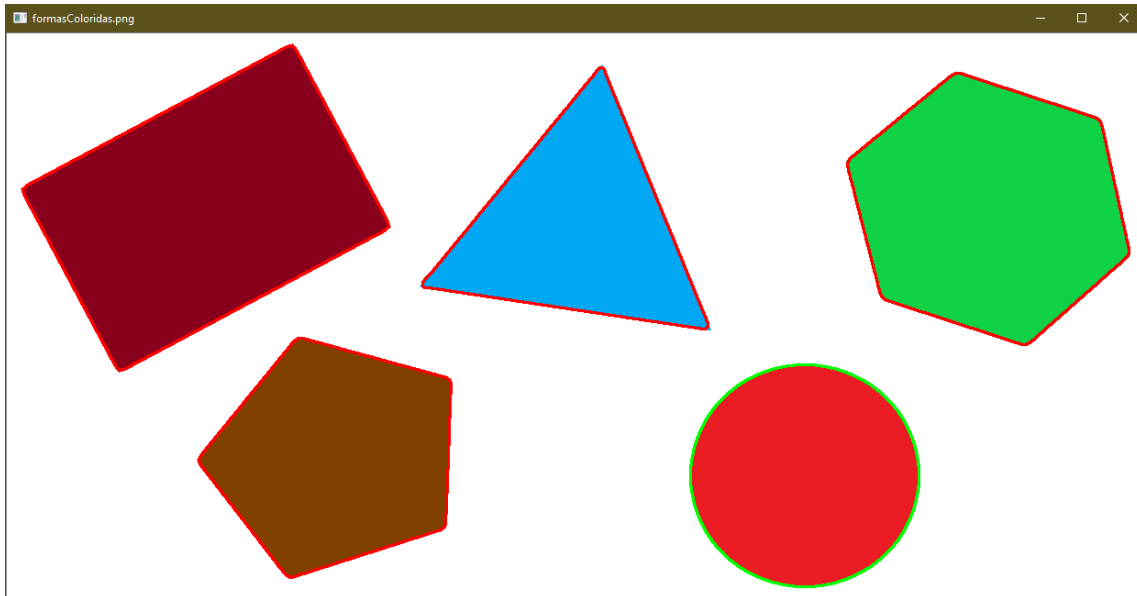
Pelo fato de o método também utilizar a identificação de contornos, as condições de uso dessa funcionalidade são muitos semelhantes às descritas no item 5.4.5. Portanto, o usuário deve proceder da seguinte maneira:

- Fazer o referenciamento dimensional, gerando o fator de conversão;
- Converter a imagem para escala de cinza;
- Binarizar a imagem de tal forma que fique com o fundo em preto e os objetos em branco;
- Usar o filtro de ruídos para limpar possíveis manchas na imagem, qualquer mancha será identificada como contorno aparecerá na tabela de resultados caso não seja eliminada;
- Na aba de áreas (Figura 47), clicar no botão ‘Identificar Contornos’ (item 6) para que o programa encontre todas os contornos presentes na imagem;
- Selecionar o contorno que deseja medir através da caixa de seleção (Figura 47, item 7). Os contornos serão realçados na imagem conforme o mouse passar por cima das opções da caixa de seleção para facilitar a identificação.
- Clicar no botão ‘Calcular’ (Figura 47, item 8);

O botão ‘Identificar Contornos’ aciona uma função que faz a identificação dos contornos na imagem, armazena seus em uma lista acessível globalmente, preenche a caixa de seleção com as opções referentes aos contornos encontrados, desenha-os em uma cópia da imagem original e exibe essa imagem com os contornos na tela.

O realce do contorno selecionado é feito através de um evento presente no componente ‘combo box’ do PyQt5 que permite a chamada de uma função passando como parâmetro o índice da opção sobre a qual está o ponteiro do mouse. A função criada percorre os contornos da lista gerada no passo anterior e os desenha em uma cópia da imagem original, sendo que o contorno correspondente ao índice atualmente selecionado é desenhado em verde e o restante dos contornos é desenhado em vermelho, resultando em uma resposta visual clara para o usuário sobre qual contorno está selecionando. Um exemplo do realce é mostrado a seguir:

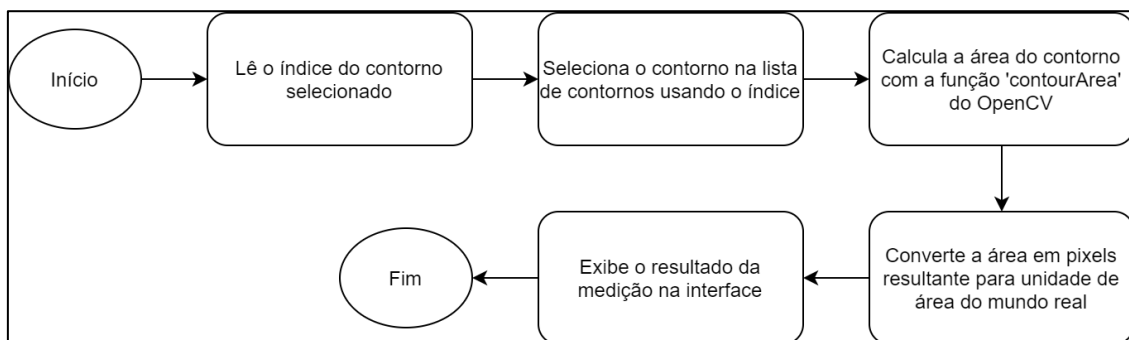
Figura 62: Contorno selecionado (círculo) colorido em verde e o restante em vermelho



Fonte: Elaborada pelo autor.

Feita a seleção do contorno, o botão 'Calcular' aciona o algoritmo principal, que funciona conforme segue:

Figura 63: Cálculo da área do contorno



Fonte: Elaborada pelo autor.

A conversão da área em pixels para unidade selecionada é feita através da equação 22, o resultado é exibido na interface conforme mostrado no item 9 da Figura 47.

5.4.7 Medição de raios e do centro de circunferências

Os dois métodos disponíveis para calcular o raio e determinar o centro de circunferências e arcos funcionam com base no mesmo algoritmo de cálculo baseado no uso de três pontos pertencentes à circunferência, sendo que a única diferença entre eles é a forma como esses pontos são selecionados. No primeiro, o usuário seleciona entre os pontos criados manualmente através dos métodos descritos em 5.4.2. No segundo, os pontos são selecionados automaticamente pelo programa.

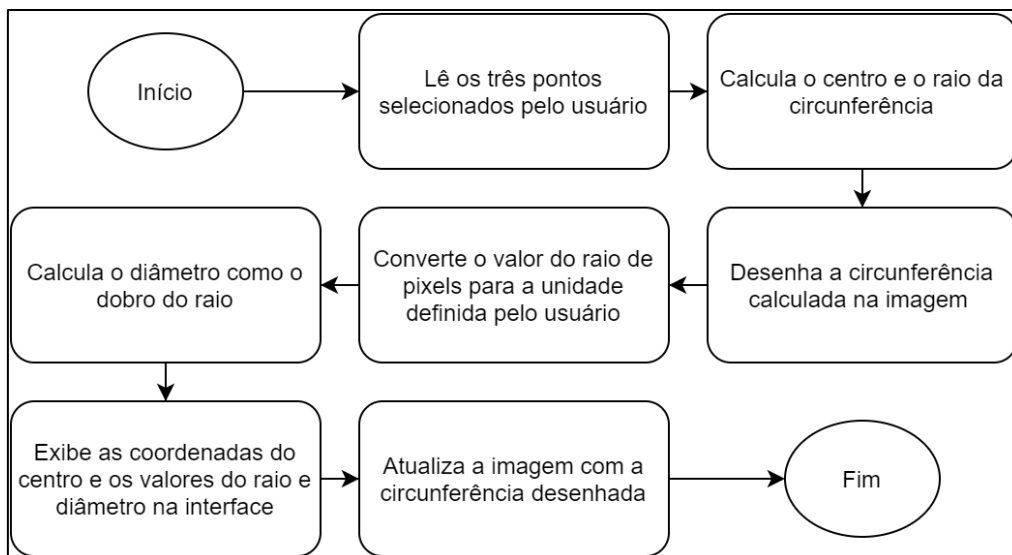
a) Pela seleção manual dos pontos

Para medir circunferências por esse método, o usuário deve seguir os seguintes passos:

- Carregar a imagem;
- Fazer o referenciamento dimensional, gerando o fator de conversão;
- Criar três pontos sobre a circunferência, arco ou borda de círculo alvo da medição;
- Selecionar os três pontos criados através das caixas de seleção na Figura 49, item 4;
- Se desejar que os pontos selecionados sejam desenhados na imagem, marcar as opções do item 5 (Figura 49);
- Clicar no botão ‘Calcular’ (Figura 49, item 6);

O algoritmo que baseia a função executada funciona conforme segue:

Figura 64: Função de cálculo da circunferência com seleção manual dos pontos



Fonte: Elaborada pelo autor.

O cálculo do centro e do raio da circunferência é calculado conforme explicado em 5.3.6. A função que implementa os cálculos recebe como parâmetros três pontos, opera sobre as suas coordenadas e retorna as coordenadas do centro da circunferência formada com base nos três pontos e o valor de seu raio em pixels.

O desenho da circunferência é feito usando a função ‘circle’ do OpenCV, que adiciona a circunferência na imagem alvo recebendo como parâmetros as coordenadas do centro, raio, espessura da linha e cor.

A conversão do valor de raio de pixels para unidade real é feita utilizando a equação 21.

b) Pela seleção automática dos pontos

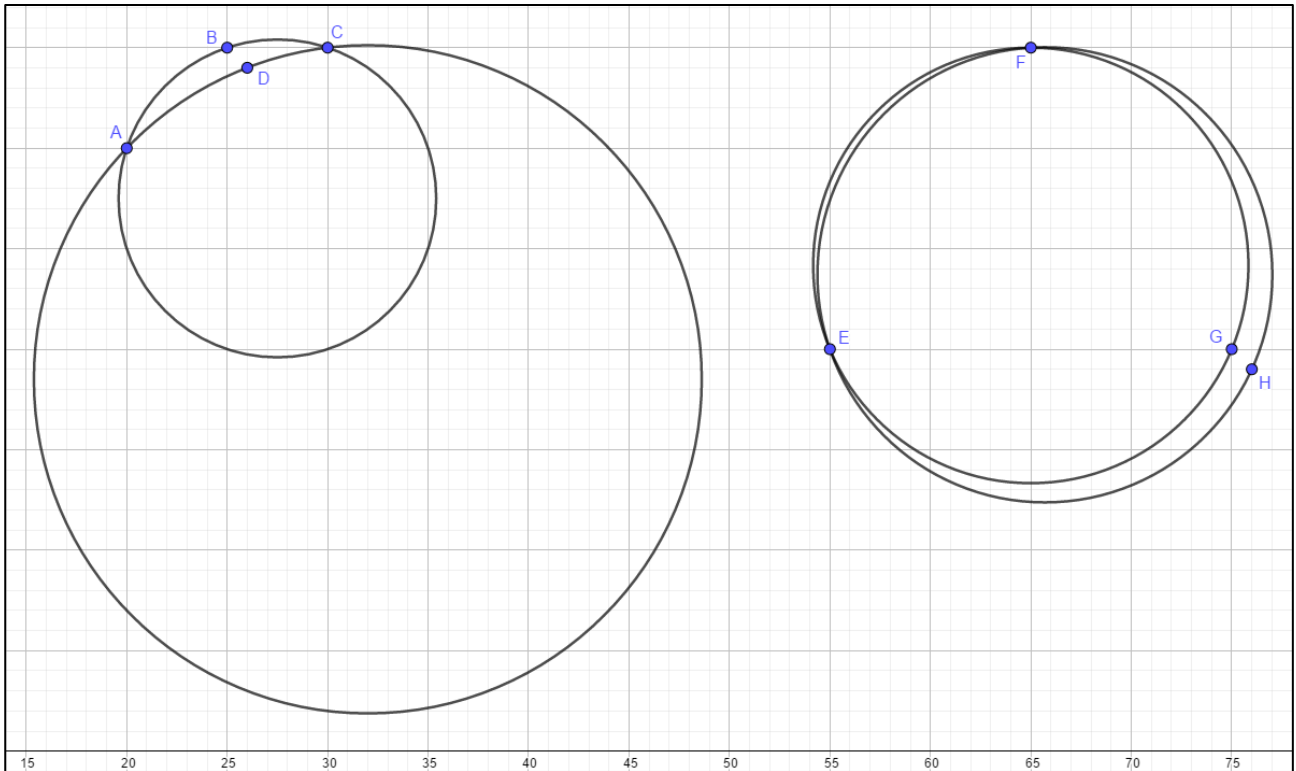
Para construir a circunferência com esse método, o usuário deve:

- Carregar a imagem;
- Fazer o referenciamento dimensional, gerando o fator de conversão;
- Converter a imagem para escala de cinza;
- Binarizar a imagem de tal forma que fique com o fundo em preto e os objetos em branco;
- Usar o filtro de ruídos para limpar manchas indesejadas na imagem;
- Clicar no botão ‘Detectar Círculo’ (Figura 49, item 10);
- Selecionar círculo a ser medido (Figura 49, item 11);
- Clicar no botão ‘Calcular’ (Figura 49, item 12);

Esse método usa a detecção de contornos do OpenCV, utilizando novamente o número de vértices para detectar contornos como sendo círculos.

O desafio nesse caso é escolher pontos pertencentes aos contornos que estejam posicionados bem o suficiente para serem tomados como base na construção da circunferência. Para ilustrar o que isso quer dizer será utilizada a Figura 65. Nela são mostradas quatro circunferências, cada qual construída a partir de três pontos. As duas primeiras compartilham os pontos A e C e se diferem pelos pontos B e D. As duas últimas compartilham os pontos E e F e se diferem pelos pontos G e H.

Figura 65: Distribuição dos pontos geradores da circunferência para minimização de erros



Fonte: Elaborada pelo autor.

As coordenadas dos pontos são mostradas na tabela:

Tabela 8: Coordenadas dos pontos da Figura 65

Ponto A	Coordenada X	Coordenada Y
A	20	30
B	25	35
C	30	35
D	26	34
E	55	20
F	65	35
G	75	20
H	76	19

Observando a imagem é fácil notar que os pontos geradores das primeiras duas circunferências estão concentrados numa mesma região enquanto os que geram as duas últimas estão espalhados de maneira quase equidistante ao longo do contorno. Verificando as coordenadas dos pontos, pode-se observar também que a distância entre os pontos B e D é a mesma que a distância entre os pontos G e H. Por último, é notável que a diferença entre os diâmetros das duas primeiras circunferências é muito maior do que a diferença entre os diâmetros das duas últimas.

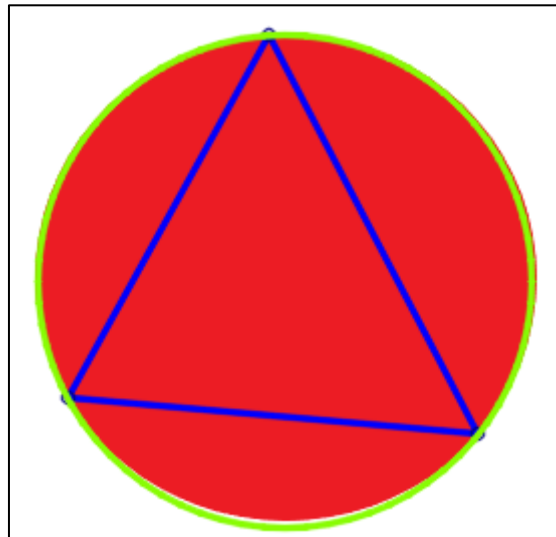
Considerando todas as observações feitas, pode-se concluir que quanto mais uniformemente

espalhados estiverem os pontos selecionados ao longo da circunferência, menor será a influência negativa de seus erros de posicionamento sobre o resultado da medição.

Dessa forma, o algoritmo que irá selecionar os pontos automaticamente deve usar algum critério que favoreça essa distribuição. A solução encontrada para isso foi selecionar os pontos dentre os vértices encontrados na circunferência, que são pontos gravados em uma lista. Para escolher quais os pontos dentre os vértices são os melhores de acordo com o critério de espalhamento ao longo do contorno, utiliza-se o número total de vértices dividido por três; o resultado dessa divisão é utilizado para definir os índices dos três pontos desejados, por exemplo, em uma circunferência para a qual o OpenCV identifique 15 vértices, o intervalo entre os índices é $15/3 = 5$, portanto os índices dos vértices selecionados serão: 0, 5 e 10.

Testando o algoritmo que segue a lógica descrita, pode-se observar que a estratégia se mostrou eficiente, proporcionando uma boa distribuição dos pontos. Na amostra do teste (Figura 66), o círculo original aparece em vermelho, as linhas em azul ligam os pontos selecionados dentre os vértices através do algoritmo e a circunferência em verde é o resultado final obtido através do cálculo de centro e raio.

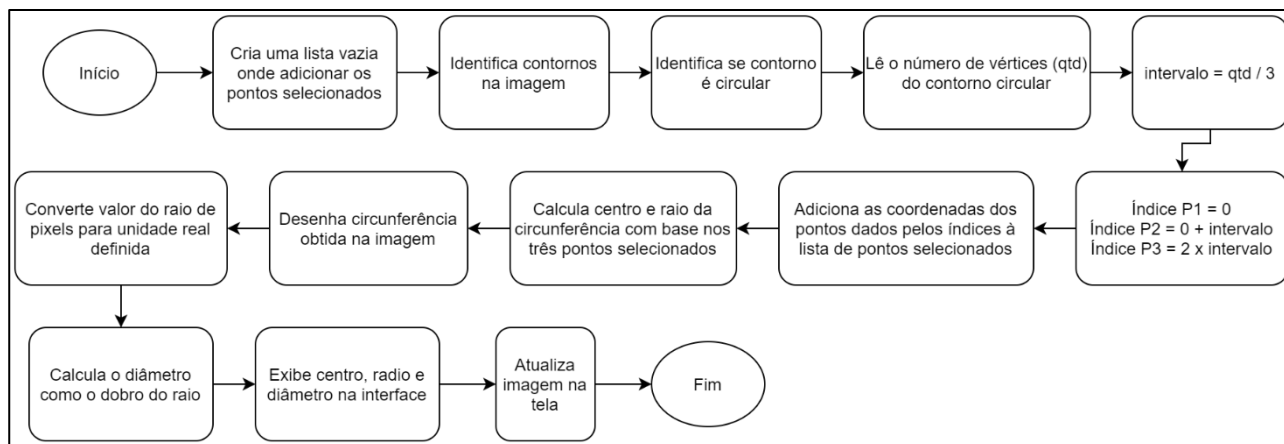
Figura 66: Teste do algoritmo de seleção automática dos pontos para construção da circunferência



Fonte: Elaborada pelo autor.

O algoritmo que implementa a funcionalidade no programa é apresentado a seguir:

Figura 67: Função de cálculo da circunferência com seleção automática dos pontos



Fonte: Elaborada pelo autor.

A identificação do contorno circular é feita chegando o número de vértices identificado pelo OpenCV. O cálculo de centro e raio, o desenho da circunferência e a conversão do raio de pixels para a unidade definida pelo usuário são feitos utilizando as mesmas funções usadas no método de seleção manual dos pontos.

Cada um desses métodos tem ao menos uma vantagem em relação ao outro. O método de seleção manual de pontos permite a medição de arcos e dispensa a conversão da imagem para escala de cinza a o processo de binarização, enquanto o método de seleção automática de pontos dispensa o usuário da necessidade de criar os pontos e permite a medição de várias circunferências ao mesmo tempo.

5.4.8 Medição de ângulos

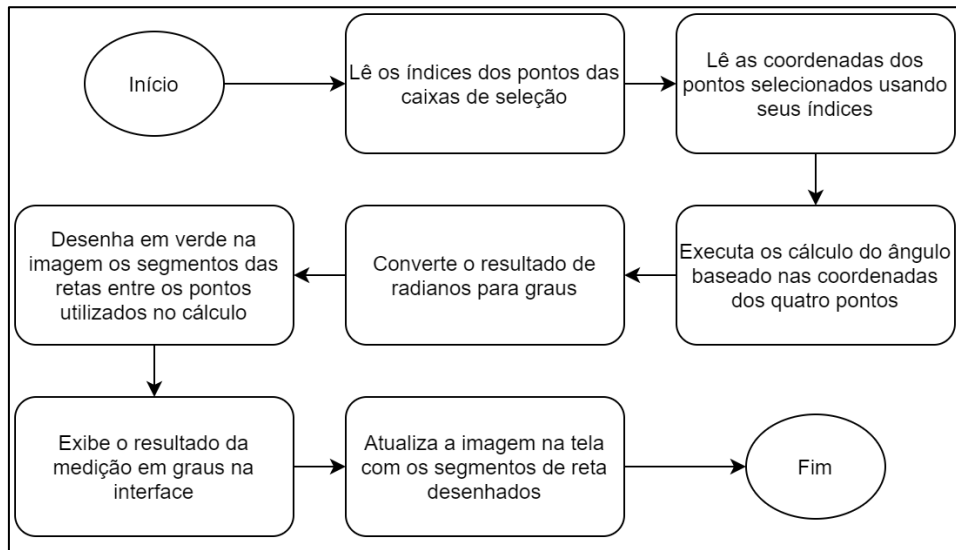
A medição de ângulos é feita através das coordenadas de quatro pontos criados pelo usuário, utilizando a equação 20, conforme explicado em 5.3.7. Dentre as funcionalidades principais é a que tem menos requisitos a serem preenchidos antes de ser aplicada, pois não depende do fator de conversão e não exige que sejam feitos processamentos sobre a imagem.

Para medir ângulos entre retas o usuário deve proceder da seguinte maneira:

- Carregar a imagem;
- Criar quatro pontos, dois em cada reta entre as quais se deseja medir o ângulo;
- No menu de ângulos da aba ‘Ferramentas’ (Figura 50), selecionar os dois pontos da primeira reta (item 8) e os dois pontos da segunda reta (item 9);
- Clicar no botão ‘Calcular’ (Figura 50, item 10);

O cálculo do ângulo será feito utilizando o algoritmo seguinte:

Figura 68: Função para o cálculo de ângulos



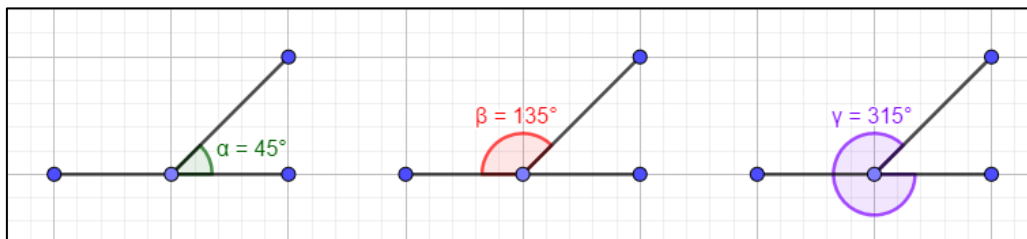
Fonte: Elaborada pelo autor.

O resultado é exibido em um elemento (Figura 50, item 11) logo abaixo do botão ‘Calcular’.

É importante observar que o ângulo calculado através da equação 20 sempre será o menor ângulo entre as duas retas geradas pelos pontos dados, ou seja, o resultado da medição sempre representará o valor do ângulo agudo (entre 0° e 90°) ou obtuso (entre 90° e 180°) entre as retas alvo. Sabendo que haverá casos em que o usuário desejará saber de outro ângulo entre as retas que não seja o calculado, optou-se por calcular também o ângulo suplementar (aquele que somado ao obtido resulta em 180°) e o ângulo replementar (aquele que somado ao obtido resulta em 360°) através de operações simples de subtração. Esses cálculos foram implementados no programa, mas seus resultados não chegaram a ser adicionados na interface gráfica.

A Figura 69 ilustra o que significam o ângulo agudo (α - calculado pelo software) e seus ângulos suplementar ($\beta = 180^\circ - \alpha$) e replementar ($\gamma = 360^\circ - \alpha$).

Figura 69: Possíveis ângulos desejados entre os mesmos segmentos de reta



Fonte: Elaborada pelo autor.

6. VALIDAÇÃO EXPERIMENTAL

Para validar as funcionalidades do programa foram feitos primeiro testes usando imagens criadas digitalmente, com o objetivo de avaliar a precisão das medições sem a influência dos erros provenientes da fotografia e com valores teóricos das medidas bem controlados. E por fim foram feitos testes utilizando peças reais, com os resultados comparados aos de medições feitas com instrumentos do laboratório de metrologia do IFSC. Devido a inacessibilidade de um bloco padrão, os testes com peças reais foram feitos apenas com referenciamento por meio de uma régua usando a marcação manual de pontos.

Os resultados dos testes são avaliados principalmente em relação ao seu erro máximo, dado pela equação 23:

$$E_{max} = 0,1 + \frac{L}{100} \quad (23)$$

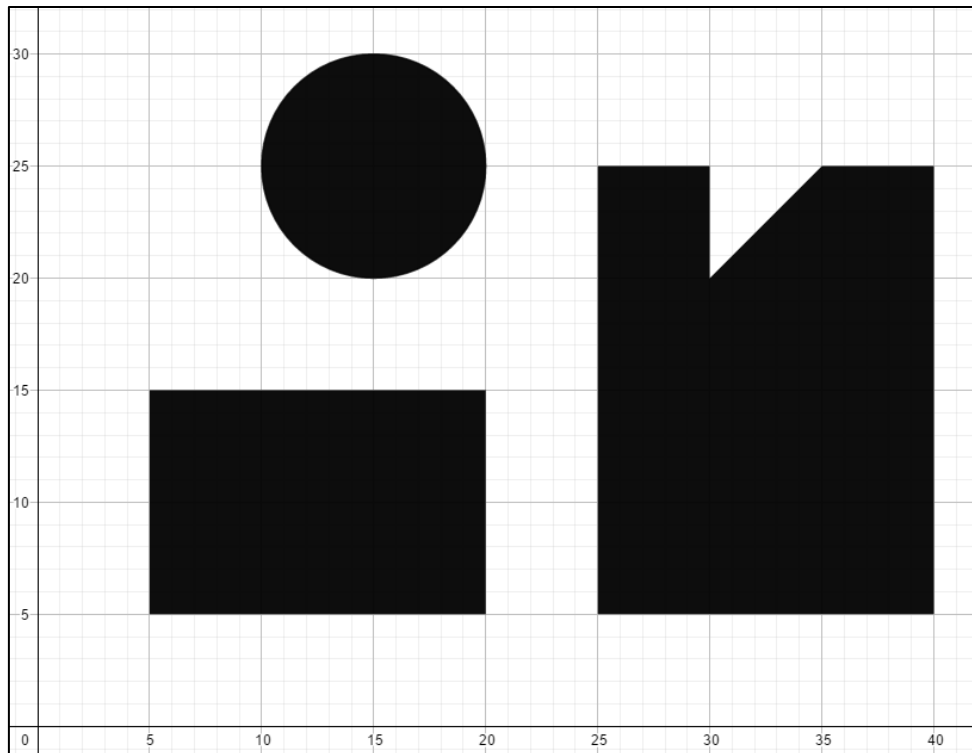
Onde:

- L – Valor nominal da medida

6.1 Testes com imagens geradas digitalmente

Por meio do programa Geogebra, criou-se uma imagem a partir de qual é possível testar as principais funcionalidades implementadas no projeto tanto com referenciamento com base em uma distância de referência quanto usando o algoritmo de referenciamento com bloco-padrão. Essa imagem é mostrada a seguir (Figura 70):

Figura 70: Imagem gerada digitalmente para os testes

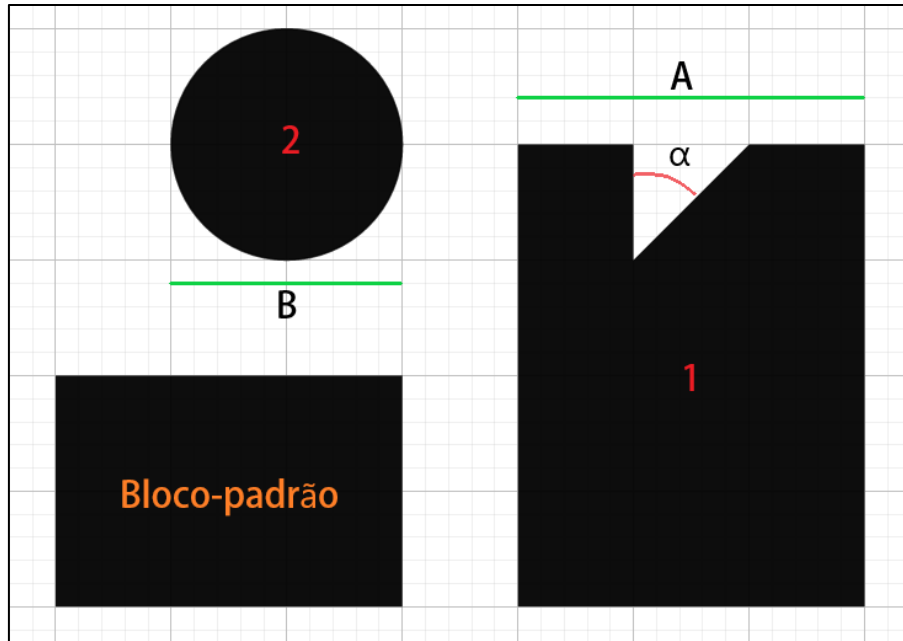


Fonte: Elaborada pelo autor.

A imagem criada permite o referenciamento através de uma distância conhecida, apresentando uma grade que facilita a noção do usuário ao criar os pontos. Ela também permite o uso do referenciamento com bloco-padrão através do retângulo criado nas dimensões corretas para isso. Permite a medição de comprimentos em vários lugares diferentes, a medição de áreas através dos dois métodos disponíveis no programa, o cálculo de circunferências também através dos dois métodos disponíveis e a medição de ângulos. A funcionalidade restante, a identificação e medição de formas, foi testada através de outra imagem.

Antes de proceder às medições, é necessário definir o que se deseja medir. Para essa imagem, os alvos da medição foram definidos conforme apresentado na Figura 71: a largura (A) e a área do polígono 1, o diâmetro (B) do círculo 2 e o ângulo presente na reentrância do polígono 1 (α).

Figura 71: Alvos de medição



Fonte: Elaborada pelo autor.

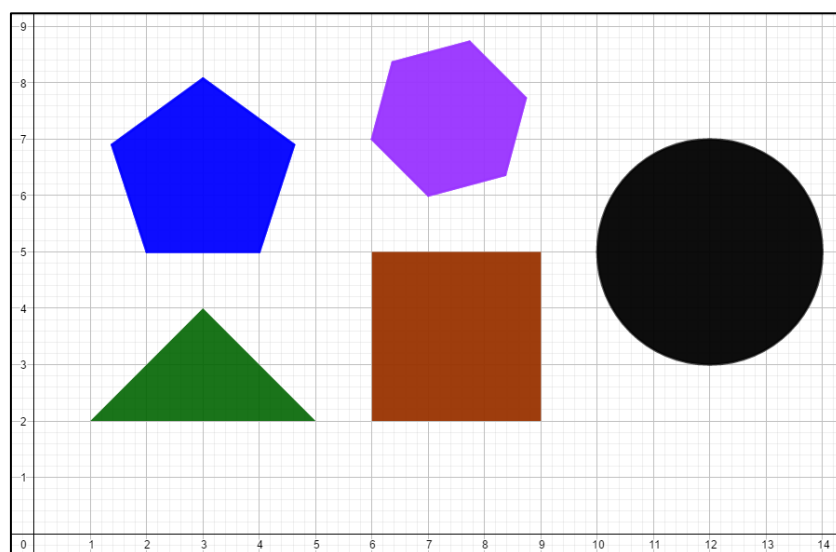
A unidade escolhida para a escala da grade é o milímetro, de forma que o bloco padrão mede 15mm x 10mm e os valores nominais para as medidas seleccionadas são:

Tabela 9: Valores nominais das medidas a serem testadas

Medida	A (mm)	B (mm)	Área 1 (mm ²)	α (°)
Valor nominal	15.00	10.00	287.50	45.00

Para os testes de detecção e medição de formas geométricas foi criada outra figura (Figura 72), também com auxílio do Geogebra.

Figura 72: Imagem gerada para testar detecção e medição de formas



Fonte: Elaborada pelo autor.

O Geogebra possui o recurso de chamada de funções; como as medições feitas nas formas são a área e o perímetro, foram utilizadas as duas funções do Geogebra que retornam os valores de área e perímetro das formas criadas. Os valores nominais das formas presentes na imagem obtidos usando essas funções são mostrados na tabela seguinte.

Tabela 10: Áreas e perímetros das formas geométricas

Forma	Triângulo	Quadrado	Pentágono	Hexágono	Círculo
Área (mm ²)	4.00	9.00	6.88	5.20	12.57
Perímetro (mm)	9.66	12.00	10.00	8.49	12.57

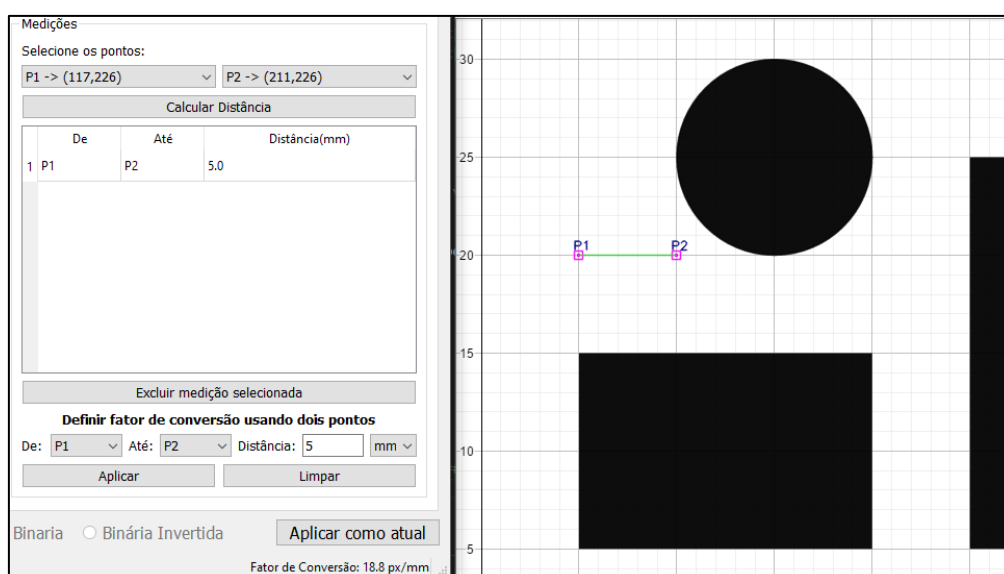
Para apresentação dos testes, inicialmente serão apresentados os procedimentos de calibração, através dos dois métodos disponíveis, e para a realização de cada tipo de medição. Posteriormente serão apresentados os resultados com 10 medições para cada medida alvo, a partir das quais foram calculados: desvio-padrão, tendência e repetibilidade e avaliados os erros máximos.

6.1.1 Procedimentos para a realização das medições

a) Referenciamento dimensional

Para a calibração feita com marcação manual dos pontos, tendo a imagem sido carregada para dentro do programa, foram criados dois pontos separados por 5 mm utilizando como referência a grade presente na imagem. Conforme mostra a Figura 73, o fator de conversão obtido foi de 18,8 px/mm e foi realizada uma medição de distância entre os mesmos pontos usados para calibração, que resultou em exatos 5,0 mm para comprovar a calibração.

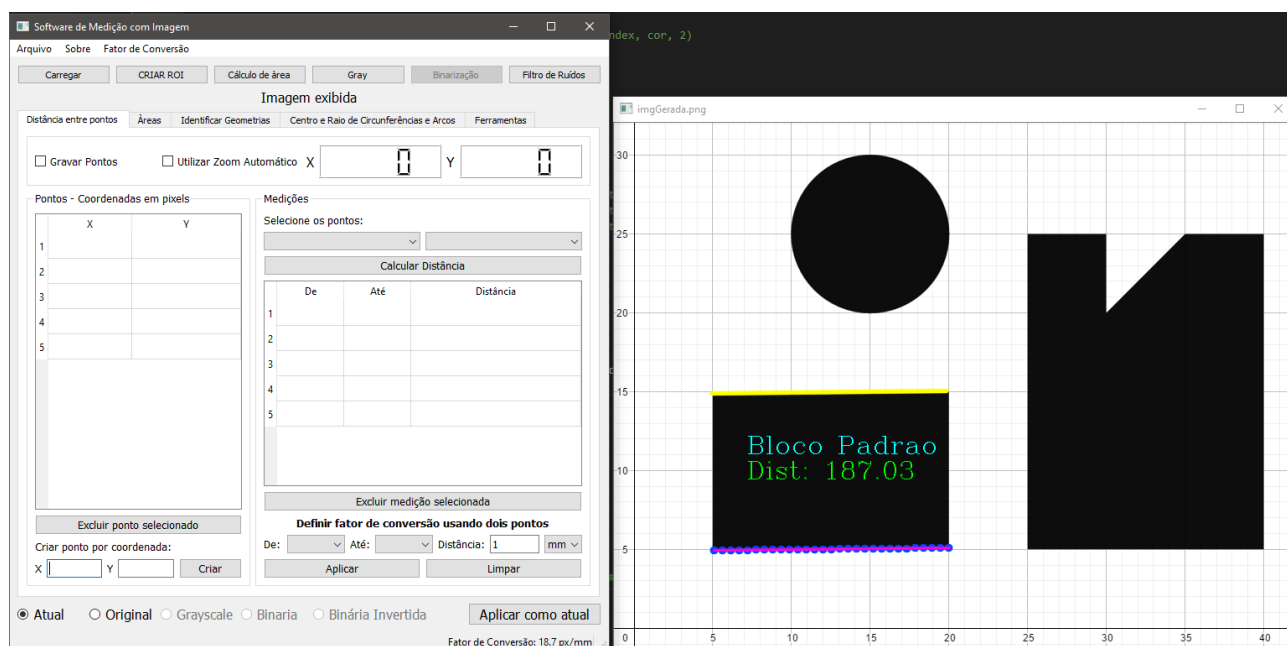
Figura 73: Referenciamento com criação manual de pontos



Fonte: Elaborada pelo autor.

Para a calibração utilizando o bloco-padrão simulado através de um retângulo, a imagem foi carregada no programa e a opção ‘Obter com bloco-padrão’ foi selecionada na barra de menu. A Figura 74 mostra o resultado da calibração com o bloco-padrão identificado e medido automaticamente. O valor de fator de conversão obtido foi de 18.7 px/mm, valor muito próximo ao obtido com a marcação manual de pontos.

Figura 74: Calibração através do bloco-padrão simulado



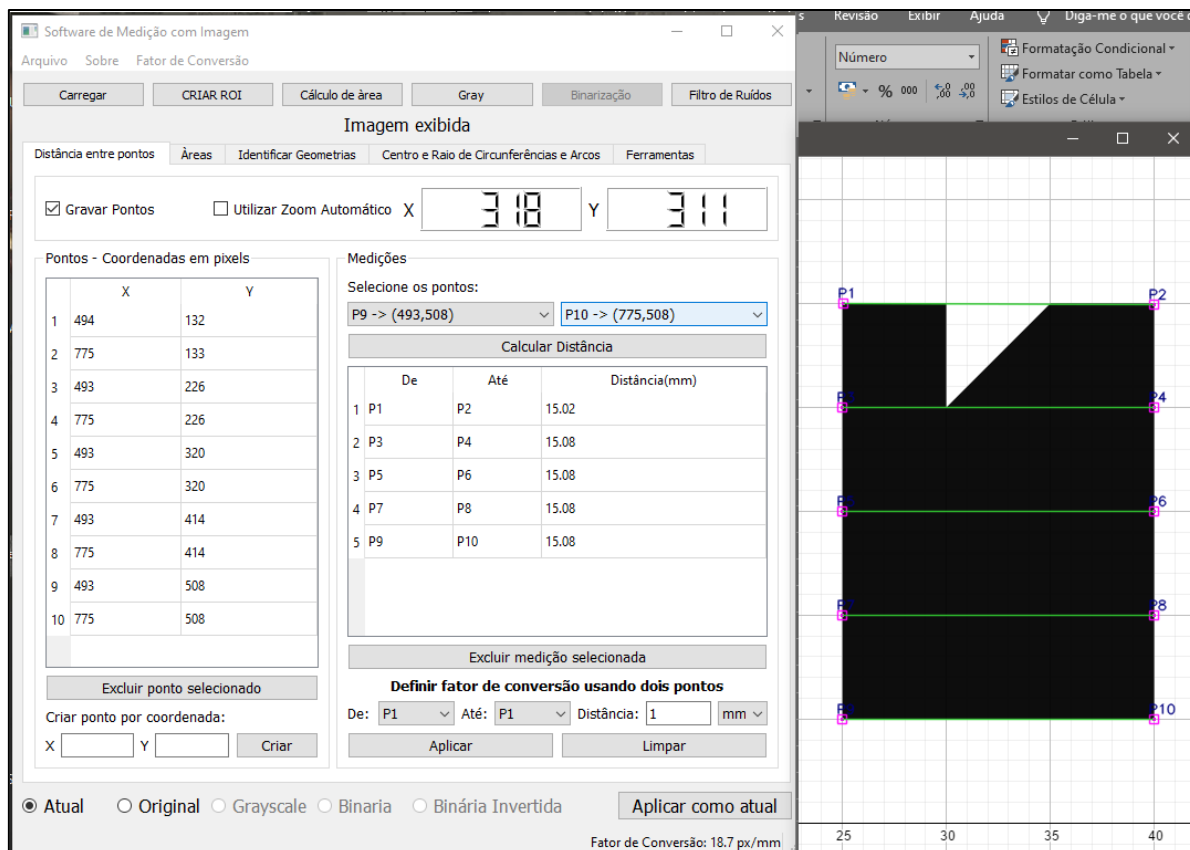
Fonte: Elaborada pelo autor.

Após a calibração, todos os procedimentos de medição ocorrem da mesma maneira, portando serão mostrados apenas uma vez.

b) Medição da largura A

Como essa largura se mantém fixa ao algo do polígono, foi possível fazer várias medições de uma só vez. Para essa medida foram criados 10 pontos e feitas 5 medições de cada vez, conforme mostra a Figura 75, captura feita durante as medições com referenciamento via bloco-padrão.

Figura 75: Medição da largura A do polígono 1

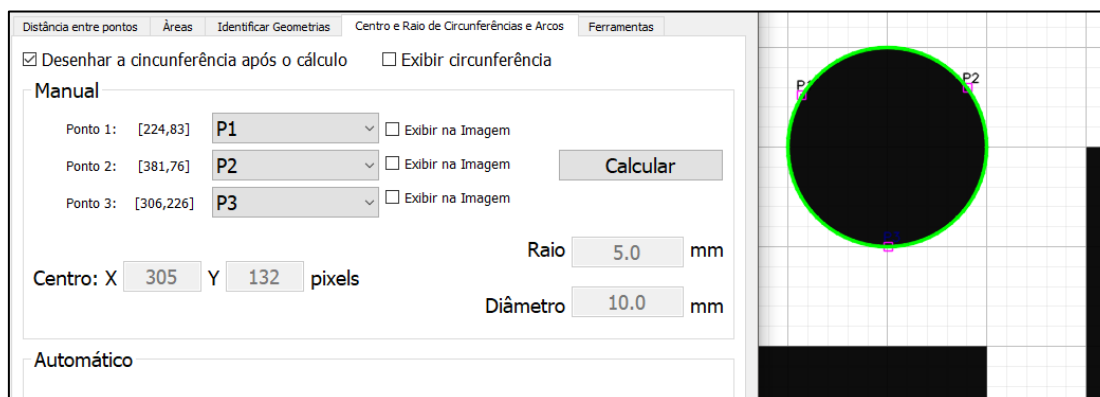


Fonte: Elaborada pelo autor.

c) Medição do diâmetro B com seleção manual dos pontos

Para essa medição foram criados 3 pontos na borda do círculo, tentando deixar um espaçamento igual entre eles. Foram feitas 10 medições usando sempre novos pontos marcados em posições arbitrárias, apenas seguindo o padrão de espaçamento. A Figura 76 apresenta a captura de umas das medições, com resultado de exatos 10 mm, mostrando os 3 pontos selecionados e a circunferência resultante da medição na imagem.

Figura 76: Medição do círculo com seleção manual de pontos

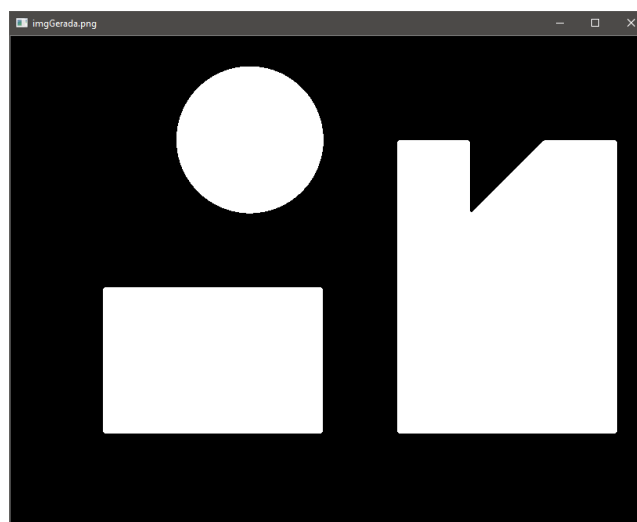


Fonte: Elaborada pelo autor.

d) Medição do diâmetro B com seleção automática dos pontos

Para essa medição, após o referenciamento, foi feita a conversão da imagem para escala de cinza, seguida da binarização.

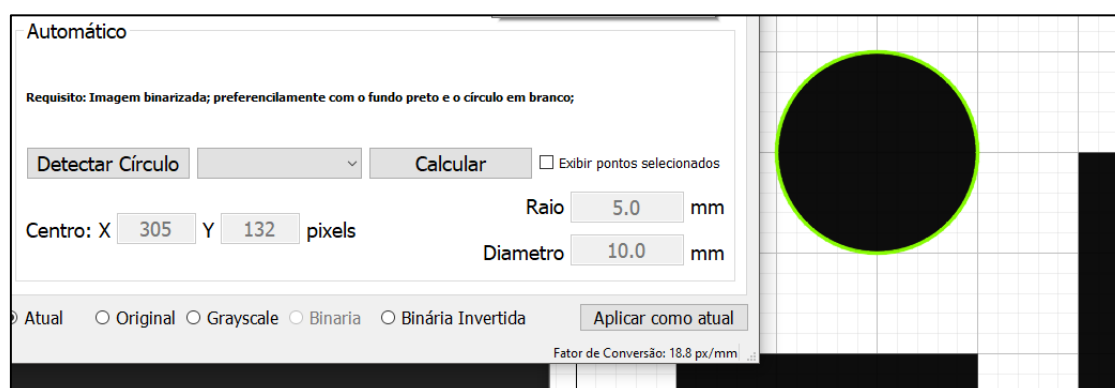
Figura 77: Imagem binarizada para detecção de contornos



Fonte: Elaborada pelo autor.

Estando a imagem preparada, exibindo apenas os contornos dos objetos (Figura 77), aplicou-se a função de detecção e cálculo do círculo para realizar a medição.

Figura 78: Medição do círculo com seleção automática de pontos



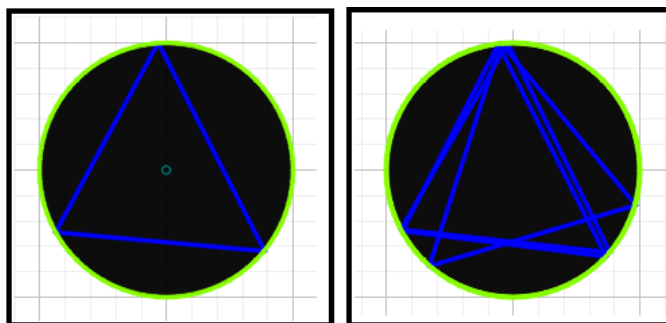
Fonte: Elaborada pelo autor.

A captura foi feita em umas das medições com uso do referenciamento manual. Assim como na medição com seleção manual de pontos, o resultado apresentado nessa captura também coincidiu com o valor nominal.

Foi feito um teste adicional exibindo linhas ligando os 3 pontos selecionados automaticamente, mostrando variações nas posições dos pontos selecionados conforme variou o limiar usado para a

binarização. Na Figura 79, à esquerda aparece um teste mostrando as linhas ligando os pontos usados como base para o cálculo, o centro encontrado e a circunferência desenhada com base no resultado da medição; à direita são mostradas as linhas já mencionadas em medições sobrepostas, mostrando a consistência do algoritmo independente dos pontos selecionados.

Figura 79: Testes com linhas ligando os pontos selecionados pelo algoritmo

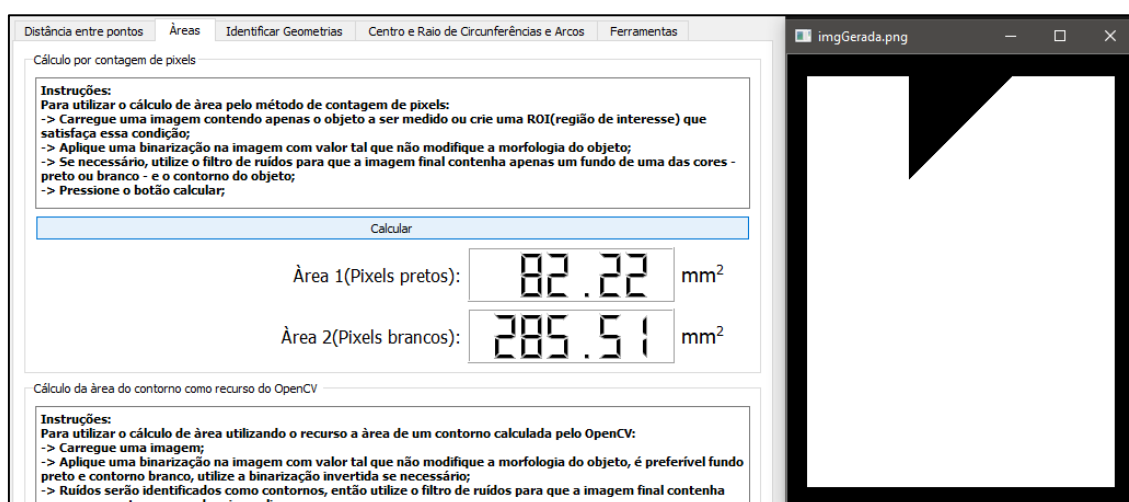


Fonte: Elaborada pelo autor.

e) Medição da área do polígono 1 pela contagem de pixels

Feita a calibração, criou-se uma região de interesse contendo o polígono a ser medido (Figura 80). Após passar pela conversão para escala de cinza e binarização foi feita a medição, com resultados aproximados a amostra a seguir:

Figura 80: Medição da área do polígono 1 pelo método da contagem de pixels



Fonte: Elaborada pelo autor.

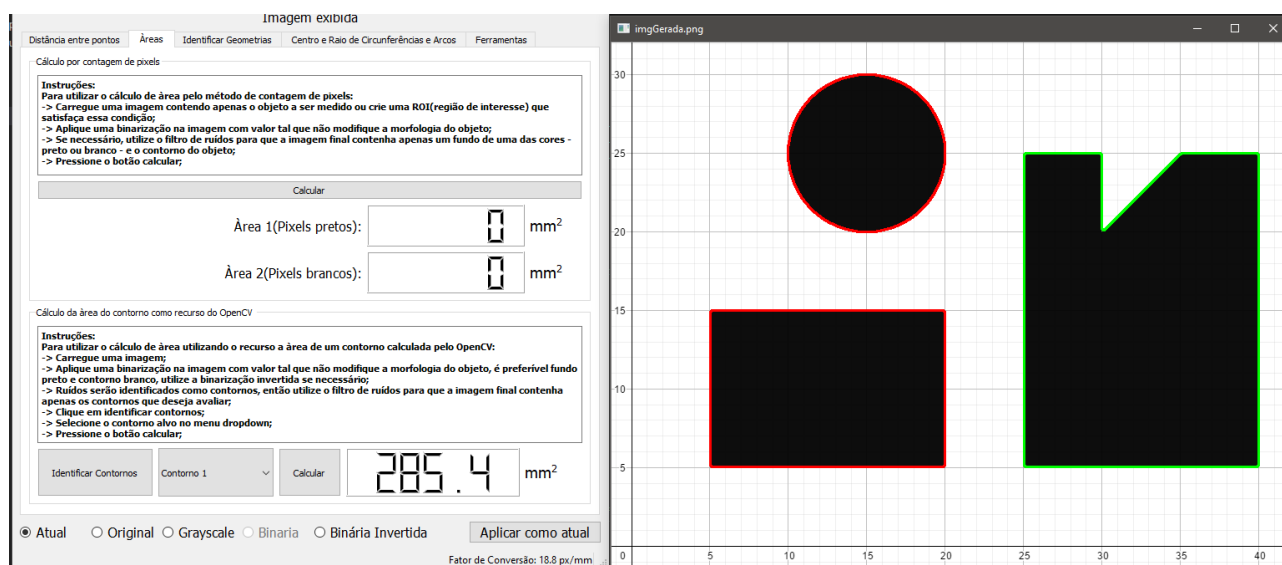
Observou-se que o resultado dessa medição é influenciado pelo limiar definido na hora de aplicar a binarização, pois o objeto pode aumentar ou diminuir dependendo desse valor.

f) Medição da área do polígono 1 pelo recurso nativo do OpenCV

Com a imagem no estado mostrado na Figura 77, aplicou-se a identificação de contornos,

seguida da seleção do contorno referente ao polígono 1 e do cálculo, com resultados conforme mostra da Figura 81.

Figura 81: Medição da área do polígono 1 com a função do OpenCV (teorema de Green)

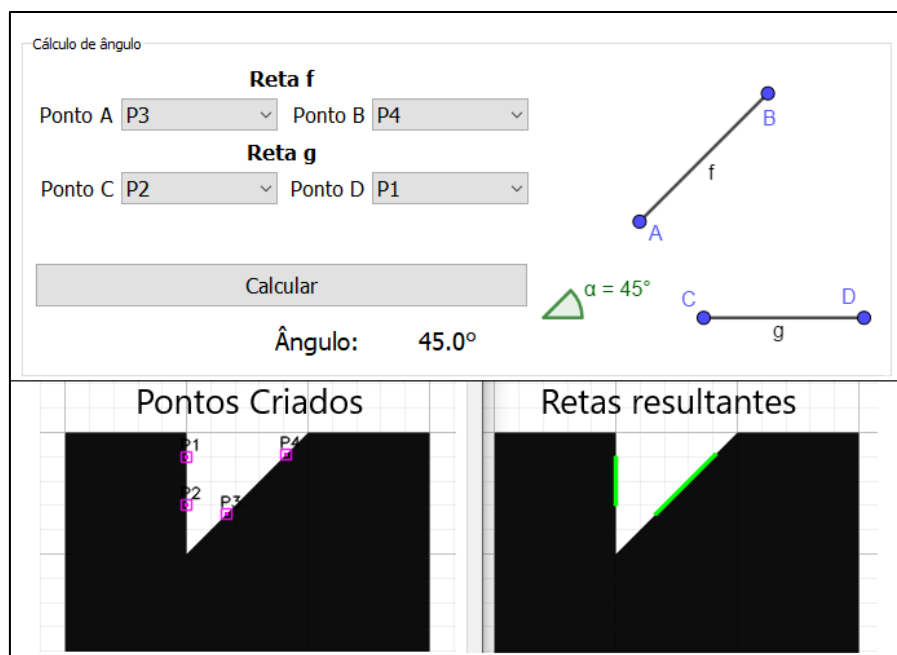


Fonte: Elaborada pelo autor.

g) Medição do ângulo presente no polígono 1

Uma vez que a medição de ângulos não exige nenhuma preparação prévia, nem mesmo referenciamento dimensional, os testes dessa funcionalidade se resumiram à criação dos 4 pontos nas ruas retas que geram o ângulo que se deseja medir, a seleção dos mesmos no menu de cálculo do ângulo e a execução do cálculo em si. A Figura 82 exibe uma amostra de como foram feitas as medições, apresentando recortes da tela com o menu da ferramenta, os 4 pontos criados e as retas exibidas após o cálculo para indicar ao usuário o que foi medido.

Figura 82: Teste da medição de ângulo com imagem gerada digitalmente

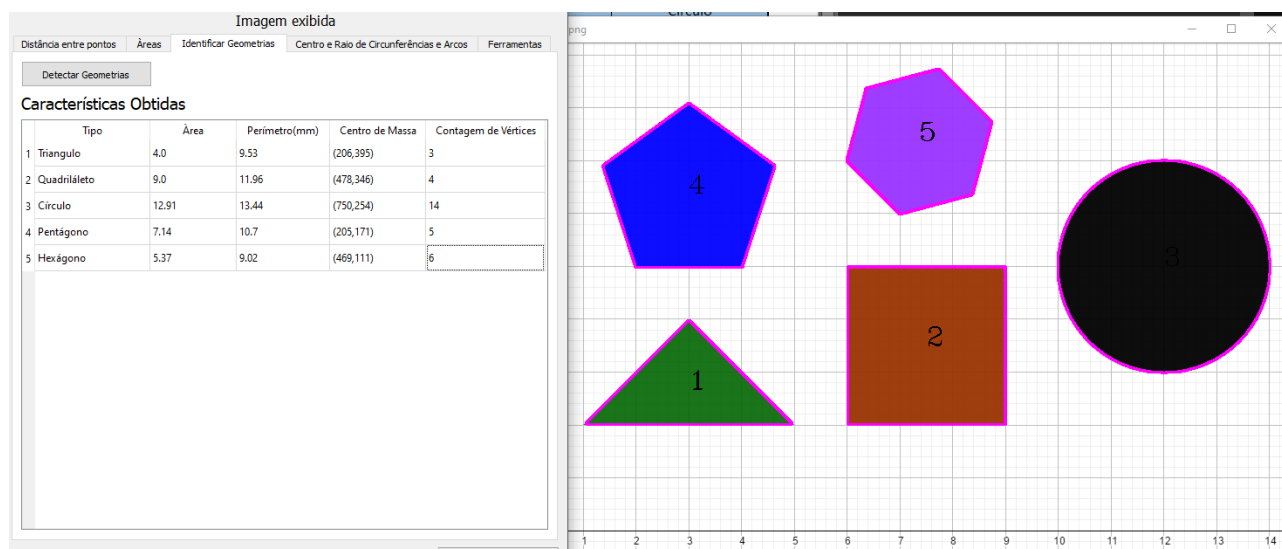


Fonte: Elaborada pelo autor.

A captura mostra uma das vezes em que o resultando coincidiu precisamente com o valor nominal do ângulo, mas adiante é mostrado que os resultados variaram consideravelmente dependendo dos pontos criados.

h) Detecção de medição de formas geométricas

Figura 83: Teste detecção de formas



Fonte: Elaborada pelo autor.

6.1.2 Medições com fator de conversão obtido a partir da distância entre pontos

Tabela 11: Resultados das medições com imagem gerada digitalmente e referenciamento manual

Medida	A	B		Área peça 1		Ângulo
		Manual	Auto	Contagem	Contorno	
Valor Nominal	15.00	10.00	10.00	287.50	287.50	45.00
Medição 1	14.95	10.00	10.00	285.51	285.50	45.57
Medição 2	14.95	10.00	10.01	285.51	287.40	46.02
Medição 3	14.95	10.00	10.00	287.36	287.50	45.28
Medição 4	15.00	10.00	10.00	287.36	287.50	44.48
Medição 5	14.89	10.00	10.00	287.36	285.50	45.00
Medição 6	14.95	10.00	10.22	288.49	285.40	44.51
Medição 7	14.89	10.00	10.01	285.51	285.50	44.70
Medição 8	14.89	10.00	10.00	285.51	285.50	44.69
Medição 9	14.95	10.00	10.00	287.36	287.50	45.00
Medição 10	14.95	10.00	10.00	287.36	287.50	44.69
Média	14.94	10.00	10.02	286.73	286.48	44.99
Desvio Padrão	0.04	0.00	0.07	1.11	1.05	0.50
Tendência	-0.06	0.00	0.02	-0.77	-1.02	-0.01
Repetibilidade	0.08	0.00	0.16	2.57	2.45	1.16
Erro máximo	0.15	0.00	0.18	3.34	3.47	1.16
Erro máx. esperado	0.25	0.20	0.20	2.98	2.98	0.55

Tabela 12: Resultados das medições na detecção de formas

Forma	Triângulo		Quadrado		Pentágono		Hexágono		Círculo	
	A	P	A	P	A	P	A	P	A	P
Valores nominais	4.00	9.66	9.00	12.00	6.88	10.00	5.20	8.49	12.57	12.57
Medição 1	3.84	9.36	8.71	11.76	6.91	10.52	5.17	8.87	12.48	13.22
Medição 2	3.97	9.50	9.00	11.96	7.14	10.70	5.34	9.02	12.89	13.44
Medição 3	4.00	9.53	9.10	12.03	7.16	10.71	5.40	9.03	12.93	13.46
Medição 4	4.00	9.53	9.00	11.96	7.14	10.70	5.37	9.02	12.91	13.44
Medição 5	3.84	9.34	8.70	11.76	6.91	10.52	5.16	8.87	12.47	13.22
Medição 6	3.87	9.37	8.85	11.86	6.93	10.53	5.22	8.88	12.52	13.25
Medição 7	3.87	9.37	8.75	11.80	6.92	10.52	5.21	8.88	12.50	13.24
Medição 8	3.87	9.37	8.80	11.83	6.93	10.53	5.22	8.88	12.51	13.24
Medição 9	3.84	9.34	8.71	11.76	6.91	10.52	5.12	8.84	12.46	13.19
Medição 10	3.84	9.34	8.71	11.76	6.91	10.52	5.15	8.85	12.46	13.19
Média	3.89	9.41	8.83	11.85	6.99	10.58	5.24	8.91	12.61	13.29
Desvio Padrão	0.07	0.08	0.15	0.10	0.11	0.09	0.10	0.08	0.21	0.11
Tendência	-0.11	-0.25	-0.17	-0.15	0.11	0.58	0.04	0.42	0.04	0.72
Repetibilidade	0.16	0.19	0.34	0.23	0.26	0.20	0.23	0.18	0.48	0.26
Erro máximo	0.26	0.44	0.51	0.39	0.36	0.78	0.27	0.60	0.52	0.98
Erro máx. esperado	0.14	0.20	0.19	0.22	0.17	0.20	0.15	0.18	0.23	0.23
<div style="text-align: right; padding-right: 10px;"> área mm² perímetro mm </div>										

6.1.3 Medições com fator de conversão obtido através do bloco-padrão

Tabela 13: Medições com imagem gerada digitalmente e referenciamento com bloco-padrão

Medida	A	B		Área peça 1	
		Manual	Auto	Contagem	Contorno
Valor Nominal	15.00	10.00	10.00	287.50	287.50
Medição 1	15.08	10.06	10.06	288.43	290.4
Medição 2	15.08	10.16	10.06	288.47	286.5
Medição 3	15.02	10.06	10.06	288.47	286.5
Medição 4	15.02	10.06	10.26	288.47	290.5
Medição 5	15.08	10.16	10.06	288.47	286.4
Medição 6	15.02	10.06	10.16	288.47	286.4
Medição 7	15.08	10.06	10.06	288.47	286.4
Medição 8	15.08	10.06	10.16	290.34	290.4
Medição 9	15.08	9.84	10.06	288.47	286.4
Medição 10	15.08	10.06	10.06	288.47	286.4
Média	15.06	10.06	10.10	288.65	287.63
Desvio Padrão	0.03	0.09	0.07	0.59	1.94
Tendência	0.06	0.06	0.10	1.15	0.13
Repetibilidade	0.07	0.20	0.16	1.38	4.49
Erro máximo	0.13	0.26	0.26	2.53	4.62
Erro máx. esperado	0.25	0.20	0.20	2.98	2.98

Comparando os valores do erro máximo obtido através dos testes com os valores esperados calculados através da equação 23, observa-se que os resultados com mais ocorrências dentro das características de precisão desejadas foram os correspondentes às medições de distância entre pontos e do diâmetro do círculo, enquanto que as medições de área e ângulo apresentaram maiores erros.

É notável que todos os erros obtidos nos testes das medições das formas geométricas foram superiores aos esperados.

Uma das possíveis razões para esse resultado parece ser o fato de que as dimensões dos objetos tendem a variar de forma sensível durante o processo de binarização, dependendo do valor de limiar selecionado, sendo visível o aumento e diminuição de suas fronteiras enquanto se movimenta a barra de seleção do limiar. No caso da medição de áreas e detecção de formas, todos os métodos dependem da binarização da imagem e, portanto, sofrem influências dessa variação.

Na Tabela 11, nota-se um fato curioso nas medições do diâmetro com seleção manual de pontos, onde todos os valores coincidiram com o valor nominal apesar de todas as medições terem sido feitas com pontos diferentes. Esse bom resultado possivelmente se deve à facilidade de marcação dos pontos usando a grade presente na imagem como referência visual.

Comparando os resultados das medições em relação ao método de referenciamento, não é possível observar nenhuma indicação clara que permita concluir qual dos métodos alcança maior exatidão, apresentando resultados muito próximos, com a média das diferenças entre o erro máximo esperado e o obtido de 0,30 mm para o referenciamento manual e de 0,47 mm para o referenciamento com bloco-padrão.

6.2 Testes com fotografias de peças reais

Para os testes com peças reais, foram usados os procedimentos indicados no desenvolvimento, de forma que o maior foco desse tópico será exibir as capturas de tela mostrando partes dos procedimentos e os dados obtidos. Assim como nos testes usando imagens geradas digitalmente, também foram realizadas 10 medições para cada medida avaliada e sendo calculados os mesmos parâmetros para avaliação do erro máximo obtido em relação ao desejado.

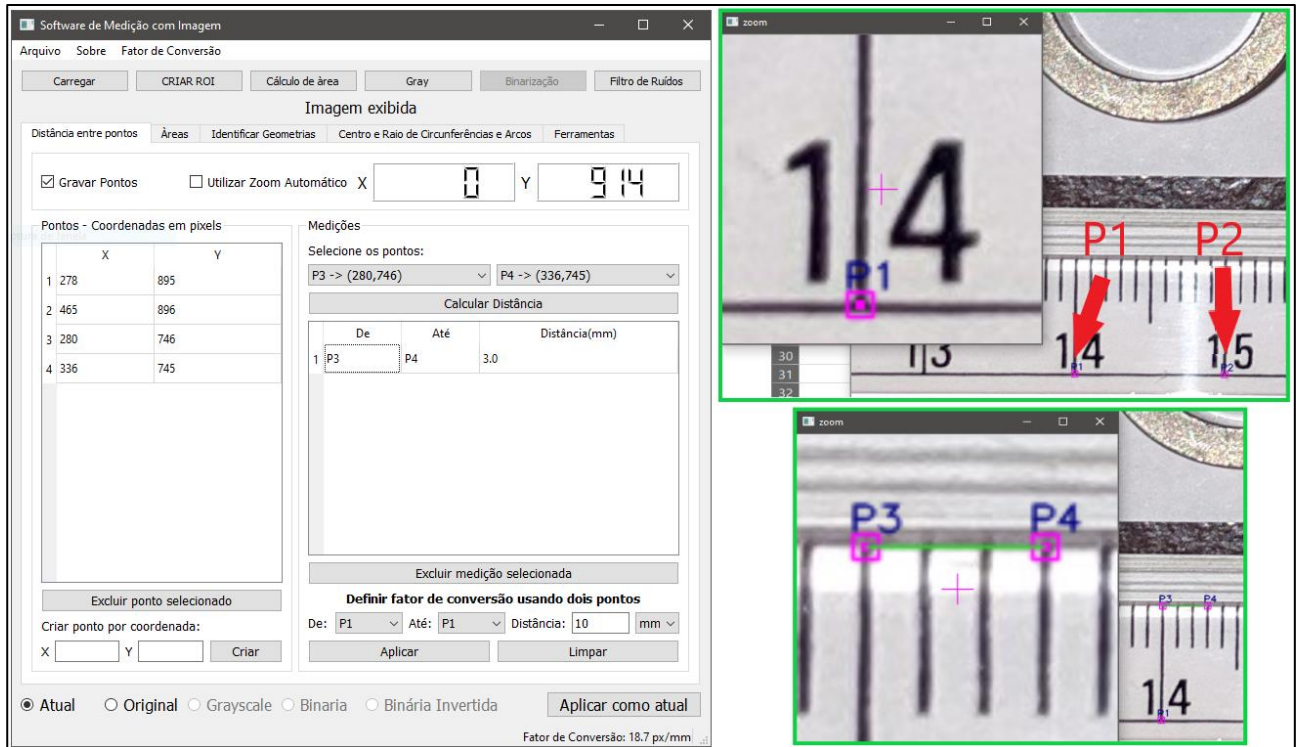
Diferente dos testes com a imagem gerada digitalmente, onde foram testadas todas as funcionalidades com suas variações de método (medição de diâmetro e áreas com os dois métodos disponíveis em cada caso), as medições das peças reais foram feitas utilizando o método que se julgou mais conveniente para cada caso.

Como todos os testes tem em comum o procedimento de calibração com a régua presente na imagem, apenas um exemplo será exibido.

6.2.1 Procedimento padrão utilizado para o referenciamento dimensional

Como padrão, optou-se pela criação dos pontos na região em forma de cruz na divisão dos centímetros da régua. No exemplo (Figura 84), os pontos criados estão localizados nas divisórias que marcam 14 cm e 15 cm na régua; no menu de definição do fator de conversão do programa, os dois pontos foram selecionados, o valor de distância informado foi de 10 mm e a unidade selecionada foi o milímetro, resultando em um fator de conversão de 18,7 px/mm.

Figura 84: Procedimento de referenciamento utilizando pontos marcados sobre uma régua



Fonte: Elaborada pelo autor.

A imagem ainda exibe um teste da calibração feito medindo dois pontos (P3 e P4) criados com distância de 3 mm entre eles sobre a régua, o resultando medido coincidindo com o valor esperado, mostrando que a calibração está correta.

Esse procedimento foi seguido para cada imagem testada, com exceção das medições de ângulo, que dispensam o referenciamento por serem feitas baseadas diretamente nas coordenadas dos pixels.

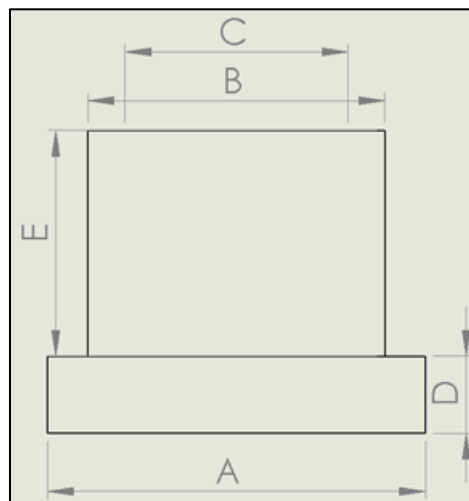
6.2.2 Peças utilizadas nos testes

As peças utilizadas durante os testes reais são exibidas na Figura 85, identificadas por números:

Figura 85: Peças utilizadas nos testes

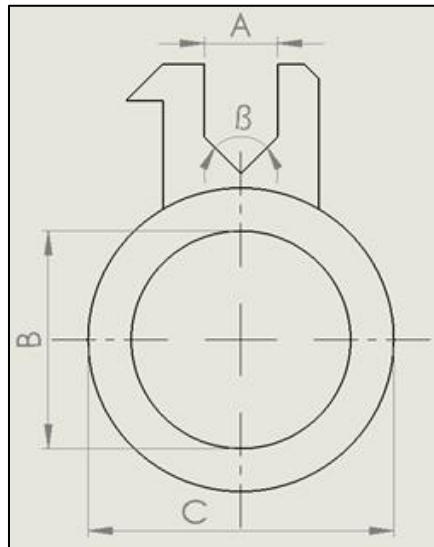
Fonte: Elaborada pelo autor.

A partir das medições das peças feitas com os instrumentos do laboratório de metrologia do IFSC, foram criados desenhos com a representação das medidas alvo:

Figura 86: Desenho da peça 1

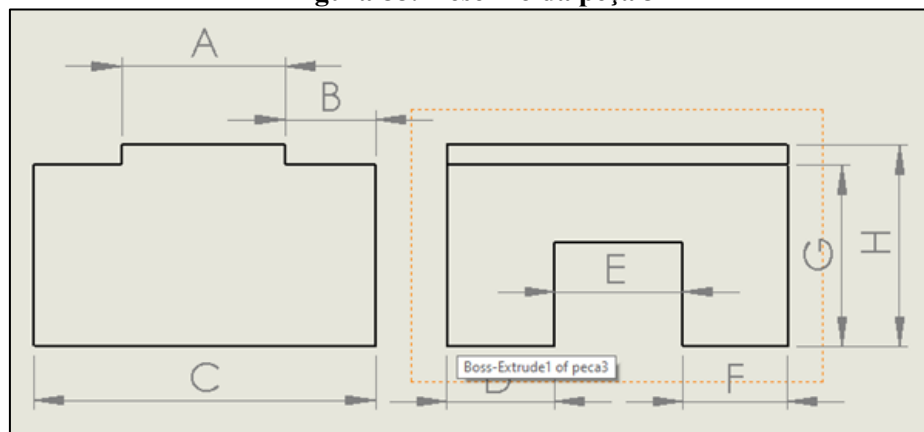
Fonte: Elaborada pelo autor.

Figura 87: Desenho da peça 2



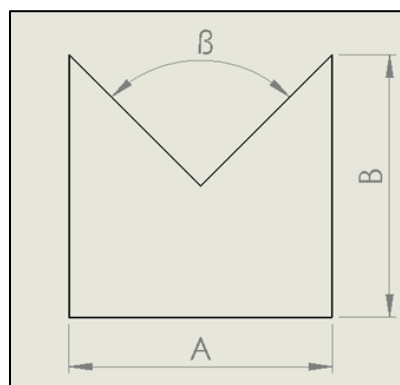
Fonte: Elaborada pelo autor.

Figura 88: Desenho da peça 3



Fonte: Elaborada pelo autor.

Figura 89: Desenho da peça 4



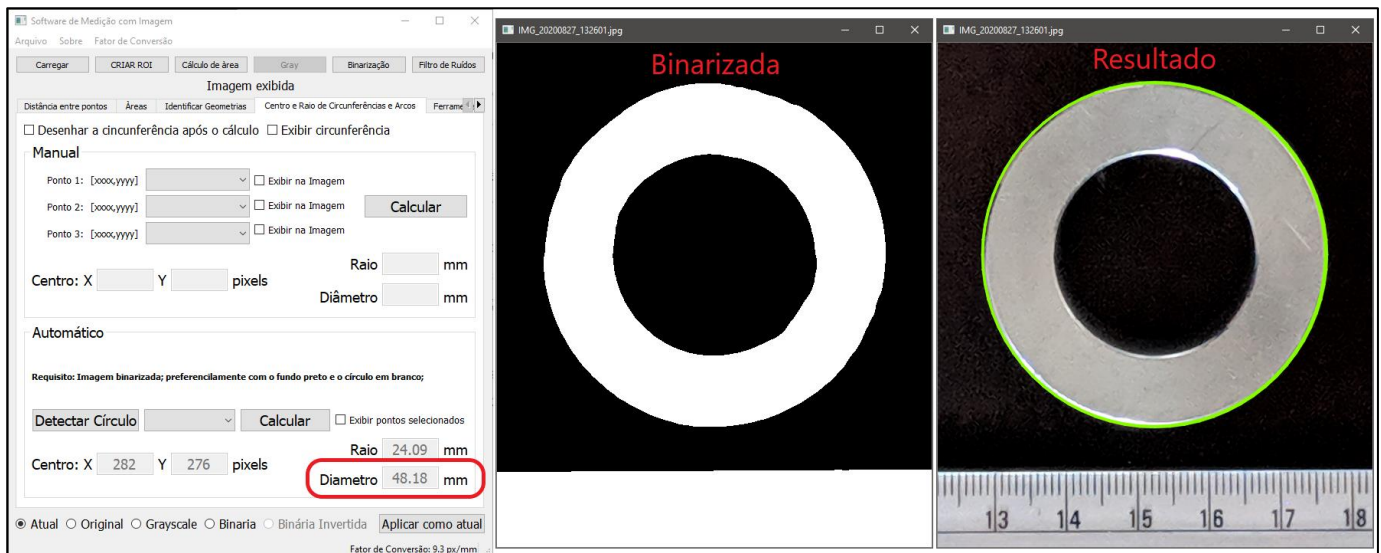
Fonte: Elaborada pelo autor.

6.2.3 Medições na peça 1

a) Medição do diâmetro A

Feita utilizando a seleção automática de pontos do contorno.

Figura 90: Medição do diâmetro A na peça 1

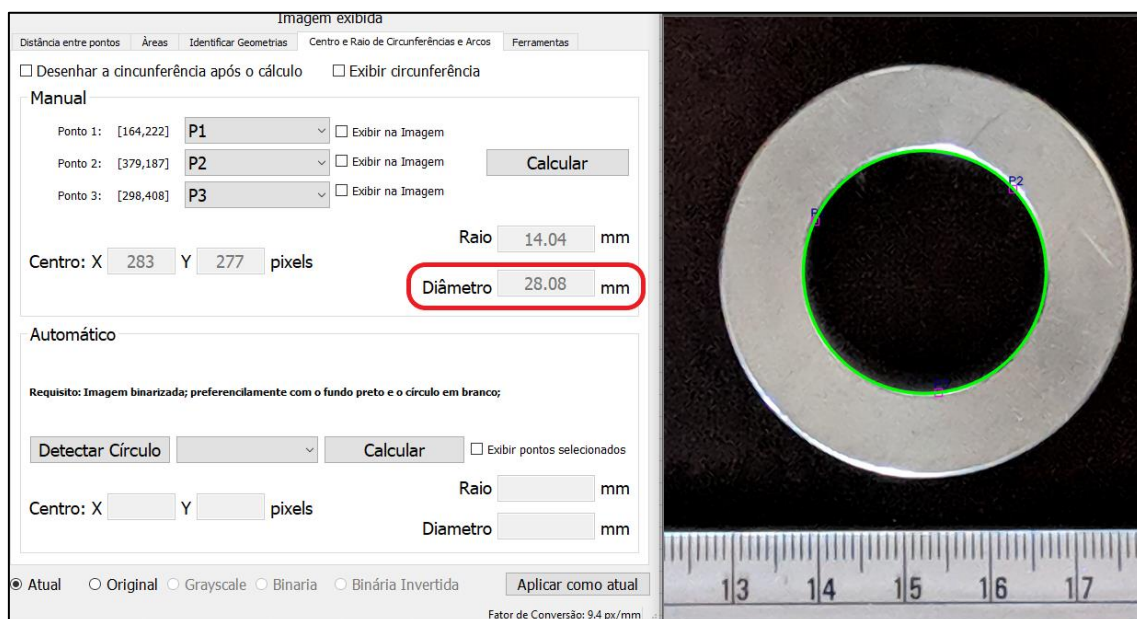


Fonte: Elaborada pelo autor.

b) Medição do diâmetro B

Feita utilizando a criação e seleção manual dos pontos.

Figura 91: Medição do diâmetro B da peça 1

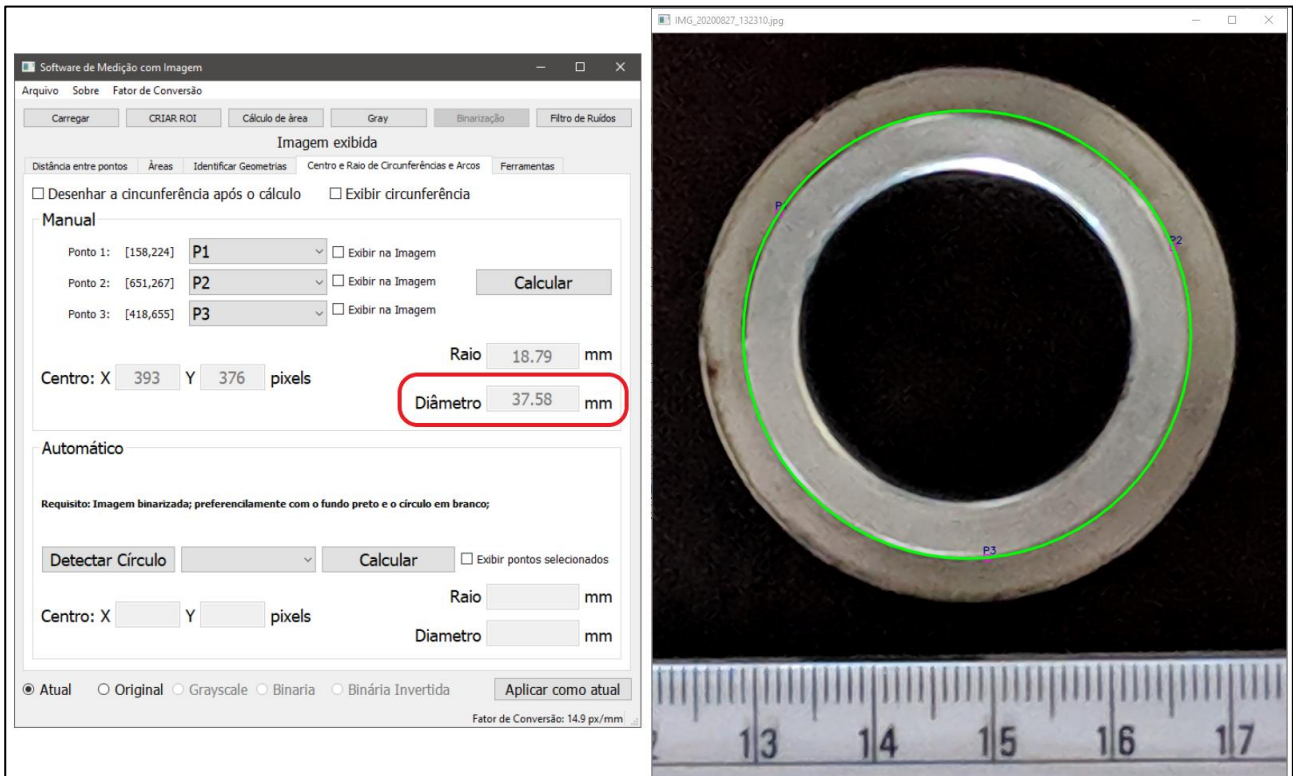


Fonte: Elaborada pelo autor.

c) Medição do diâmetro C

Feita utilizando a criação e seleção manual dos pontos.

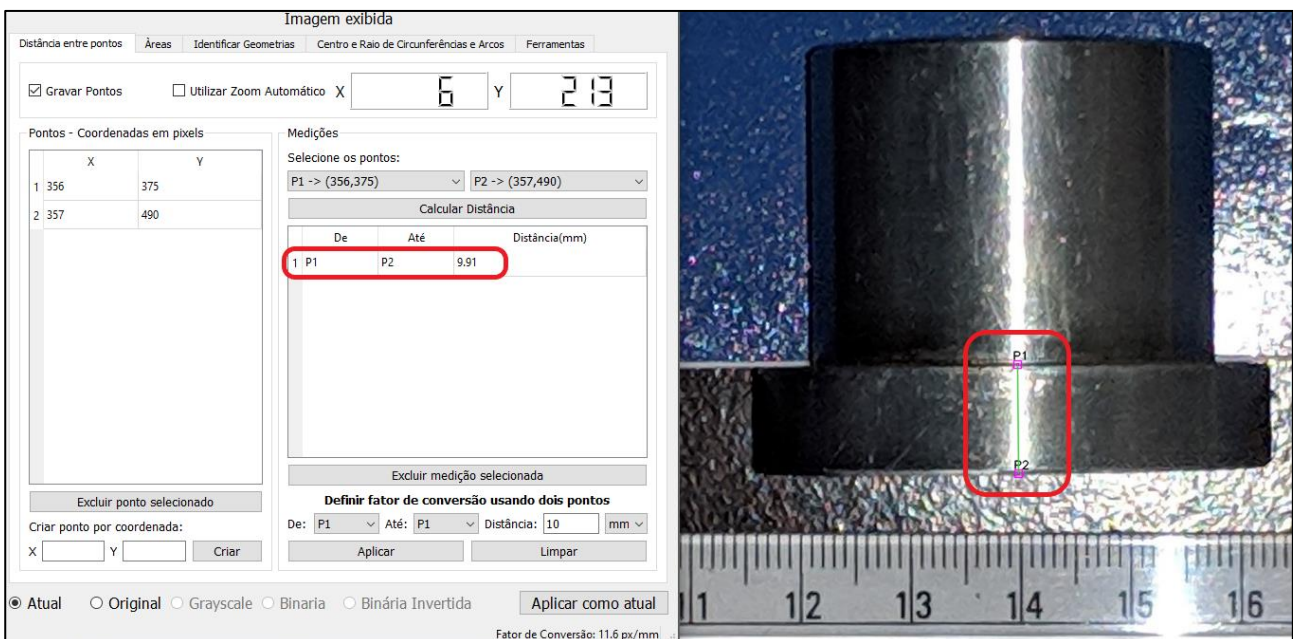
Figura 92: Medição do diâmetro C da peça 1



Fonte: Elaborada pelo autor.

d) Medição da distância D

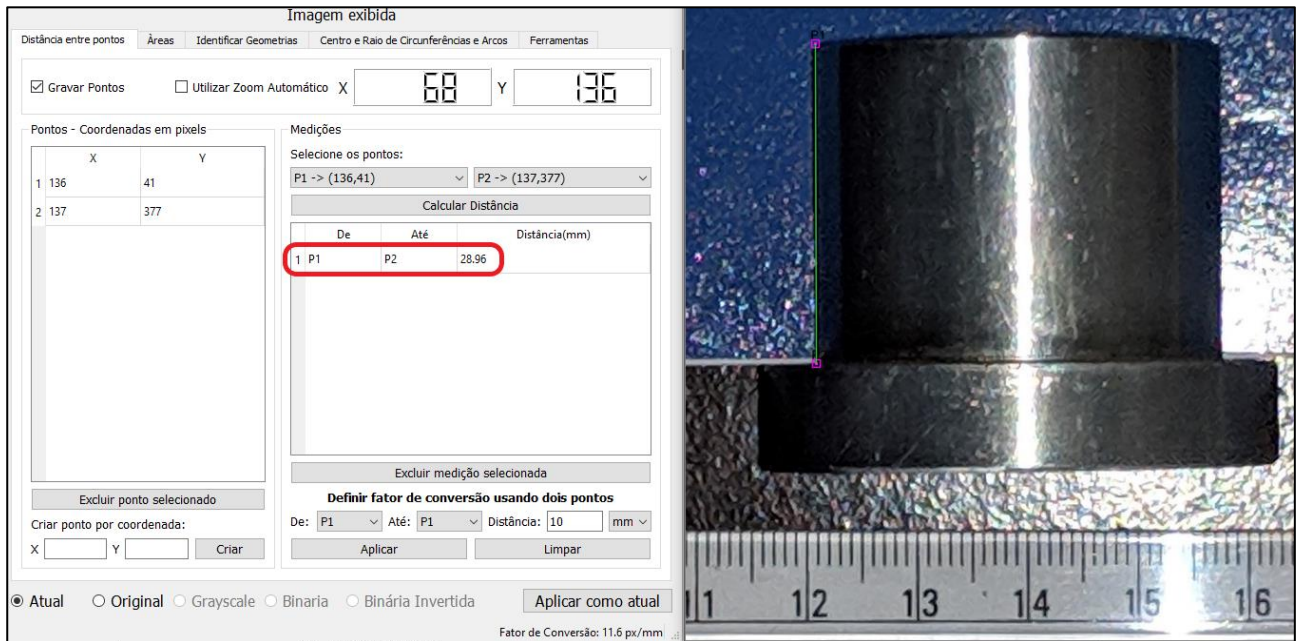
Figura 93: Medida D na peça 1



Fonte: Elaborada pelo autor.

e) Medição da distância E

Figura 94: Medida E da peça 1



Fonte: Elaborada pelo autor.

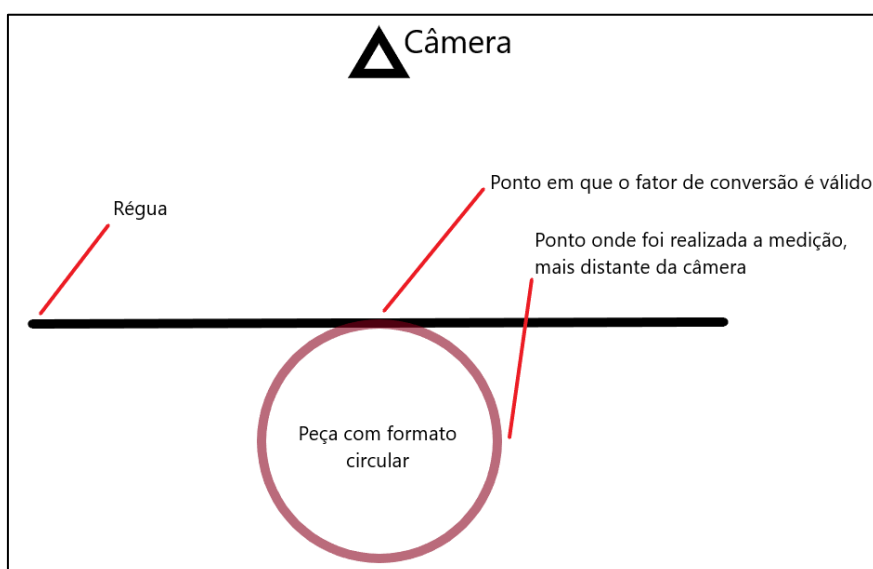
f) Resultados

Após repetir essas medições 10 vezes para cada medida alvo, criou-se a seguinte (Tabela 14) tabela contendo os resultados:

Tabela 14: Resultados das medições da peça 1

Medida	A	B	C	D	E
Valor Referência	47.89	37.62	28.25	9.77	28.73
Medição 1	47.86	37.58	28.08	9.88	29.04
Medição 2	48.18	37.78	27.88	9.63	28.92
Medição 3	47.88	37.56	28.18	9.77	29.34
Medição 4	48.08	37.56	28.07	9.76	28.92
Medição 5	47.88	37.92	28.18	9.88	28.92
Medição 6	48.08	37.96	27.88	9.64	29.16
Medição 7	47.89	37.20	28.08	9.76	29.19
Medição 8	47.86	37.00	27.62	9.96	29.28
Medição 9	47.66	37.38	27.88	9.92	29.03
Medição 10	47.78	37.58	28.08	9.75	29.88
Média	47.92	37.55	27.99	9.80	29.17
Desvio Padrão	0.16	0.30	0.18	0.11	0.29
Tendência	0.02	-0.07	-0.26	0.03	0.44
Repetibilidade	0.36	0.70	0.41	0.26	0.68
Erro máximo	0.39	0.77	0.66	0.29	1.12
Erro máx. esperado	0.58	0.48	0.38	0.20	0.39

Na avaliação das medições dessa peça é interessante o fato do pior resultado ser o relativo à medida E, pois era possível prever esse comportamento levando em consideração o efeito ilustrado a seguir (Figura 95):

Figura 95: Erro ao medir um cilindro deitado

Fonte: Elaborada pelo autor.

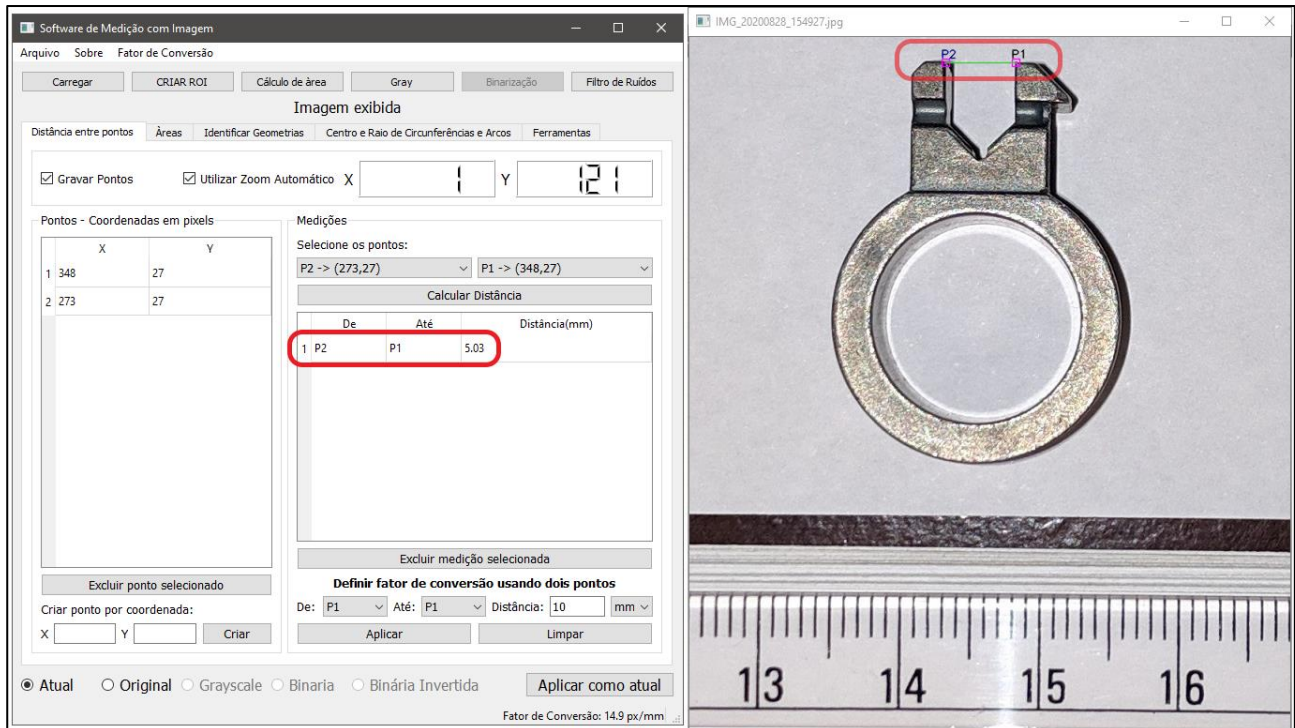
Como a peça é de forma cilíndrica e a régua só pode ser posicionada em uma certa altura em relação ao perfil circular do cilindro deitado, ocorre que o fator de conversão só é preciso na altura do cilindro que coincide com a posição da régua. No caso da fotografia que gerou essa medição, a régua estava posicionada tangente à parte cilíndrica mais longa da peça, ponto onde teria sido mais

correto realizar a medição.

6.2.4 Medições na peça 2

a) Medição de A

Figura 96: Medida do espaçamento A na peça 2



Fonte: Elaborada pelo autor.

b) Medição do diâmetro B

Figura 97: Medida do diâmetro B na peça 2



Distância entre pontos Áreas Identificar Geometrias Centro e Raio de Circunferências e Arcos Ferramentas

Desenhar a circunferência após o cálculo Exibir circunferência

Manual

Ponto 1: [210,241] **P1** Exibir na Imagem

Ponto 2: [404,245] **P2** Exibir na Imagem **Calcular**

Ponto 3: [300,412] **P3** Exibir na Imagem

Centro: X Y pixels Raio mm

Diâmetro mm

Automático

Requisito: Imagem binarizada; preferencialmente com o fundo preto e o círculo em branco;

Detectar Círculo **Calcular** Exibir pontos selecionados

Centro: X Y pixels Raio mm

Diâmetro mm

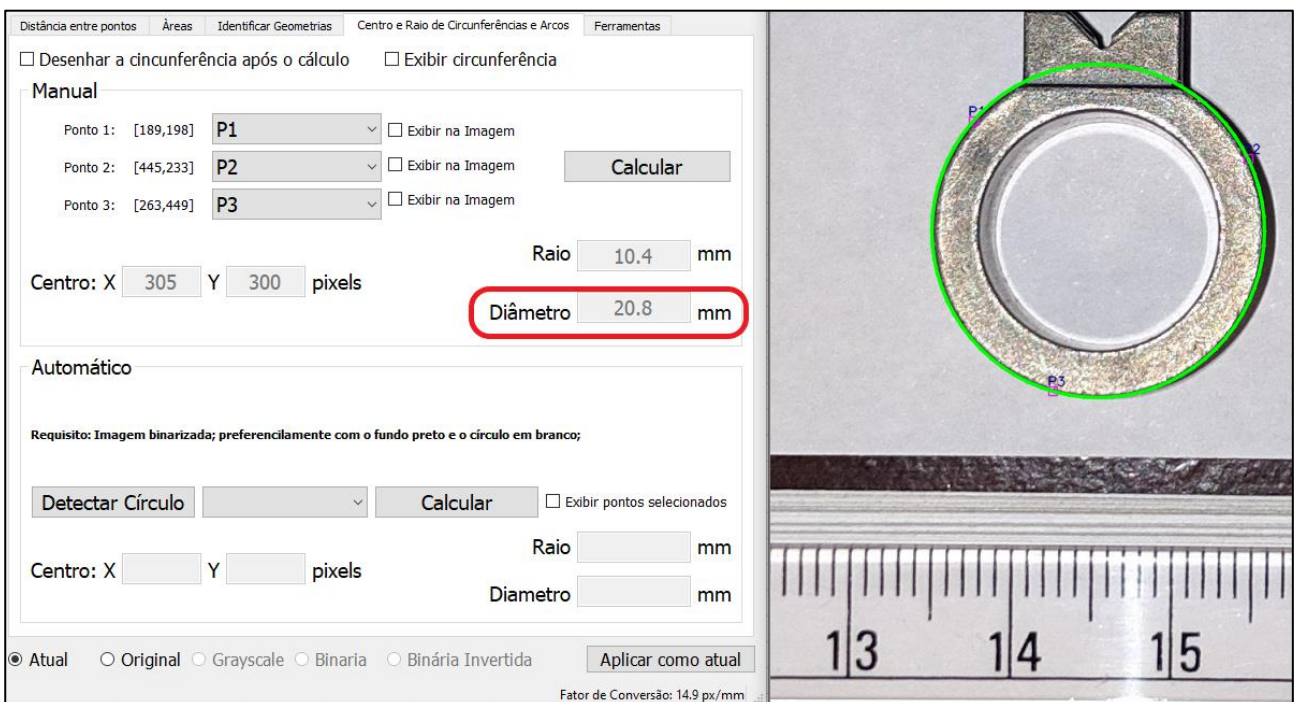
Atual Original Grayscale Binária Binária Invertida **Aplicar como atual**

Fator de Conversão: 14.9 px/mm

Fonte: Elaborada pelo autor.

c) Medição do diâmetro C

Figura 98: Medição do diâmetro C na peça 2



Distância entre pontos Áreas Identificar Geometrias Centro e Raio de Circunferências e Arcos Ferramentas

Desenhar a circunferência após o cálculo Exibir circunferência

Manual

Ponto 1: [189,198] **P1** Exibir na Imagem

Ponto 2: [445,233] **P2** Exibir na Imagem **Calcular**

Ponto 3: [263,449] **P3** Exibir na Imagem

Centro: X Y pixels Raio mm

Diâmetro mm

Automático

Requisito: Imagem binarizada; preferencialmente com o fundo preto e o círculo em branco;

Detectar Círculo **Calcular** Exibir pontos selecionados

Centro: X Y pixels Raio mm

Diâmetro mm

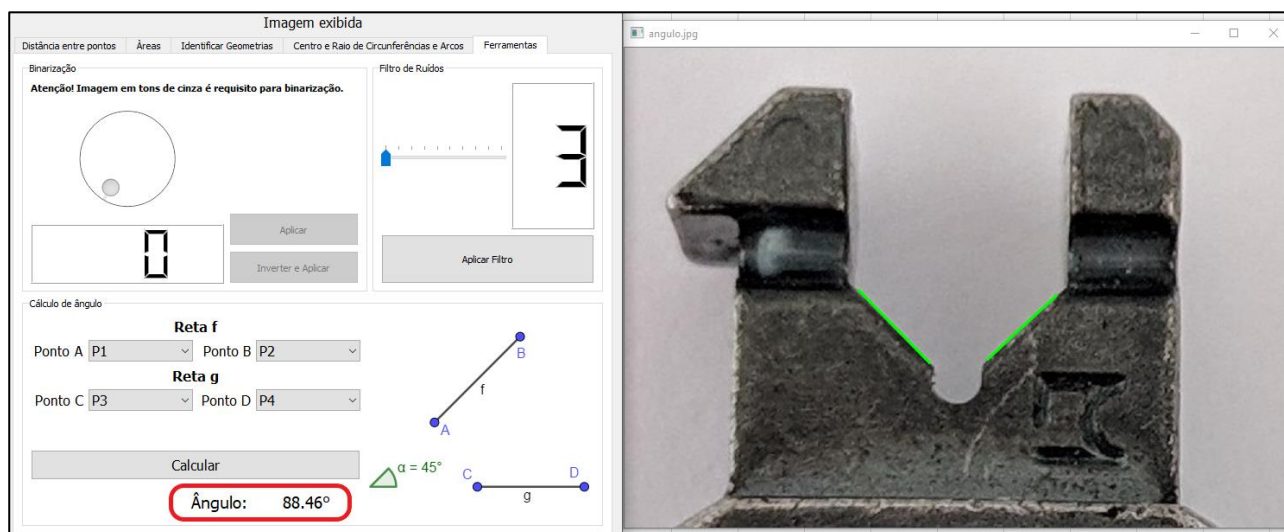
Atual Original Grayscale Binária Binária Invertida **Aplicar como atual**

Fator de Conversão: 14.9 px/mm

Fonte: Elaborada pelo autor.

d) Medição do ângulo β

Figura 99: Medição do ângulo β na peça 2



Fonte: Elaborada pelo autor.

e) Resultados

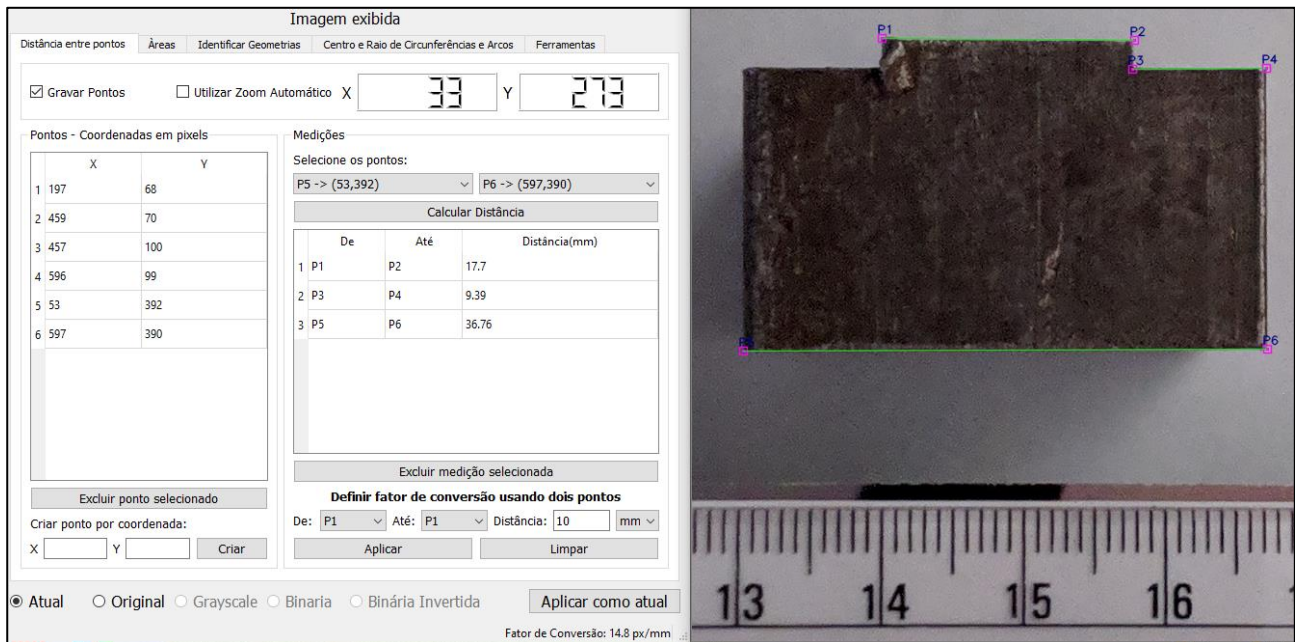
Tabela 15: Resultados das medições na peça 2

Medida	A	B	C	β
Valor Referência	5.00	15.02	20.94	90.00
Medição 1	5.05	14.90	20.64	89.69
Medição 2	5.03	14.88	20.80	88.52
Medição 3	5.05	15.06	20.86	88.61
Medição 4	5.06	15.05	20.86	88.34
Medição 5	5.05	15.04	20.68	88.46
Medição 6	5.00	15.12	20.64	87.66
Medição 7	5.09	15.06	20.86	87.95
Medição 8	5.11	14.92	20.86	87.87
Medição 9	5.00	14.96	20.88	90.07
Medição 10	5.06	14.88	20.65	87.81
Média	5.05	14.99	20.77	88.50
Desvio Padrão	0.03	0.09	0.11	0.80
Tendência	0.05	-0.03	-0.17	-1.50
Repetibilidade	0.08	-2.23	0.25	1.86
Erro máximo	0.13	0.24	0.41	3.36
Erro máx. esperado	0.15	0.25	0.31	1.00

6.2.5 Medições na peça 3

a) Medições A, B e C

Figura 100: Medições A, B e C na peça 3



Fonte: Elaborada pelo autor.

b) Medições D, E, F e G

Figura 101: Medições D, E, F e G na peça 3



Fonte: Elaborada pelo autor.

c) Medição de H

A medição de H foi feita de maneira indireta, como a soma das distâncias entre P2 e P3 e P4

e P6, pontos mostrados na Figura 100.

d) Resultados

Tabela 16: Resultados das medições na peça 3

Medida	A	B	C	D	E	F	G	H
Valor Referência	18.10	10.05	37.94	11.96	14.22	11.63	20.11	22.30
Medição 1	17.61	9.46	36.90	11.79	14.03	11.25	19.63	21.64
Medição 2	17.72	9.46	36.96	11.84	14.13	11.31	19.57	21.52
Medição 3	17.62	9.43	36.86	12.00	14.03	11.41	20.05	22.01
Medição 4	17.64	9.39	36.85	11.91	14.09	11.28	19.52	21.43
Medição 5	17.66	9.35	36.90	11.86	14.10	11.17	19.52	21.53
Medição 6	17.66	9.46	36.85	11.81	13.98	11.32	19.60	21.61
Medição 7	17.70	9.41	36.76	12.00	14.04	11.33	19.57	21.52
Medição 8	17.54	9.42	36.65	11.76	14.07	11.33	19.57	21.52
Medição 9	17.66	9.40	36.85	11.79	13.96	11.31	19.49	21.50
Medição 10	17.56	9.43	36.75	11.84	13.91	11.26	19.49	21.44
Média	17.64	9.42	36.83	11.86	14.03	11.30	19.60	21.57
Desvio Padrão	0.06	0.04	0.09	0.08	0.07	0.06	0.16	0.17
Tendência	-0.46	-0.63	-1.11	-0.10	-0.19	-0.33	-0.51	-0.73
Repetibilidade	0.13	0.08	0.21	0.20	0.16	0.15	0.38	0.39
Erro máximo	0.59	0.71	1.32	0.30	0.34	0.48	0.89	1.12
Erro máx. esperado	0.28	0.20	0.48	0.22	0.24	0.22	0.30	0.32

6.2.6 Medições na peça 4

a) Medições A e B

Figura 102: Medições de A e B na peça 4

Imagem exibida

Distância entre pontos Áreas Identificar Geometrias Centro e Raio de Circunferências e Arcos Ferramentas

Gravar Pontos Utilizar Zoom Automático X: 0 Y: 241

Pontos - Coordenadas em pixels

	X	Y
1	121	372
2	462	375
3	461	27

Medições

Selecione os pontos:
P2 -> (462,375) P3 -> (461,27)

Calcular Distância

	De	Até	Distância(mm)
1	P1	P2	18.43
2	P2	P3	18.81

Excluir medição selecionada

Definir fator de conversão usando dois pontos

De: P1 Até: P1 Distância: 10 mm

Aplicar Limpar

Atual Original Grayscale Binaria Binária Invertida Aplicar como atual

Fator de Conversão: 18,5 px/mm

Fonte: Elaborada pelo autor.

b) Medição do ângulo β

Figura 103: Medição do ângulo β na peça 4

Imagem exibida

Distância entre pontos Áreas Identificar Geometrias Centro e Raio de Circunferências e Arcos Ferramentas

Binarização

Atenção! Imagem em tons de cinza é requisito para binarização.

Filtro de Ruídos

Cálculo de ângulo

Reta f
Ponto A: P1 Ponto B: P2

Reta g
Ponto C: P3 Ponto D: P4

Calcular

Ângulo: 87.79°

$\alpha = 45^\circ$

Fonte: Elaborada pelo autor.

c) Resultados

Tabela 17: Resultados das medições na peça 4

Medida	A	B	β
Valor Referência	19.03	18.96	90
Medição 1	18.59	18.78	88.1
Medição 2	18.38	18.78	88.27
Medição 3	18.51	18.83	88.73
Medição 4	18.46	18.83	88.28
Medição 5	18.43	18.82	88.31
Medição 6	18.42	18.81	87.79
Medição 7	18.46	18.87	88.28
Medição 8	18.42	18.84	87.88
Medição 9	18.46	18.92	88.24
Medição 10	18.42	18.87	88.11
Média	18.46	18.84	88.20
Desvio Padrão	0.06	0.04	0.26
Tendência	-0.57	-0.13	-1.80
Repetibilidade	0.14	0.10	0.60
Erro máximo	0.71	0.22	2.40
Erro máx. esperado	0.29	0.29	1.00

Avaliando os resultados do erro máximo obtido nas medições em relação ao erro máximo esperado calculado via equação 23, verificou-se que apenas 9 das 41 medidas testadas foram aprovadas em relação ao requisito de exatidão estabelecido.

6.2.7 Teste do tempo de medição

O objetivo desse teste é avaliar o tempo médio necessário para realização do processo completo de medição, desde a aquisição da imagem até a visualização do resultado, utilizando o programa desenvolvido.

Como as imagens utilizadas nos testes foram adquiridas através da câmera de um celular e só depois transferidas para o computador, o que é muito diferente do uso ideal, foi utilizado um cenário imaginário, porém realista, para estimar o tempo de aquisição da imagem.

O tempo de aquisição da imagem foi estimado com base em um cenário em que haja um aparato com uma câmera diretamente ligada ao computador e um local preparado para o posicionamento da peça. Considerando o tempo de posicionamento da peça, do posicionamento do padrão de referência e do comando de fotografar, o tempo de aquisição estimado é de 25 segundos.

Em termos do processo de medição utilizando o software, foram escolhidos dois cenários de teste, um com grande dependência do usuário e outro com maior automatização, conforme descritos abaixo:

- Cenário 1: Medição do diâmetro A da peça 1 (Figura 86), calibrando com marcação manual dos pontos sobre a régua e com medição baseada também em pontos criados manualmente na borda do cilindro.
- Cenário 2: Medição do diâmetro do círculo contido na imagem gerada digitalmente (Figura 70), calibrando com detecção automática do bloco padrão e medindo com seleção automática dos pontos do contorno.

Apesar de existirem questões subjetivas na velocidade com que são feitos os procedimentos de medição, foi feito um esforço para que os testes fossem feitos considerando um usuário bem habituado à interface do software, mas realizando os procedimentos sem pressa. Foram feitas cinco medições em cada cenário com o tempo cronometrado desde a seleção e carregamento do arquivo da imagem, até a exibição do resultado da medição na tela. Essas informações foram registradas e usadas para calcular o tempo médio para cada situação testada, conforme mostrado na tabela a seguir:

Tabela 18: Tempos de medição para cenários propostos

	Tempo (s) - Cenário 1	Tempo (s) - Cenário 2
Medição 1	35.41	21.34
Medição 2	31.55	17.31
Medição 3	25.34	18.69
Medição 4	29.72	18.55
Medição 5	31.48	23.13
Tempo médio	30.7	19.80
Média Global	25.25	

Observa-se que a média do tempo medição para o cenário 2, em que a maior parte dos procedimentos são feitos automaticamente, foi aproximadamente 11 segundos mais rápido do que o cenário com procedimentos predominantemente manuais, conforme esperado.

Somando a média global de tempo de medição obtida nos testes com o tempo de aquisição estimado, foi obtido um tempo de medição total médio de 50,25 segundos.

7. CONCLUSÕES

Conforme proposto, esse trabalho apresentou um estudo de ferramentas da visão computacional aplicadas à metrologia dimensional, partindo do processo de aquisição de imagens digitais, pesquisando métodos de processamento de imagens, referenciamento dimensional e estudando possíveis fontes de erro de medição inerentes ao processo.

Foram apresentadas as etapas de desenvolvimento do software de medição, desde sua concepção, mostrando a metodologia utilizada, os princípios da geometria que basearam os processamentos matemáticos sobre os pixels, necessários para efetuar as medições, e os algoritmos criados para implementar esses processamentos no software. Na parte final do trabalho, foram apresentados testes práticos com medições feitas utilizando o software criado, a avaliação dos resultados.

Em relação aos requisitos estabelecidos na concepção do projeto, os erros percentuais se mostraram maiores do que o esperado na maioria dos testes e os tempos de medição ficaram dentro do valor esperado. Como não foram observadas grandes variações na precisão das medições quando se compara os dois métodos de referenciamento implementados e o tempo de medição se mostrou menor com o uso do bloco-padrão, concluiu-se que o uso da aplicação com referenciamento por bloco-padrão é mais indicado.

Apesar de alguns resultados terem ficado com erros acima do planejado, os algoritmos implementados se mostraram eficientes em termos de cálculo e desempenho.

Analisando os objetivos gerais do trabalho, pode-se afirmar que foram cumpridos, pois tanto as pesquisas e a consolidação de conhecimentos propostas, quanto o desenvolvimento do software cumprindo todas as funcionalidades idealizadas foram concluídos.

Uma melhoria proposta no sistema seria a inclusão de um sistema de redimensionamento da imagem durante o uso do programa, que permita o uso de imagens de maior resolução, pois o redimensionamento implementado atualmente reduz a resolução da imagem para que caiba totalmente dentro do campo de visão da tela utilizada, e essa perda de resolução tem impacto negativo na precisão das medições. A implementação dessa melhoria traria o desafio adicional de mudar dinamicamente o fator de conversão para se adequar à nova relação de pixels da imagem redimensionada.

REFERÊNCIAS

CHASTAIN, Sue. **Vector vs. Bitmap Images**: what are vector and bitmap images, and how are they different? 2020. Disponível em: <https://www.lifewire.com/vector-and-bitmap-images-1701238>. Acesso em: 05 abr. 2020.

ROZZO, Fernando. **AS LENTES E SUAS CARACTERÍSTICAS – “A DISTANCIA FOCAL”**. 2020. Disponível em: <https://blog.emania.com.br/as-lentes-e-suas-caracteristicas-a-distancia-focal/>. Acesso em: 6 abr. 2020.

ESCANDAR, Nizar. **Diafragma e obturador, os olhos da câmera**. 2020. Disponível em: <https://blog.emania.com.br/diafragma-e-obturador-os-olhos-da-camera/>. Acesso em: 6 abr. 2020.

NORTHRUP, Anthony. **How cameras see color: RGB vs RGBW, Bayer vs Fuji X-Trans, Purple isn't real?!** 2019. Disponível em: https://youtu.be/ss8_qRkwG5I. Acesso em: 10 maio 2020.

BOHRER, Márcio. **Como funciona uma câmera digital?** 2016. Disponível em: <https://www.oficinadanet.com.br/post/12125-como-funciona-uma-camera-digital>. Acesso em: 15 maio 2020.

WELLER, Hannah. **Color Spaces**. 2018. Disponível em: <https://hiweller.github.io/colordistance/color-spaces.html>. Acesso em: 15 maio 2020.

GRAYSCALE to RGB Conversion. Disponível em: https://www.tutorialspoint.com/dip/grayscale_to_rgb_conversion.htm. Acesso em: 20 maio 2020.

Torgashov, Pavel. **Contour Analysis for Image Recognition in C#**. 2014. Disponível em: <https://www.codeproject.com/Articles/196168/Contour-Analysis-for-Image-Recognition-in-C>. Acesso em: 20 maio 2020.

G. JUNIOR, Armando Albertazzi; SOUSA, André R. de. **Fundamentos da metrologia científica e industrial**. Florianópolis: Monole, 2008.

MORDVINTSEV, Alexander. **OpenCV-Python Tutorials**. 2013. Disponível em: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html. Acesso em: 06 jun. 2020.

Qt for Python Documentation. 2020. Disponível em: <https://doc.qt.io/qtforpython/contents.html>. Acesso em: 06 jun. 2020.

SCRUM. 2013/2014. Disponível em: <https://www.desenvolvimentoagil.com.br/scrum/>. Acesso em: 10 maio 2020.

Distorção Radial em Fotografias. 2015. Disponível em: <http://cameraneon.com/tecnicas/distorcao-radial-em-fotografias/>. Acesso em: 15 jul. 2020.

PALIGA, Aline. **Somas de Riemann e Integração Numérica**. 2012. Disponível em: https://wp.ufpel.edu.br/nucleomateceng/files/2012/07/Somas-de-Riemann-e-integracao-numerica_.pdf. Acesso em: 15 ago. 2020.