

Universidade Federal do Rio de Janeiro

Escola Politécnica

Departamento de Eletrônica e de Computação

Análise de Dutos Curvos Utilizando Processamento de Imagens

Autor:

Diego Santos Wanderley

Orientadora:

Prof. Mariane Rembold Petraglia, Ph.D.

Orientador:

Prof. José Gabriel Rodriguez Carneiro Gomes, Ph.D.

Examinador:

Prof. Julio Cesar Boscher Torres, D.Sc.

Examinador:

Prof. Gelson Vieira Mendonça, Ph.D.

DEL

Agosto de 2010

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Escola Politécnica - Departamento de Eletrônica e de Computação

Centro de Tecnologia, bloco H, sala H-212-A, Cidade Universitária

Rio de Janeiro - RJ CEP 21949-900

Este exemplar é de propriedade da Universidade Federal do Rio de Janeiro, que poderá incluí-lo em base de dados, armazenar em computador, microfilmear ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

DEDICATÓRIA

Aos meus pais.

AGRADECIMENTO

Aos meus pais, pelo apoio incondicional que permitiu a realização desta conquista.

Aos meus professores e orientadores, Mariane Rembold Petraglia e José Gabriel Rodriguez Carneiro Gomes, pelos ensinamentos e conselhos, além da paciência e disponibilidade em me orientar na Iniciação Científica e neste projeto.

Aos professores do Departamento de Engenharia Eletrônica e de Computação da UFRJ pela formação de alto nível. Em especial ao professor Joarez Bastos Monteiro, de quem tive o prazer de ser monitor na disciplina Eletrônica III.

Aos grandes amigos que fiz durante a graduação e que em determinados momentos não mediram esforços para que tudo acabasse bem. Como Daniel Cayres e Carlos Caldas durante a finalização do nosso Projeto Integrado, Gustavo Alves às vésperas das provas de Sistemas Lineares II e da entrega do PI. Além de Vinícius Ferrão com quem sempre pude contar nas matérias de computação, e Física II.

Aos amigos e colegas do Laboratório de Processamento Analógico de Digital de Sinais com os quais tive a felicidade de conviver, aprender, trabalhar e estudar junto. Com menção especial a Trevor Dobbin e Edmundo Hoyle que colaboraram com este projeto.

RESUMO

Este trabalho utiliza técnicas de Processamento Digital de Sinais e Imagens para localizar, em uma fotografia, tubulações e identificar possíveis curvaturas destas, com a finalidade de realizar uma inspeção automática.

A análise e a inspeção da qualidade de tubulações são atividades extremamente importantes para a prevenção de diversos acidentes, como a explosão de um gasoduto ou o vazamento de óleo em uma baía, os quais podem causar problemas instantâneos e futuros para o meio ambiente e a população local, além de grandes prejuízos para a empresa responsável pela tubulação.

Tubulações são estruturas longas localizadas muitas vezes em regiões isoladas e de difícil acesso. Por este motivo muitas vezes a inspeção é feita através de câmeras de vídeo, que enviam o sinal para estações base onde técnicos ficam responsáveis por analisar as imagens. Muitas vezes a inspeção humana está sujeita a falhas, portanto, um sistema de detecção automática de curvaturas de dutos pode ser de grande auxílio.

Inicialmente, aplicamos um detector de bordas a cada imagem a ser analisada, visando isolar as tubulações das demais componentes da imagem, e nos permitindo extrair informações de interesse para serem processadas. O algoritmo empregado foi o detector de bordas de Canny, gerando imagens binárias com bordas de um pixel de espessura.

Após isolar os segmentos da imagem, selecionamos os mais propícios a serem as bordas da tubulação. Em seguida as variações nos segmentos são computadas e armazenadas em um histograma, que posteriormente será comparado com histogramas em uma base de dados. O objetivo é identificar se existem alguma torção em vez de um duto reto.

Palavras-Chave: Processamento de Imagens, Visão Computacional, Reconhecimento de Padrões, Detecção de Bordas, Inspeção Automática.

ABSTRACT

This work uses techniques of Digital Signal and Image Processing to locate, in a photography, pipes and identify possible curvatures in them, with the purpose of carrying out an automatic inspection.

Analysis and inspection of pipes quality are extremely important for preventing several accidents, like the explosion of a pipeline or the oil spill in a bay, which can generate problems immediately and to the future of the environment and the local population, besides large losses for the company responsible for the pipe.

Pipelines are, in general, long structures, often located in isolated and inaccessible locations. For this reason, the inspection is often done via video cameras that send their signals to a base station where technicians are responsible for reviewing their contents. Often human inspection is subject to failure, therefore, a system for detecting curvatures in pipes automatically can be of a great help.

Initially, we apply an edge detector to each image for isolating the pipes from other parts of the image, allowing us to extract information of interest for the processing. For such, the employed algorithm was the Canny Edge Detector, which generated a binary image with edge segments of just one pixel of thickness.

After obtaining the edge segments of an image, the system selects the ones most likely to be the edges of the pipe. Then the slopes in all pixels of each selected segment are computed and stored in a histogram, which will be compared later with the histograms of a database. The ultimate goal is to identify any kind of twist along the pipe.

Key-words: Image Processing, Computer Vision, Pattern Recognition, Edge Detection, Auto Inspection.

SIGLAS

C.I.E. - Comissão Internacional de Iluminação (Commission internationale de l'éclairage)

DFT - Discret Fourier Transform

FFT - Fast Fourier Transform

SNR - Razão Sinal-Ruído (Signal to Noise Ratio)

UFRJ - Universidade Federal do Rio de Janeiro

Sumário

Lista de Figuras	x
Lista de Tabelas	xii
1 Introdução	1
1.1 Tema	1
1.2 Delimitação	1
1.3 Justificativa	2
1.4 Objetivos	2
1.5 Metodologia	3
1.6 Descrição	4
2 Análise de Imagens Digitais	5
2.1 Luz e Luminância	6
2.2 Representação de Imagens Digitais	9
2.3 Filtros de Imagens	10
2.3.1 Filtros no Domínio Espacial	11
2.3.2 Filtros no Domínio da Frequência	12
2.4 Segmentação	13
2.4.1 Detecção de Pontos	14
2.4.2 Detecção de Linhas	14
2.4.3 Detecção de Bordas	15
3 Algoritmo	23
3.1 Descrição Geral	23

3.2	Detector de Bordas de Canny	24
3.2.1	Filtragem Gaussiana	25
3.2.2	Supressão Não-Máxima	27
3.2.3	Limiars com Histereses	28
3.3	Separando Segmentos	30
3.3.1	Segmentos Fechados	34
3.4	Inclinações dos Segmentos	35
3.5	Comparação de Segmentos	37
3.5.1	DFT dos Segmentos	39
3.5.2	Comparação das DFTs	41
4	Testes e Resultados	43
4.1	Classes dos Dutos Analisados	43
4.2	Parâmetros Utilizados	45
4.3	Composição da Base de Dados	46
4.4	Comparação com Novos Segmentos	48
4.4.1	Primeira Comparação	49
4.4.2	Segunda Comparação	50
5	Conclusão	53
	Bibliografia	55

Lista de Figuras

2.1	Imagem colorida e sua luminância	7
2.2	Imagens com as luminâncias de R, G e B	8
2.3	Representação de uma Imagem Digital	10
2.4	Máscara 3×3 com seus pesos w	11
2.5	Máscara de um filtro passa-altas	14
2.6	Máscaras para a detecção de linhas e retas	15
2.7	Detecção de Bordas por Operadores de Derivação	16
2.8	Região 3×3 de uma imagem	18
2.9	Operadores de Gradiente de Roberts	19
2.10	Detecção de Bordas pelo Operador de Sobel	21
2.11	Máscara para o cálculo do Laplaciano	22
3.1	Bordas de uma Imagem	25
3.2	Máscara 5×5 de um Filtro Gaussiano	26
3.3	Imagem Filtrada por um filtro Gaussiano	27
3.4	Exemplo de Supressão Não-Máxima	28
3.5	Comparação entre limiares do Detector de Bordas de Canny	29
3.6	Imagem com Bordas Sobrepostas	30
3.7	Pixels de um Segmento	31
3.8	Vizinhança de um pixel	33
3.9	Inclinações em um Segmento	36
3.10	Histograma com as inclinações da curva da Figura 3.9	37
3.11	Inclinações em um segmento com rotação de 90°	39
3.12	Módulo da DFT do Histograma da Figura 3.10	41

4.1	Classes consideradas	44
4.2	Imagens de Loop segmentada com os melhores parâmetros de entrada . .	46
4.3	Imagens de Torção segmentada com os melhores parâmetros de entrada .	46
4.4	Loop classificado como Torção	52
4.5	Torção classificado como Corcova	52
4.6	Duto Reto classificado como Corcova	52

Lista de Tabelas

2.1	Máscaras dos Gradientes	20
3.1	Coordenadas correspondentes aos pixels do segmento da Figura 3.7	32
4.1	Parâmetros do Detector de Borda de Canny	46
4.2	Organização da Matriz com as DFTs	47
4.3	Organização da Segunda Matriz com as DFTs	48
4.4	Resultados da Comparação com a Primeira Base de Dados	49
4.5	Imagens com dois segmentos classificados corretamente	50
4.6	Resultados da Comparação com a Segunda Base de Dados	51
4.7	Imagens com todas as bordas classificadas corretamente	51

Capítulo 1

Introdução

1.1 Tema

Em Engenharia Eletrônica, processar uma imagem consiste em realizar operações em um sistema, onde a entrada e saída correspondem a algum tipo de imagem, utilizando métodos de Processamento de Sinais.

Através de técnicas de Processamento de Imagem é possível extrair informações importantes da imagem utilizando-as como dados para um sistema artificial capaz de imitar a percepção visual humana. Esta área de pesquisa é conhecida como Visão Computacional, e se insere dentro de estudos de Inteligência Computacional, Processamento de Sinais e Imagens.

Este trabalho utiliza técnicas de Processamento Digital de Sinais e Imagens para criar um algoritmo capaz de localizar, em uma fotografia, tubulações e identificar possíveis curvas ou defeitos na estrutura.

1.2 Delimitação

O principal objeto de estudo é criar um algoritmo capaz de localizar e identificar possíveis defeitos em tubulações. Tais tubulações podem apresentar curvas que precisam ser identificadas e analisadas utilizando técnicas de Processamento de Sinais e Imagens.

Desde a antiguidade o homem construiu aquedutos responsáveis por levar água de

regiões distantes para cidades e centros urbanos. Atualmente tubulações cruzam continentes, mares e cidades carregando água, gases e petróleo sob pressão para abastecer os mais diversos serviços da sociedade contemporânea.

Com a evolução dos processadores tornou-se possível o Processamento de Imagens Digitais, capaz de realizar algoritmos e operações de visão computacional. Devido à necessidade de fiscalizar constantemente as condições externas das tubulações, torna-se interessante automatizar esses processos utilizando técnicas de Processamento de Imagens e Inteligência Artificial.

1.3 Justificativa

A análise e inspeção da qualidade de tubulações são atividades extremamente importantes para a prevenção de diversos acidentes, como a explosão de um gasoduto ou o vazamento de óleo em uma baía. Os problemas resultantes podem causar danos instantâneos e futuros para o meio ambiente e a população local, além de grandes prejuízos para a empresa responsável pela tubulação.

Tubulações costumam ser estruturas longas e muitas vezes encontram-se em regiões isoladas e de difícil acesso. Por este motivo muitas vezes a inspeção é feita através de câmeras de vídeo, que enviam o sinal para estações base onde técnicos ficam responsáveis por analisar as imagens.

Devido à importância de manter as tubulações em boas condições, a análise da qualidade dos dutos não deve cometer erros. Por ser uma atividade monótona, muitas vezes a inspeção humana perde detalhes que representam algum pequeno problema devido à desatenção humana. Mesmo pequeno, qualquer defeito ou irregularidade pode ser fundamental para identificar um grande problema.

1.4 Objetivos

O objetivo geral é propor um modelo capaz de automatizar o processo de reconhecimento de curvatura de uma tubulação em uma imagem digital. Desta forma, têm-se como objetivos específicos:

1. Encontrar os melhores parâmetros e modelos de detectores de borda para o ambiente das imagens;
2. Localizar os principais segmentos (bordas de um duto) resultantes do detector de bordas;
3. Analisar a variação da inclinação das bordas da tubulação;
4. Identificar e qualificar as possíveis curvas de acordo com os dados obtidos no item anterior.

1.5 Metodologia

A aplicação de detectores de borda em uma imagem resulta em uma imagem binária, com as mesmas dimensões da primeira, onde os pixels ‘uns’ correspondem às bordas da imagem original. O Detector de Bordas de Canny [1] resulta em imagens binárias onde cada borda possui apenas um pixel de espessura. Por este motivo, juntamente ao fato de obter, de maneira geral, melhores resultados em relação aos demais detectores de bordas existentes, o algoritmo proposto por Canny foi escolhido para ser utilizado na primeira etapa do projeto

Devido ao fato deste detector de borda possuir apenas um pixel de espessura, é possível medir o comprimento de um segmento utilizando recursos computacionais simples. Desta forma podemos identificar os principais segmentos resultantes do filtro detector de bordas, eliminando eventuais ruídos.

Conhecendo um segmento, podemos “caminhar” sobre ele, e desta forma calcular como varia a inclinação ao longo do segmento. A análise destes dados no domínio da Transformada Discreta de Fourier visa fornecer informações que não estão explícitas ao observar os ângulos que a curva do segmento descreve.

O processamento das imagens e a classificação dos dutos são realizados utilizando métodos computacionais. Como as imagens em questão são digitas, podemos utilizar a precisão e a capacidade de processamento dos computadores atuais para realizar todas as operações matemáticas propostas neste projeto rápida e eficientemente.

1.6 Descrição

No Capítulo 2 será apresentada uma introdução básica ao estudo de processamento digital de imagens, com enfoque para as ferramentas utilizadas neste projeto. Serão definidos conceitos como a luminância e a representação espacial de uma imagem. O objetivo é resumir o conhecimento básico necessário para realizar uma filtragem capaz de detectar bordas.

O Capítulo 3 descreve detalhadamente, de forma teórica, o algoritmo utilizado para realizar o projeto, explicando e exemplificando os motivos para a utilização de cada parte.

Testes e resultados estão presentes no Capítulo 4, onde são ilustradas as classes básicas que consideramos para realizar a classificação. A formação da base de dados para comparações futuras e os resultados da aplicação de algumas imagens de teste para a inspeção automática são apresentados nesta seção.

A Conclusão, no Capítulo 5, discute os problemas e resultados do projeto, levando em consideração as condições e objetivos para a realização deste.

Capítulo 2

Análise de Imagens Digitais

A finalidade deste capítulo é introduzir ao leitor alguns conceitos básicos sobre Processamento de Imagens Digitais, fundamentais para a compreensão e execução deste projeto. Em Visão Computacional, a execução de um bom processamento de imagens permite a extração de parâmetros que tornam possível executar uma operação matemática capaz de compreender e interpretar uma situação de interesse.

Na primeira parte é explicado o conceito de luminância como uma propriedade da luz. Além disso, é apresentada a transformação de uma cor qualquer, escrita por três cores espectrais fundamentais, em um valor de luminância, levando em consideração a resposta em frequência do olho humano. Em seguida tratamos da representação de imagens digitais como funções bidimensionais discretas, onde cada pixel corresponde a luminância, ou nível de cinza, de um dado ponto.

O princípio da filtragem digital de imagens é apresentado em seqüência, de forma a introduzir o conceito utilizado em segmentação de imagens, ponto principal deste capítulo. A filtragem é utilizada para realce de imagens e pode ser feita no domínio da frequência ou espacial. Este último caso receberá maior destaque neste capítulo.

Na seção dedicada à segmentação, trataremos basicamente dos métodos de detecção de descontinuidades, em especial a localização de bordas. A realização de uma boa segmentação é fundamental para a maioria dos projetos de Visão Computacional. Uma boa detecção de bordas pode decidir sobre o sucesso de um sistema como o apresentado neste trabalho.

2.1 Luz e Luminância

Consideramos luz, a radiação eletromagnética capaz de estimular a resposta visual humana. A região do espectro eletromagnético visível encontra-se entre os comprimentos de onda (λ) 350 nm e 780 nm. A luz pode ser expressa como uma distribuição espectral de energia $L(\lambda)$, em W/m^3 . A luz recebida, que nos permite visualizar um objeto, pode ser escrita como

$$I(\lambda) = \rho(\lambda)L(\lambda) \quad (2.1)$$

onde $\rho(\lambda)$ é uma grandeza adimensional que representa a refletividade de um objeto [2].

O sistema visual humano possui dois tipos de células fotoreceptoras, sensíveis à radiação com comprimento de onda correspondente à luz. Essas células são conhecidas pelos nomes de cones e bastonetes. Tais células encontram-se na retina, região posterior do olho, onde se forma a imagem.

A retina possui uma quantidade muito maior de bastonetes em relação aos cones. Os bastonetes tendem a apresentar respostas em frequência idênticas, e não são capazes de distinguir cores, apenas níveis de cinza. São células fotoreceptoras capazes de captar uma baixa intensidade luminosa, e por este motivo são basicamente responsáveis pela visão noturna.

Existem três tipos de cones, que apresentam respostas em frequência distintas, correspondentes a cada uma das três cores primárias: vermelho, verde e azul. Estas cores são consideradas primárias, pois os picos das respostas em frequência dos cones são próximos aos comprimentos de onda destas. A combinação das cores primárias permite a visualização de qualquer cor no espectro visível [2].

A luminância corresponde a uma medida de densidade da intensidade da luz refletida, medida pela unidade candela por metro quadrado (cd/m^2). Em engenharia é comum utilizar o termo luminância para se referir ao brilho, por exemplo, de um monitor. Neste caso, uma forma de representação é:

$$Y = 0,177R + 0,813G + 0,011B \quad (2.2)$$

onde R, G e B são fontes monocromáticas, correspondentes às cores espectrais primárias, propostas pelo padrão C.I.E. (*Commission internationale de l'éclairage*). Este padrão



Figura 2.1: Imagem colorida e sua luminância

considera as cores primárias como três fontes impulsivas com comprimentos de onda em 700 nm, 546 nm e 435 nm, para as cores vermelho (R), verde (G) e azul (B), respectivamente.

A relação apresentada na Equação (2.2) vem do fato de que podemos representar qualquer cor através de uma mistura de luzes das três cores primárias. A luminância de uma mistura de cores é igual à soma das luminâncias destas cores. Então a luminância de uma cor $C(\lambda)$ pode ser obtida como

$$Y = Y(C) = \int C(\lambda)V(\lambda)d\lambda \quad (2.3)$$

sendo $V(\lambda)$ uma função com formato de sino, similar a resposta do olho humano.

O olho humano não é capaz de distinguir as componentes de uma mistura de cores. Isto significa que uma fonte de luz monocromática e sua cor não são únicas em relação uma à outra. O sistema visual não é capaz de identificar o comprimento de onda de uma cor. As luminâncias provenientes de duas diferentes misturas de cores podem apresentar o mesmo valor [2].

Na Figura 2.1 podemos observar uma imagem colorida e a imagem resultante de sua luminância, calculada em relação as componentes R, G e B.

É possível definir uma luminância relativa através de um branco de referência, que recebe o valor máximo na escala que desejamos utilizar. Normalizando esta me-

dida, teremos o valor um para o branco e zero para o preto. Ou seja, a ausência de luminância corresponde à ausência de cor, o que resulta no preto. Quanto maior o valor da luminância, maior a quantidade de luz visualizada, e o maior valor será o branco que tomamos como referência.



Figura 2.2: Imagens com as luminâncias de R, G e B

A imagem da luminância apresentada na Figura 2.1b corresponde à soma das luminâncias das fontes luminosas das três cores primárias, ponderadas por multiplicadores conforme a Equação (2.2). Na Figura 2.2, são apresentadas as luminâncias relativas das fontes de luz vermelha, verde e azul, primeiro em tons de cinza e depois com tonalidades relativas às cores primárias.

Somando as Figuras 2.2a, 2.2b e 2.2c, obtemos a luminância da Figura 2.1b. Da soma das Figuras 2.2d, 2.2e e 2.2f resulta a imagem colorida apresentada na Figura 2.1a.

Podemos observar que a região vermelha da bandeira aparece mais escura nas Figuras 2.2b e 2.2c, em relação a Figura 2.2a. Isto acontece, pois a intensidade de luz vermelha é maior naquela região da imagem do que as componentes verde ou azul. O mesmo efeito ocorre para a imagem referente à cor verde, quando olhamos o gramado.

No caso da cor azul, o gramado apresenta um brilho muito baixo, assumindo quase o valor do preto, tanto na imagem em tons de cinza, quanto em tons de azul.

2.2 Representação de Imagens Digitais

Uma imagem monocromática¹ pode ser expressa por $f(x, y)$, onde x e y correspondem às coordenadas espaciais, e o valor da função em qualquer ponto apresenta o valor da luminância da imagem naquele ponto. A Figura 2.3 demonstra uma convenção comumente adotada na literatura sobre o assunto.

Para uma imagem digital, a função $f(x, y)$ é discretizada tanto nas coordenadas espaciais quanto em relação ao brilho (valor da função em um ponto x, y). Podemos então representar uma imagem digital monocromática como uma matriz contendo valores discretos, e a posição de um ponto na imagem é dada em relação aos índices das linhas e colunas. O valor da função f , em uma coordenada representada na linha m e coluna n , corresponde ao nível de cinza naquele ponto. Um ponto de uma imagem é conhecido como *pixel*.

No caso de uma imagem digital colorida, podemos tratá-la como uma matriz tridimensional, ou basicamente três matrizes, como apresentado na Figura 2.2, que sobrepostas resultam na imagem em cores. O par de coordenadas (m, n) de cada uma das matrizes R, G e B, representa o mesmo pixel na imagem final, colorida.

O tamanho da matriz que representanta a imagem é conhecido como resolução da imagem. Quanto maior a resolução, maior o número de pixels. A resolução é normalmente apresentada como Número de linhas x Número de colunas.

Além da discretização espacial, imagens digitais também são discretizadas em relação à amplitude do sinal. A escolha dos níveis de quantização assume um compromisso entre qualidade e tamanho do arquivo. Quanto menor o passo de quantização, maior será a quantidade de valores entre o preto e o branco de referência. Conseqüentemente precisaremos de mais bits para armazenar a informação de cada pixel da imagem.

¹Popularmente conhecida por imagem em preto & branco, é uma imagem em nível de cinza, baseada nos níveis de luminância, que varia do preto (menor valor) até o branco (maior valor possível na escala).



Figura 2.3: Representação de uma Imagem Digital

Uma imagem monocromática quantizada com oito bits permite à luminância assumir 256 valores, onde zero é o preto e 255 o branco de referência.

2.3 Filtros de Imagens

Em Processamento de Imagem, filtros digitais são basicamente utilizados com a finalidade de realçar determinadas regiões de interesse. As operações de filtragem podem ser realizadas tanto no domínio espacial quanto na frequência (período por medida de espaço ou radianos por pixels). Os filtros são classificados normalmente em três categorias básicas, conforme descritos abaixo:

- **Passa-baixas** - Atenua as componentes de altas frequências, realçando as de baixas. O efeito visual deste filtro em uma imagem é a suavização desta, pois as altas frequências representam transições abruptas na imagem.
- **Passa-altas** - Realça as componentes de altas frequências, e atenua as baixas. A aplicação deste tipo de filtro em uma imagem visa, normalmente, realçar detalhes como as bordas, que correspondem a transições entre duas regiões da imagem.

- **Passa-faixa** - Seleciona uma faixa de frequências que desejamos realçar. Quando queremos atenuar um intervalo de frequências, este filtro costuma receber o nome de rejeita-faixa.

2.3.1 Filtros no Domínio Espacial

Um filtro de imagem no domínio espacial utiliza a informação do nível de cinza de um dado ponto e seus vizinhos para determinar o novo valor para aquele pixel. O domínio espacial corresponde ao conjunto de pixels que formam a imagem. Em geral, a operação matemática usada para calcular o novo valor de um determinado pixel na imagem considera uma maior influência dos pontos mais próximos ao pixel em relação aos mais afastados.

Este tipo de filtragem é realizado normalmente utilizando o método das máscaras, ou janelas, implementado por meio de matrizes com pesos (coeficientes numéricos), que percorrem a imagem calculando os novos valores. Sendo um pixel na coordenada (X, Y) , onde X é a linha e Y a coluna correspondente a este ponto, a aplicação da máscara com centro nesta posição consiste em substituir o seu valor pela soma dos vizinhos ponderados pelos pesos da máscara. As janelas podem assumir diferentes formas e seu centro pode estar localizado fora da região com os coeficientes numéricos.

$w(1,1)$	$w(1,2)$	$w(1,3)$
$w(2,1)$	$w(2,2)$	$w(2,3)$
$w(3,1)$	$w(3,2)$	$w(3,3)$

Figura 2.4: Máscara de tamanho 3×3 com pesos $w(l,k)$

A Figura 2.4 ilustra uma máscara de tamanho 3×3 com pesos para cada uma das posições $w(l, k)$. Esta máscara caminha sobre uma imagem, de forma que seu pixel central sobreponha todos os pontos da imagem. A operação matemática para o cálculo

da resposta deste filtro é dada por:

$$R(x, y) = \sum_{l=1}^3 \sum_{k=1}^3 w_{(l,k)} f_{(x-(3-l)+1, y-(3-k)+1)} \quad (2.4)$$

onde x e y são as coordenadas do pixel sobreposto pelo centro da máscara, e $R(x, y)$ representa o novo valor que este ponto receberá após a conclusão da filtragem.

Para realizar uma filtragem espacial é necessário empregar uma imagem auxiliar com as mesmas dimensões da original. Esta imagem auxiliar ($R(x, y)$) receberá os valores calculados com a aplicação do filtro, sem que tal operação altere os valores da imagem original, evitando que a filtragem de um pixel anterior não influencie no seguinte.

Utilizando a Figura 2.4 e a Equação (2.4) como referência, podemos construir um filtro passa-baixas através da média dos pixels da região de dimensão 3×3 . Uma solução seria atribuir o valor $1/9$ para cada um dos pesos $w(l, k)$. Um outro exemplo seria aproximar os pesos de uma máscara para uma função Gaussiana bidimensional, como será apresentado na Seção 3.2.1.

De forma análoga uma aproximação de máscara para realizar um filtro passa-altas daria um peso maior para o centro da máscara em relação as extremidades, como mostrado na Seção 2.4.1, onde a Figura 2.5 apresenta uma máscara para detecção de pontos isolados em uma imagem.

2.3.2 Filtros no Domínio da Frequência

As técnicas de filtragem no domínio da frequência são baseadas no teorema da convolução, aplicado no domínio espacial e bidimensional. A convolução de uma imagem $\mathbf{f}(x, y)$ com um operador linear $\mathbf{h}(x, y)$ resulta na imagem $g(x, y)$, expresso por:

$$\mathbf{g}(x, y) = \mathbf{f}(x, y) * \mathbf{h}(x, y) = \sum_l \sum_k f(l, k) h(x - l, y - k). \quad (2.5)$$

De acordo com as propriedades da Transformada de Fourier, no domínio da frequência, a convolução de duas funções corresponde ao produto de suas transformadas, obedecendo à relação a seguir:

$$\mathbf{G}(u, v) = \mathbf{F}(u, v)\mathbf{H}(u, v). \quad (2.6)$$

onde \mathbf{G} , \mathbf{F} e \mathbf{H} são os resultados da Transformada de Fourier Bidimensional nas matrizes \mathbf{g} , \mathbf{f} e \mathbf{h} , respectivamente. $H(u, v)$ é chamada de Função de Transferência do sistema, que representa o tipo de filtro que desejamos aplicar [3].

2.4 Segmentação

A segmentação é a principal técnica de Processamento de Imagens utilizada neste trabalho. Costuma ser a primeira etapa em análise de imagens, desconsiderando o pré-processamento, pois consiste em subdividir uma imagem em partes ou regiões, com características distintas. Busca-se na segmentação isolar um objeto de interesse. No caso deste projeto, queremos isolar tubulações apresentadas em imagens, para realizar as operações de identificação e classificação dos dutos.

Segmentação é uma das tarefas mais complicadas em um projeto de processamento de imagens. Devido à sua fundamental importância, a precisão na detecção e isolamento dos objetos de interesse é determinante para que a continuação do processo seja concluída da maneira desejada.

Os algoritmos de segmentação aplicados em imagens monocromáticas se baseiam na descontinuidade e similaridade dos valores dos pixels. Em ambos os casos, o conceito de segmentação baseado nas variações da luminância da imagem pode ser aplicado em imagens estáticas (que não variam) ou dinâmicas (que variam com o tempo). Normalmente a segmentação de imagens dinâmicas pode ser facilitada pela possibilidade de comparar variações na imagem ao longo do tempo.

Detecção de descontinuidade consiste em subdividir a imagem com base em variações no nível de cinza. As principais áreas de interesse são a detecção de pontos isolados, detecção de linhas e detecção de bordas. Nos três casos de detecção de descontinuidade, podemos utilizar, basicamente, filtros passa-altas no domínio espacial para localizar as regiões de interesse. A segmentação por similaridade se baseia em limiarização, crescimento de regiões, e divisão e fusão de regiões [4].

2.4.1 Detecção de Pontos

Em uma imagem, podemos detectar os pontos isolados de maneira simples. Aplicando uma máscara, como a da Figura 2.4, calcula-se uma nova imagem $R(x, y)$, através da Equação (2.4). Estabelecendo um *threshold* positivo (T), podemos determinar se um ponto está isolado em relação aos vizinhos, através da comparação:

$$|R| > T. \quad (2.7)$$

Na Figura 2.5 é ilustrada uma máscara utilizada para filtragem passa-altas, apresentada em [4] e [3]. Porém no caso da detecção de pontos isolados, somente aqueles que superarem o limiar serão considerados de interesse. Esta máscara apresenta um peso alto no ponto central e pequeno nas periferias, e o resultado da sua aplicação é uma filtragem que avalia o ponto central em relação aos vizinhos, capaz de detectar pontos isolados em um fundo constante.

-1	-1	-1
-1	8	-1
-1	-1	-1

Figura 2.5: Máscara de um filtro passa-altas utilizada para detecção de pontos isolados

Para este caso um valor de threshold que pode ser utilizado é $T = 1$.

2.4.2 Detecção de Linhas

A detecção de linhas apresenta um nível de complexidade maior do que a detecção de pontos. Detecção de linhas se baseia na comparação dos pixels de uma determinada direção com os demais pixels sobrepostos pela janela. A Figura 2.6 apresenta quatro máscaras, propostas em [4] e [3], onde cada uma é responsável por uma direção.

Observando a janela da Figura 2.6a notamos que esta apresenta pesos positivos em uma reta horizontal que passa pelo centro, e negativos nos demais pontos. Desta forma a máscara é capaz de comparar se existe diferença entre uma região superior e outra inferior, em relação ao centro, por este motivo é uma janela responsável por detectar retas horizontais. Analogamente, as Figuras 2.6b 2.6c 2.6d são máscaras relativas à localização de retas nas direções da diagonal secundária da matriz (45°), vertical, e diagonal principal da matriz (135° ou -45°), respectivamente.

-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2
(a) Horizontal	(b) 45°	(c) Vertical	(d) 135°								

Figura 2.6: Máscaras para a detecção de linhas e retas

2.4.3 Detecção de Bordas

A detecção de bordas é a principal aplicação de segmentação em detecção de descontinuidades de nível de cinza. Uma borda é a fronteira entre duas regiões com valores distintos de luminância. Bordas são elementos mais comuns em imagens naturais do que segmentos retos e pontos isolados, para uma boa detecção é necessário maior complexidade computacional.

As imagens naturais costumam apresentar variações suaves do nível de cinza ao longo de sua superfície. O próprio olho humano funciona como um filtro passa-baixas. As imagens digitais apresentam poucas descontinuidades abruptas devido à suavização introduzida pela estrutura física da maioria dos dispositivos de captura e processamento utilizado para a composição da imagem em alta resolução. Além disso, apresentam bordas levemente borradas devido à quantização.

Em geral, a detecção de bordas é realizada através de um cálculo com operador

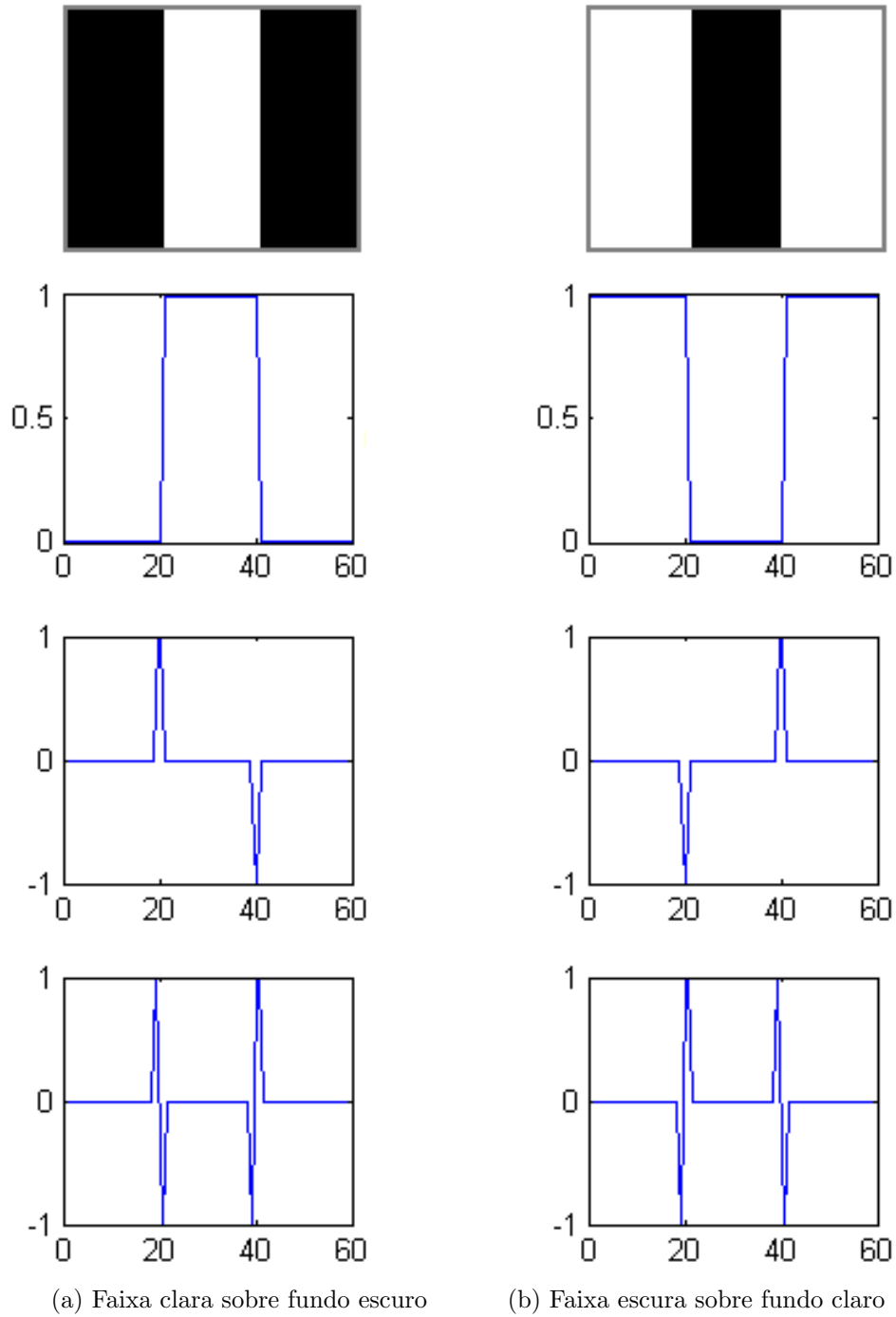


Figura 2.7: Detecção de Bordas por Operadores de Derivação

local diferencial. Esta é a principal abordagem para algoritmos com a finalidade de localizar bordas. A Figura 2.7 mostra duas imagens formadas por regiões claras e escuras. Logo abaixo da imagem é apresentado o perfil de uma linha horizontal da imagem, onde

podemos observar as variações no valor da luminância, ao longo das colunas. Abaixo, é apresentada a primeira derivada do sinal da linha horizontal apresentado. Por último aparece a representação gráfica da segunda derivada do sinal.

Observando as imagens da Figura 2.7, nota-se que a primeira derivada é nula em regiões de cinza constante, positiva nas transições de uma região escura para uma clara, e negativa em uma transição contrária. Já a segunda derivada é positiva do lado escuro de uma borda e negativa do lado positivo, sendo nula para as regiões constantes e no ponto onde existe a transição.

Tais informações indicam que a magnitude da primeira derivada pode ser utilizada na detecção de uma borda. A segunda derivada apresenta um cruzamento em zero, indicação de uma mudança no nível de cinza, no ponto onde a borda está localizada na imagem.

De forma análoga à detecção de linhas e retas, a abordagem apresentada acima pode ser aplicada em diversas direções da imagem. O cálculo da magnitude do gradiente em qualquer pixel da imagem indica a primeira derivada neste ponto. Para calcular a segunda derivada, utiliza-se um operador Laplaciano.

2.4.3.1 Operador de Gradiente

Uma imagem depende de duas coordenadas espaciais, e as bordas podem ser expressas por derivadas parciais. O operador *gradiente* é um vetor no qual a direção indica os locais onde a luminância sofre maior variação. Como descrito em [4] e [3], o gradiente de uma imagem $f(x, y)$ na posição (x, y) é dado pelo vetor:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (2.8)$$

O módulo ou magnitude do vetor *gradiente* pode ser chamado simplesmente de gradiente e é denotado por ∇f , onde

$$\nabla f = \sqrt{G_x^2 + G_y^2} = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}. \quad (2.9)$$

Para reduzir o custo computacional, a seguinte aproximação é aceita:

$$\nabla \mathbf{f} \approx |G_x| + |G_y|. \quad (2.10)$$

A *direção* do vetor gradiente é apresentada através do ângulo $\theta(x, y)$ calculado em relação ao eixo x como:

$$\theta(x, y) = \arctan\left(\frac{G_y}{G_x}\right) \quad (2.11)$$

onde θ é o ângulo calculado em relação ao eixo x .

Como comentado anteriormente, uma variação na intensidade entre pixels vizinhos pode indicar a existência de uma borda. Diferenças entre pontos adjacentes horizontais são capazes de detectar bordas verticais, enquanto bordas verticais podem ser detectadas através da diferença vertical entre pontos adjacentes da imagem.

A Figura 2.8 ilustra uma região de tamanho 3×3 de uma imagem, representada por uma função bidimensional f . A posição dos pixels é apresentada em função do pixel central, que está na coordenada (x, y) . Podemos aproximar de várias maneiras a variação da magnitude no ponto $f(x, y)$. Uma aproximação utilizando a diferença nas direções horizontal e vertical consiste em:

$$\nabla \mathbf{f} \approx |f(x, y) - f(x, y + 1)| + |f(x, y) - f(x + 1, y)|. \quad (2.12)$$

$f(x - 1, y - 1)$	$f(x - 1, y)$	$f(x - 1, y + 1)$
$f(x, y - 1)$	$f(x, y)$	$f(x, y + 1)$
$f(x + 1, y - 1)$	$f(x + 1, y)$	$f(x + 1, y + 1)$

Figura 2.8: Região 3×3 de uma imagem

Os operadores cruzados de gradiente de Roberts [5], apresentados nas máscaras da Figura 2.9, são outra forma de aproximar a magnitude do gradiente. A implementação dos operadores de Roberts pode ser expressa pela Equação (2.13). A resposta corresponde à soma dos resultados do valor absoluto da resposta da aplicação de cada uma das máscaras na imagem. Neste caso a posição referente ao ponto $f(x, y)$ na máscara corresponde à coordenada $(1, 1)$, sendo

$$\nabla \mathbf{f} \approx |f(x, y + 1) - f(x + 1, y)| + |f(x, y) - f(x + 1, y + 1)|. \quad (2.13)$$

1	0
0	-1

(a) G_x

0	1
-1	0

(b) G_y

Figura 2.9: Operadores de Gradiente de Roberts

Uma aproximação mais robusta para o cálculo do gradiente da Equação (2.10), utilizando uma maior quantidade de vizinhos, pode ser calculada utilizando as máscaras de Prewitt [6], que utilizam uma região 3×3 , com o pixel central da janela sendo a posição de origem ($f(x, y)$ da Figura 2.8) do cálculo.

As máscaras de Prewitt podem ser vistas na Tabela 2.1, onde G_x corresponde ao cálculo das bordas horizontais e G_y ao das bordas verticais. Os operadores diferenciais de Prewitt aproximam a derivada na direção horizontal através da diferença entre as colunas das extremidades da máscara G_x , enquanto que a diferença das linhas um e três da matriz G_y aproxima a derivada na direção vertical.

Na Tabela 2.1, apresentada em [2], são ilustrados alguns operadores diferenciais de baixa complexidade. Esses operadores utilizam apenas variações verticais e horizontais para determinar as bordas de uma imagem. Máscaras que comparam as linhas e as colunas das extremidades são comumente conhecidas como operadores de Sobel.

O operador de Sobel [7] aproxima a magnitude do gradiente ponderando a diferença entre os valores dos níveis de cinza da imagem. Na Figura 2.10 podemos ver a aplicação do filtro de Sobel em uma imagem monocromática (Figura 2.10a). As respostas das máscaras G_x e G_y são apresentadas nas Figuras 2.10b e 2.10c, respectivamente, enquanto que a resposta final do detector de bordas de Sobel, com a aplicação sobreposta das respostas de $|G_x|$ e $|G_y|$, está na Figura 2.10d.

Partindo do princípio dos operadores apresentados anteriormente, surgiram outros operadores que utilizam um maior número de máscaras. O operador de Krisch [8] utiliza oito máscaras de convolução orientadas em 45° . O gradiente é obtido através da resposta máxima em conjunto das oito máscaras. O operador de Robinson [9], assim como o

Tabela 2.1: Máscaras dos Gradientes

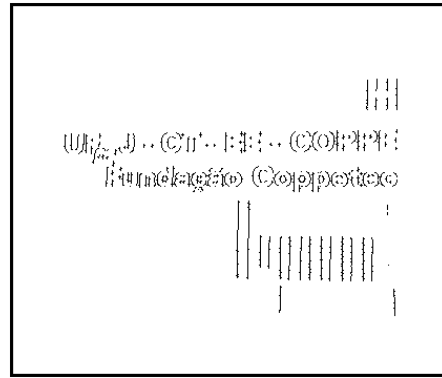
Origem do Gradiente	G_x	G_y
Roberts	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$
Prewitt	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$
Sobel	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$
Isotrópica	$\begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$

de Krisch, utiliza oito máscaras e é calculado através do valor máximo entre elas. Já o operador de Frei-Chen [10] utiliza nove máscaras que formam uma base ortogonal. Existem neste caso máscaras responsáveis por bordas, outras por retas e uma pela média dos pixels na região 3×3 da janela [3].

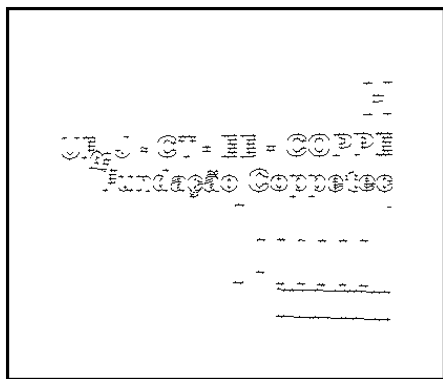
É possível aplicarmos máscaras com dimensões maiores. Porém o tempo de processamento aumenta, uma vez que mais variáveis serão consideradas para o cálculo do gradiente dos pixels. A vantagem seria a redução de efeito aditivos de ruídos.



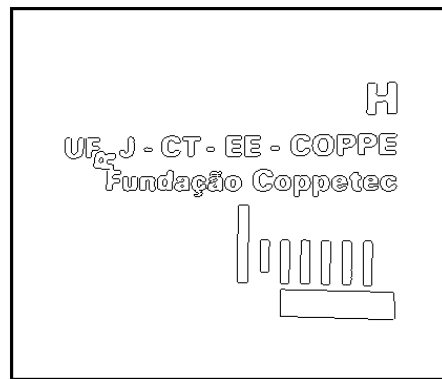
(a) Imagem Original



(b) Resultado de G_x



(c) Resultado de G_y



(d) Composição entre os dois gradientes

Figura 2.10: Detecção de Bordas pelo Operador de Sobel

2.4.3.2 Laplaciano

O Laplaciano de uma função bidimensional ($\nabla^2 f$) corresponde à soma das derivadas parciais de segunda ordem:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (2.14)$$

Semelhante ao caso do cálculo do gradiente em imagens digitais, o Laplaciano pode ser implementado de diversas maneiras, inclusive utilizando filtragem bidimensional pelo método das máscaras. No caso clássico, do cálculo utilizando uma máscara de tamanho 3×3 , a forma mais encontrada é expressa por

$$\nabla^2 f_{(x,y)} = f_{(x,y)} = 4f_{(x,y)} - [f_{(x+1,y)} + f_{(x-1,y)} + f_{(x,y+1)} + f_{(x,y-1)}], \quad (2.15)$$

onde $f(x, y)$ corresponde à posição do pixel da imagem sobreposto pela posição central da máscara. A exigência para definir o Laplaciano em sua forma digital é que o coeficiente associado ao pixel central, $f(x, y)$, seja positivo e os outros extremos sejam negativos, como ilustrado na Figura 2.11 [4].

Por se tratar de uma derivada de segunda ordem, o Laplaciano é muito sensível a ruídos, o que é indesejável em detecção de bordas. Além disso, este cálculo apresenta bordas duplas, para uma única borda, como observado na Figura 2.7. Tais motivos tornam o Laplaciano apenas uma ferramenta auxiliar na detecção de bordas, e não um detector. Sua função básica é indicar se um pixel está do lado claro ou escuro de uma borda. É possível utilizar o Laplaciano para localizar bordas através do cruzamento por zero.

0	-1	0
-1	4	-1
0	-1	0

Figura 2.11: Máscara para o cálculo do Laplaciano

Capítulo 3

Algoritmo

3.1 Descrição Geral

O algoritmo proposto neste trabalho visa localizar as bordas de uma tubulação em uma imagem, e através destas bordas classificar o duto de acordo com classes pré-estabelecidas. Tais classes representam grupos de defeitos, ou formas, que a estrutura de uma tubulação pode apresentar. São utilizados métodos de Processamento Digital de Imagens e Sinais para realizar as etapas deste algoritmo.

A detecção de bordas é um dos métodos de segmentação de imagens, em que a fronteira entre duas regiões relativamente distintas é identificada como borda. Para localizar as bordas presentes na imagem é utilizado o Detector de Bordas de Canny. Este detector foi escolhido por propor uma “resposta mínima” na detecção de bordas.

Uma vez localizadas todas as bordas de uma imagem, é necessário que o sistema decida quais delas representam a fronteira entre o duto que queremos classificar e o fundo da imagem. Inicialmente todas as bordas encontradas pelo filtro de Canny são identificadas e as coordenadas de todos os pixels classificados como borda são armazenadas na memória do sistema. A etapa de separar os segmentos consiste justamente em agrupar em uma estrutura todas as coordenadas de uma borda, também chamada de segmento, e o número de pixels que este contém. Cada posição na estrutura corresponde a uma borda.

Conhecendo o número de pixels de cada borda, estas são ordenadas dentro da

estrutura de acordo com os seus comprimentos. O segmento com o maior número de pixels recebe a primeira posição, e o com o menor número é inserido na última posição. A princípio, o sistema escolherá os dois maiores segmentos para classificar o duto, pois partimos da hipótese de que o contraste entre o duto e o fundo da imagem representa as duas maiores bordas possíveis na imagem.

Esta suposição mostra a necessidade da etapa de segmentação apresentar resultados próximos do ideal. A boa detecção das bordas é fundamental para a seqüência deste projeto, uma vez que após detectar uma borda que não corresponde à tubulação em questão, o sistema classificará o duto na imagem de maneira equivocada.

Depois de encontradas, as maiores bordas da imagem são comparadas com outras bordas armazenadas em um banco de dados. A comparação é feita com base em um histograma com as inclinações que uma borda apresenta ao longo de sua trajetória. O cálculo deste histograma é descrito na Seção 3.4, enquanto que a comparação entre as bordas é descrita na Seção 3.5.

3.2 Detector de Bordas de Canny

O Detector de Bordas de Canny foi proposto por John Canny em 1986 [1]. Um conjunto de objetivos é definido para o cálculo ótimo da localização dos pontos de uma borda em uma imagem.

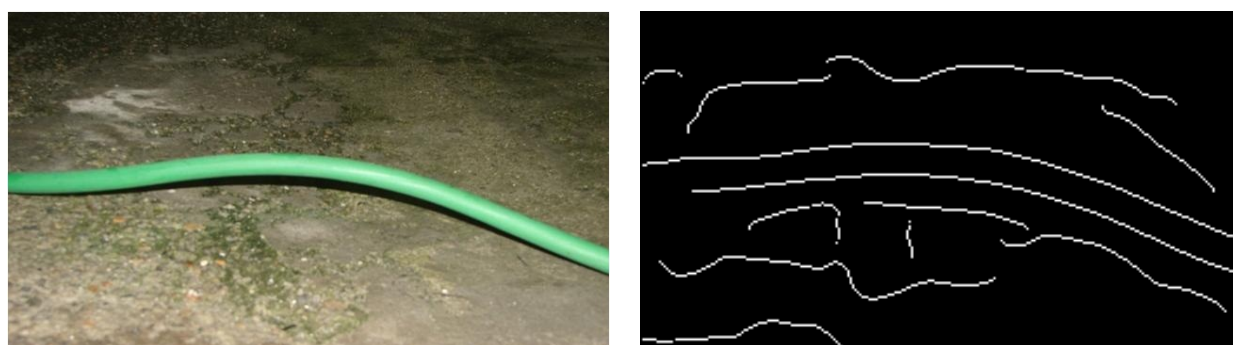
- **Boa detecção** - O algoritmo deve ser capaz de encontrar todas as bordas possíveis na imagem;
- **Boa localização** - As bordas encontradas devem estar o mais próximo possível da borda na imagem original;
- **Resposta mínima** - Cada borda deve ser marcada apenas uma vez, com espessura mínima, minimizando a influência do ruído da imagem sobre a geração de bordas falsas.

Inicialmente a imagem é filtrada por um filtro passa-baixas Gaussiano, com o objetivo de reduzir ruídos presentes na imagem. Em seguida a magnitude e a direção do

gradiente são calculadas utilizando aproximações de filtragem pelo método de janelas em quatro sentidos (vertical, horizontal e as duas diagonais).

Em seguida as bordas são localizadas utilizando o método da *supressão não-máxima*, responsável por reduzir a espessura das bordas à dimensão mínima. O algoritmo de Canny utiliza dois limiares com histerese para evitar a detecção de bordas inexistentes e a ocultação de bordas com baixa magnitude.

A resposta de um filtro de Canny é uma imagem binária com as mesmas dimensões da original, onde os pixels brancos representam as bordas detectadas, conforme ilustra a Figura 3.1.



(a) Imagem Original

(b) Imagem Binária Contendo Bordas

Figura 3.1: Bordas de uma Imagem

3.2.1 Filtragem Gaussiana

O filtro utilizado no algoritmo de Canny tem o objetivo de borrar ou suavizar a imagem, reduzindo as altas frequências e ruídos presentes na imagem. Para realizar esta operação é utilizado um filtro Gaussiano, que basicamente consiste de um filtro passa-baixas, com resposta ao impulso, para o caso unidimensional, dado por

$$h(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (3.1)$$

onde σ^2 corresponde à variância da Gaussiana.

No caso bidimensional, a resposta ao impulso do filtro Gaussiano pode ser expressa por

$$h(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.2)$$

O desvio padrão corresponde à raiz quadrada da variância. Quanto maior este parâmetro, maior a largura do filtro Gaussiano. Ao reduzir o desvio padrão, aceitaremos mais altas frequências, permitindo ruídos de transições abruptas.

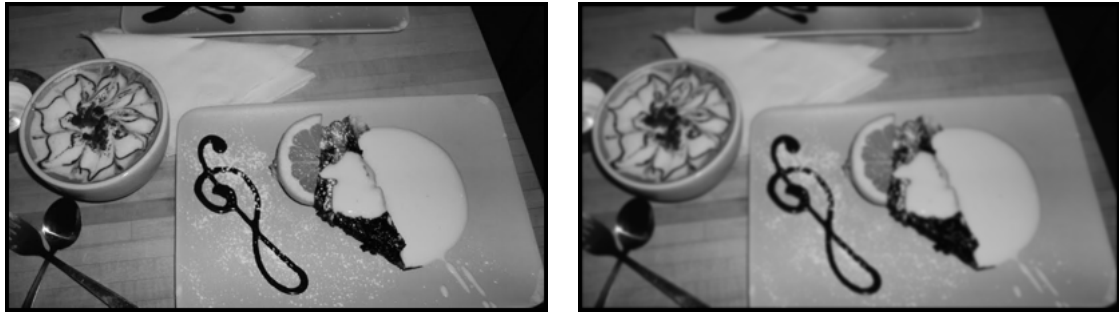
Devido ao termo $(x^2 + y^2)$, notamos que a gaussiana bidimensional assume o mesmo valor em diferentes direções, para um mesmo raio. Quando maior este raio, menor será o valor de h .

Um exemplo aproximado de máscara para um filtro Gaussiano bidimensional, de tamanho 5×5 e desvio padrão 1, pode ser visto na Figura 3.2. O cálculo da máscara é baseado na função apresentada na Equação (3.2). O centro da máscara corresponde ao pico da Gaussiana, pois está relacionado à posição $(0, 0)$. Na Figura 3.3 podemos observar o resultado da aplicação desta máscara em uma imagem. É fácil observar que as componentes de alta frequência (transições rápidas na Figura 3.3a) aparecem borradas na imagem filtrada, mostrada na Figura 3.3b, como se os objetos na cena estivessem desfocados.

0.0030	0.0133	0.0219	0.0133	0.0030
0.0133	0.0596	0.0983	0.0596	0.0133
0.0219	0.0983	0.1621	0.0983	0.0219
0.0133	0.0596	0.0983	0.0596	0.0133
0.0030	0.0133	0.0219	0.0133	0.0030

Figura 3.2: Máscara 5×5 de um Filtro Gaussiano com $\sigma = 1$

Após esta etapa, são calculadas as derivadas parciais direcionais de primeira ordem, implementadas através de máscaras, ou filtros, como apresentado no Capítulo 2.



(a) Imagem Original

(b) Resultado do Filtro Gaussiano

Figura 3.3: Imagem Filtrada pelo filtro Gaussiano da Figura 3.2

3.2.2 Supressão Não-Máxima

O detector de bordas de Canny utiliza quatro máscaras para estimar as magnitudes dos gradientes dos pixels. Calculadas as estimativas dos gradientes de uma imagem, inicia-se uma busca pixel a pixel para determinar se a direção em que a magnitude do gradiente assume valor máximo está de acordo com o valor correspondente de seus vizinhos.

Se o gradiente máximo encontrado corresponder ao ângulo do zero graus, o ponto será considerado como uma borda se sua intensidade for maior do que a dos pixels na direção leste-oeste. De forma semelhante, se o valor máximo do gradiente corresponder ao ângulo de 90° , este ponto estará sobre uma borda se possuir maior intensidade que os vizinhos na direção vertical. Uma vez que o algoritmo de Canny considera quatro gradientes, também são considerados os valores para bordas com ângulos em 135° e 45° . Esta comparação para determinar uma borda utiliza uma janela de tamanho 3×3 .

Caso um ponto com alta magnitude de gradiente possua vizinhos com magnitude inferior nas direção do gradiente, este pixel é marcado como um pico da primeira derivada ou ponto de transição abrupta. Este processo é denominado supressão não máxima e pode ser observado na Figura 3.4, que ilustra o resultado da aplicação dos operadores de gradiente sobre uma região da imagem.

Os pixels brancos da segunda coluna podem ser considerados bordas verticais, com gradientes aproximados na direção horizontal, zero grau. Notamos que os três pixels inferiores possuem valores maiores do que seus vizinhos à esquerda e direita, logo estes são considerados como pertencentes a uma borda vertical. Porém o pixel na segunda

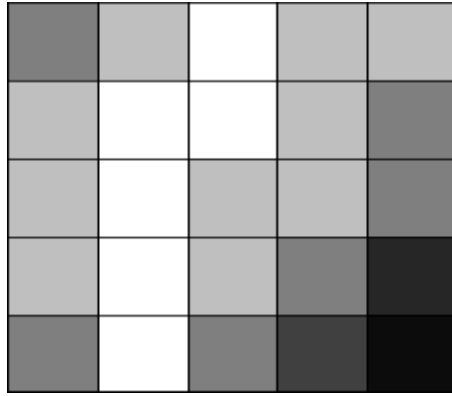


Figura 3.4: Exemplo de Supressão Não-Máxima

linha apresenta uma magnitude similar à do vizinho da direita, e não respeita o critério de maior intensidade em relação aos vizinhos na direção leste-oeste.

Na posição (4, 4) observamos um pixel com gradiente aproximado na direção de 135° . Porém seu vizinho (3, 3) apresenta maior intensidade, enquanto que o vizinho do outro lado, posição (5, 5) apresenta um valor menor. Neste caso o pixel não se sobressai em intensidade sobre os vizinhos na direção do gradiente, logo não é considerado borda. A classificação dos pixels entre borda e não borda resulta em uma imagem binária, onde os pixels brancos correspondem a bordas, e os pretos a regiões sem transição.

3.2.3 Limiares com Histereses

A supressão não-máxima provavelmente encontrará todas as bordas relevantes da imagem. Porém, os detectores de bordas costumam apresentar bordas falsas. A decisão sobre um pixel ser ou não uma borda está ligada à escolha do *threshold*. Um limiar muito alto pode levar à perda de informações importantes. Por outro lado, um limiar muito baixo acarretará na detecção de bordas irrelevantes, como as causadas por ruídos. Tentando diminuir este problema, Canny propõe uma segunda seleção, dentre um subconjunto das bordas encontradas.

São introduzidos dois valores de *threshold*, um com o valor máximo (T2) e outro com o valor mínimo (T1) para a classificação de uma borda. O primeiro busca evitar que ruídos da imagem acabem classificados como borda e o inferior serve para encontrar

bordas sutis. Inicialmente são marcados os pontos das bordas para o limiar superior. Se o pixel de uma imagem em escala de cinza apresentar gradiente direcional acima do limiar superior ele é marcado, caso contrário é desconsiderado. Isso marca traços e pontos que podem ter grandes chances de serem bordas verdadeiras.

Em seguida todos os pontos ao redor de um pixel assinalado, que possuam valor do gradiente direcional superior ao *threshold* inferior, também são marcados como bordas. Isso permite traçar pontos fracos de uma borda, sem interrompê-la constantemente. Devemos levar em consideração que bordas são descontinuidades longas e contínuas, e ruídos apresentam comprimentos curtos, que interferem na detecção correta de bordas. Este procedimento permite determinar um ponto de partida para encontrar as retas por meio de cálculo iterativo.

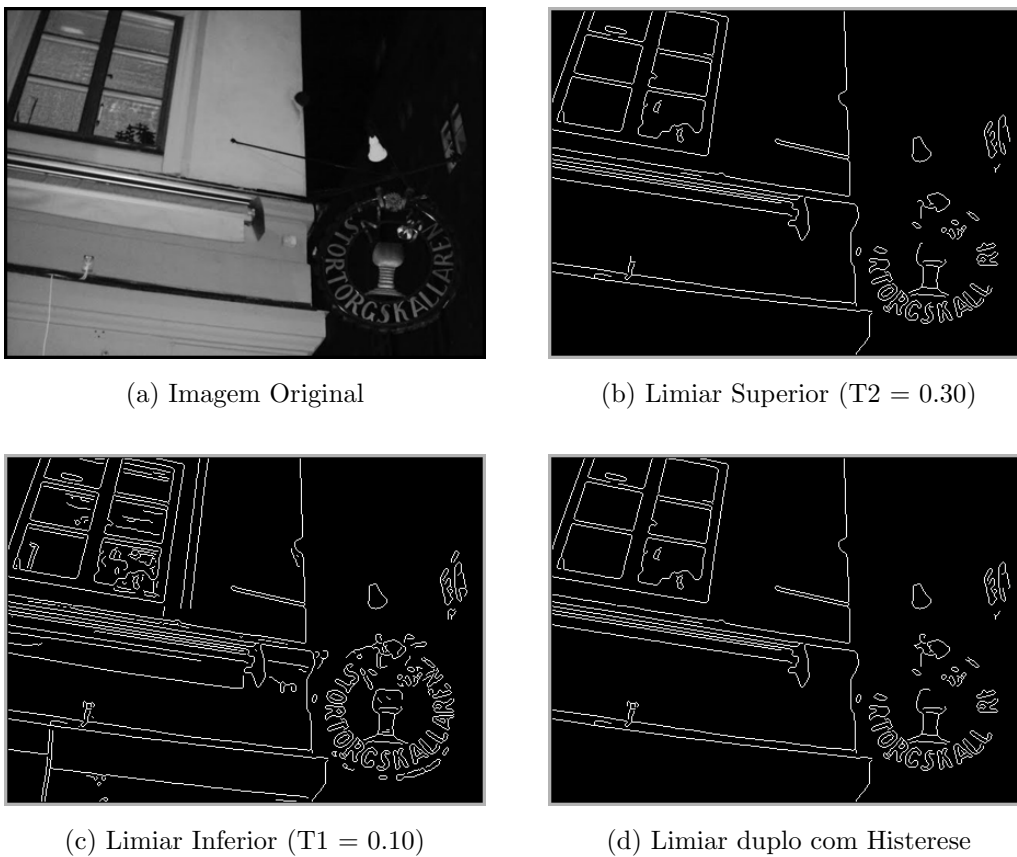


Figura 3.5: Comparação entre limiares do Detector de Bordas de Canny

O resultado dessa operação é uma imagem binária. Devido à resposta mínima o cálculo procede até que as bordas encontradas pelo detector apresentem um pixel de

espessura. A escolha dos valores dos *thresholds* T1 e T2 é realizada através de uma estimativa da razão ao sinal-ruído (SNR) da imagem.

A Figura 3.5 apresenta uma comparação entre a detecção de bordas da Figura 3.5a utilizando o *threshold* superior (T1), o inferior (T2) e a combinação dos dois limiares com histerese. A aplicação de um limiar muito baixo, como na Figura 3.5c, apresenta um grande número de bordas, pois neste caso o algoritmo está sujeito a detectar qualquer pequena magnitude dos gradientes direcionais.

3.3 Separando Segmentos

A saída de um Detector de Bordas de Canny é uma imagem binária, como mostra a Figura 3.1b, onde os pixels ‘1s’ (bordas) podem ser vistos em branco e os pixels ‘0s’ em preto. Porém a sensibilidade do detector muitas vezes apresenta bordas que não correspondem à região de interesse.

Observando a Figura 3.6, é possível distinguir visualmente quais bordas (em branco) correspondem ao duto que queremos analisar devido a sobreposição das bordas detectadas (Figura 3.1b) com a imagem real (Figura 3.1a). Se pudermos, de maneira computacional, encontrar na imagem binária os segmentos correspondentes às bordas do duto, poderemos então fazer uma análise sobre as curvas encontradas.

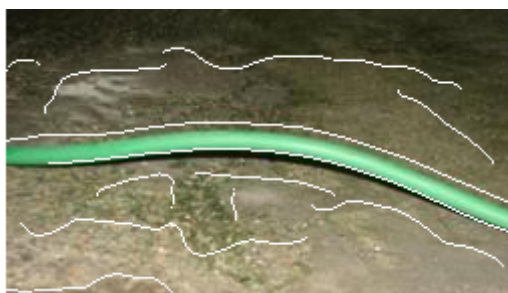


Figura 3.6: Imagem com Bordas Sobrepostas

Em imagens onde podemos identificar nitidamente a tubulação, os segmentos que representam as bordas do duto tendem a ser contínuos, apresentando um comprimento considerável em relação a outras bordas sinalizadas pelo detector de bordas. Como um dos princípios do Detector de Bordas de Canny diz que a borda tem que ter espessura

mínima (ou seja, um pixel de largura) consideramos que os dois maiores segmentos da imagem binária pertencem a bordas da tubulação fotografada.

Um problema observado é quando a imagem possui objetos que não interessam para a análise da tubulação. Estes objetos podem gerar bordas bem definidas, e com comprimentos longos. Caso este segmento seja considerado uma das bordas principais, a sua análise pode gerar uma classificação equivocada por parte do classificador. As bordas destes objetos podem ser comparadas a ruídos, que interferem no objetivo principal das próximas etapas do projeto.

Consideramos parte de um segmento, todo pixel classificado como borda que possui um ou dois pixels vizinhos que também são bordas. O primeiro passo para identificar um segmento é encontrar, na imagem binária, quais pixels possuem apenas um vizinho classificado como borda. Tais pixels são os extremos de um segmento.

Os vizinhos de um pixel são todos os pixels que estão ao seu redor, seja acima ou abaixo, dos lados, ou nas quatro diagonais. Um pixel que não está na borda da imagem possui então oito vizinhos.

A Figura 3.7 apresenta uma imagem ampliada, de forma que podemos observar todos os 14 pixels de um segmento apresentado. Os quadrados em branco, com valor '1' são bordas, enquanto os pretos correspondem a pixels que podemos desconsiderar.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0
3	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	0
4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 3.7: Pixels de um Segmento

Na coordenada Linha 4, Coluna B, vemos um pixel branco, que possui apenas um vizinho (Linha 3, Coluna C) branco. Observamos o mesmo caso com o pixel da coordenada Linha 3, Coluna O. Ambos correspondem a extremos do segmento, e como

só existe um segmento na imagem, todos os outros pixels brancos possuem dois vizinhos.

Como uma imagem digital é limitada em número de pixels, e todos são acessíveis através de suas coordenadas, podemos armazenar os pixels '1' da imagem binária em grupos, separados por segmentos, armazenando todas as posições correspondentes a cada pixel do segmento. Após encontrar um dos extremos do segmento, deve-se percorrer o segmento pixel a pixel, encontrando os vizinhos que também são classificados como borda, até encontrar o outro extremo.

Tabela 3.1: Coordenadas correspondentes aos pixels do segmento da Figura 3.7

Coordenada do Pixel N	Nº de Vizinhos	Coordenadas dos Vizinhos com valor '1'	
4,B	1	-	3,C
3,C	2	4,B	3,D
3,D	2	3,C	3,E
3,E	2	3,D	3,F
2,F	2	3,E	2,G
2,G	2	2,F	2,H
2,H	2	2,G	2,I
2,I	2	2,H	2,J
2,J	2	2,I	2,K
2,K	2	2,J	2,L
2,L	2	2,K	2,M
3,M	2	2,L	3,N
3,N	2	3,M	3,O
3,O	1	3,N	-

A Tabela 3.1 mostra todos os pixels do segmento da Figura 3.7, e seus vizinhos que também são classificados como borda, ou seja, possuem valor '1'.

Computacionalmente, uma imagem digital pode ser considerada uma matriz de tamanho $M \times N$, onde M e N são os números inteiros, que representam o total de linhas e

colunas da matriz, respectivamente. Cada pixel da imagem é uma coordenada da matriz.

Utilizando métodos computacionais percorre-se a matriz, que representa a imagem, até encontrar um pixel '1'. Em seguida os pixels ao redor são analisados, para localizar o vizinho que também é uma borda. Os vizinhos de um pixel com coordenada (m,n) correspondem às oito variações possíveis de $(m-1,n-1)$ até $(m+1,n+1)$, diferente de (m,n) . A Figura 3.8 ilustra todas as coordenadas dos vizinhos de um pixel na posição (m,n) .

$m-1, n-1$	$m-1, n$	$m-1, n+1$
$m, n-1$	m, n	$m, n+1$
$m+1, n-1$	$m+1, n$	$m+1, n+1$

Figura 3.8: Vizinhança de um pixel

No exemplo da Figura 3.8, como o pixel central possui apenas um vizinho branco, sua coordenada é armazenada como um extremo de segmento. Logo em seguida o pixel da posição $(m+1, n-1)$ tem seus vizinhos analisados. Se apenas um vizinho branco for encontrado, classifica-se esta coordenada como o outro extremo do segmento. Havendo dois vizinhos com valor '1', descarta-se o que já foi analisado (m,n) , e o pixel $(m+1,n-1)$ é armazenado como a segunda posição do segmento.

O procedimento é repetido até que a posição do segundo extremo do segmento seja encontrada. Um contador é incrementado toda vez que uma coordenada é armazenada como parte da borda que é percorrida. Ao final do segmento o contador indica o número de pixels existentes de um extremo ao outro.

Após percorrer todo um segmento, ou seja, armazenar todas as coordenadas de uma sequência de pixels '1' ininterrupta, estas são apagadas da imagem binária (recebem valor zero). Este procedimento é feito para que, ao continuar percorrendo a imagem, não se detecte pixels que já foram armazenados como parte de um segmento, evitando

assim uma comparação, e análise dos vizinhos, desnecessária, diminuindo o tempo de processamento.

Em alguns casos, como em um cruzamento de bordas, um pixel pode ter mais de dois vizinhos. Para facilitar o processamento tais pixels são apagados, recebendo valor zero. O mesmo é feito para bordas que não possuem vizinhos. Estes dois casos são analisados em uma primeira varredura da imagem. Em uma segunda passagem, buscamos os pixels brancos com apenas um vizinho da mesma cor.

Os pixels de valor '1', com mais de dois vizinhos com o mesmo valor, são apagados, pois seria necessário desenvolver um projeto de tomada de decisão para escolher qual caminho o segmento deveria seguir. Como muitos dos casos tratados são tubulações que apresentam curvas, não podemos a priori escolher um lado para seguir com base apenas no comportamento que a borda apresentou de um extremo até o cruzamento.

Como o comprimento dos segmentos é conhecido, podemos descartar bordas consideradas pequenas e ordená-las da maior para a menor. Desta forma é possível reduzir o número de bordas da seguinte maneira:

- Considerando apenas segmentos maiores que um determinado comprimento, é possível evitar ruídos desnecessários em um processamento posterior;
- Selecionando as maiores bordas da imagem podemos classificar um determinado objeto com maior confiabilidade.

3.3.1 Segmentos Fechados

O procedimento apresentado acima não encontra um segmento fechado. Porém, como os pixels dos segmentos encontrados anteriormente são apagados, restam apenas, caso existam, pixels brancos pertencentes a bordas fechadas. Portanto, qualquer pixel com valor '1', terá dois vizinhos com o mesmo valor.

De posse desta informação, percorre-se novamente a imagem pixel a pixel e, ao encontrar um pixel branco, podemos localizar todas as coordenadas do segmento fechado. A posição deste primeiro pixel '1' é armazenada, e ele é apagado da imagem. Em seguida, qualquer um dos dois vizinhos brancos pode ser escolhido.

Após escolher um dos vizinhos brancos, temos um segmento aberto, como no caso anterior. O procedimento é repetido até que todas as coordenadas da borda fiquem armazenadas, e a posição receba o valor zero.

Como o procedimento de percorrer a imagem foi interrompido ao encontrar um pixel branco, podemos retomá-lo da posição em que foi paralisado, pois tal posição é conhecida. Uma vez que todos os pixels do segmento fechado, encontrado anteriormente, agora possuem valor zero, estes não interferirão mais no procedimento quando se percorre a imagem.

Quando a imagem for completamente percorrida, todos os segmentos, abertos ou fechados estarão armazenados. Como o comprimento de cada borda é conhecido, podemos então ordená-las pelo comprimento, usando um algoritmo de ordenação.

3.4 Inclinações dos Segmentos

Conhecendo a posição de cada pixel de uma borda, é possível criar uma reta imaginária entre dois pontos do segmento, e calcular sua inclinação. Estabelecendo um intervalo P de pixels, podemos calcular diversas inclinações ao longo de uma borda.

Todo pixel é acessível por uma posição (x, y) . A distância entre dois pixels pode ser expressa utilizando a distância horizontal (ΔX) e a vertical (ΔY) entre eles, como observamos na Figura 3.9a, onde:

$$\Delta X = x_2 - x_1 \quad (3.3)$$

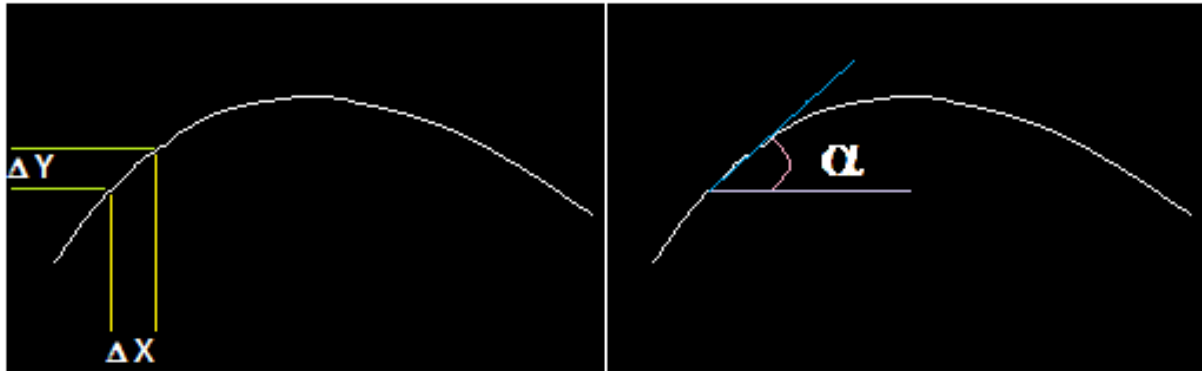
$$\Delta Y = y_2 - y_1 \quad (3.4)$$

A distância Euclidiana, ou distância métrica, entre esses dois pontos é dada por:

$$\Delta D = \sqrt{\Delta X^2 + \Delta Y^2} \quad (3.5)$$

A inclinação entre dois pontos (x_1, y_1) e (x_2, y_2) , de uma borda, é dada através do ângulo (α) , relacionado à razão entre as distâncias X e Y , da seguinte forma:

$$\alpha = \arctan \left(\frac{\Delta Y}{\Delta X} \right) \quad (3.6)$$



(a) Distâncias Horizontal e Vertical entre dois pontos

(b) Inclinação entre dois pontos

Figura 3.9: Inclinações em um Segmento

A Figura 3.9b ilustra o ângulo α , que calculamos acima, e representa a inclinação entre os pontos selecionados na Figura 3.9a.

As posições dos pixels são armazenadas de forma sequencial, ou seja, a primeira posição corresponde a um dos extremos, e em seguida é representado o seu vizinho, prosseguindo desta forma até o outro extremo. Esta informação facilita percorrer o segmento.

São usados dois ponteiros, um apontando para a posição do primeiro pixel e outro para a posição do pixel ΔP . Calcula-se α , através das distâncias ΔX e ΔY entre as coordenadas dos pixels, e o valor do ângulo é armazenado. Em seguida os ponteiros são incrementados e passam a apontar para o segundo pixel e para a posição $\Delta P + 1$. O ângulo de inclinação é calculado e armazenado. O procedimento é repetido até o ponteiro que começa em ΔP apontar para a última posição da borda, que em um segmento aberto é o outro extremo. No caso de um segmento fechado, os pixels da primeira e da última posição serão vizinhos.

De posse de todas as inclinações calculadas entre os dois extremos de um segmento, podemos criar um histograma, onde a amplitude indica a quantidade de vezes que cada ângulo ocorreu naquela borda. O histograma com os ângulos de inclinação do segmento é uma representação gráfica de como estes ângulos ocorrem em uma determinada curva.

Para facilitar uma futura comparação, normalizamos a amplitude entre zero e um, dividindo pelo número total de ângulos medidos. Os ângulos são quantizados em

360 posições, no intervalo $[-\pi; +\pi)$ radianos. Desta maneira podemos observar alguns padrões que representam as diversas formas que uma tubulação pode apresentar. A Figura 3.10 apresenta o histograma de inclinações da borda da Figura 3.9, com ΔP igual a 20.

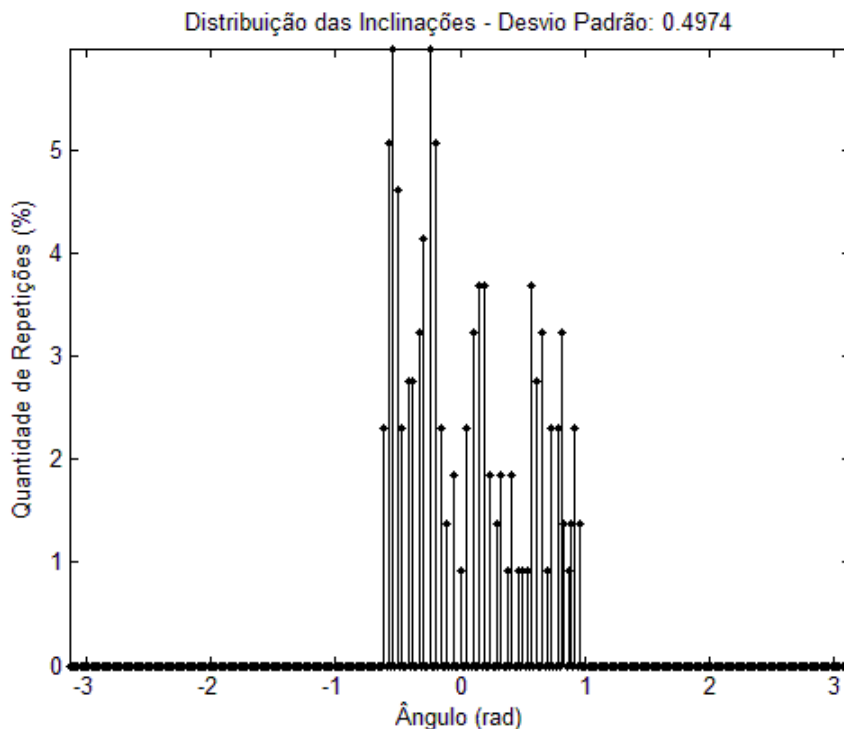


Figura 3.10: Histograma com as inclinações da curva Figura 3.9

Com os dados do histograma é possível calcular seu desvio padrão (σ).

$$\sigma = \frac{1}{N-1} \sqrt{\sum_{i=1}^N (\alpha_i - \bar{\alpha})^2} \quad (3.7)$$

onde N é o número de elementos e $\bar{\alpha}$ é o valor médio dos ângulos, calculado por:

$$\bar{\alpha} = \frac{1}{N} \sum_{i=1}^N \alpha_i \quad (3.8)$$

3.5 Comparação de Segmentos

Podemos comparar um histograma recém calculado com histogramas de uma base de dados, calculada previamente, e assim classificá-lo, de acordo com a semelhança entre

os histogramas. Este procedimento será explicado mais adiante. No momento, é importante observar um detalhe sobre o cálculo do ângulo de inclinação entre dois pontos em uma imagem.

O ângulo entre dois pontos em uma imagem, calculado pela Equação (3.6), é dependente da relação $\Delta Y/\Delta X$. Quando viramos a imagem em 90° , o que era ΔX passa a ser o novo ΔY . Desta forma, o novo ângulo será dependente da razão $\Delta X/\Delta Y$, da imagem original. Observamos a seguinte relação:

$$\alpha_2 = \arctan\left(\frac{\Delta Y_2}{\Delta X_2}\right) = \arctan\left(\frac{\Delta X_1}{\Delta Y_1}\right) \quad (3.9)$$

Como a imagem foi girada em 90° , ou $\frac{\pi}{2}$ radianos, α_2 pode ser expresso por:

$$\alpha_2 = \arctan\left(\frac{\Delta Y_1}{\Delta X_1}\right) - \frac{\pi}{2} \quad (3.10)$$

$$\alpha_2 = \alpha_1 - \frac{\pi}{2} \quad (3.11)$$

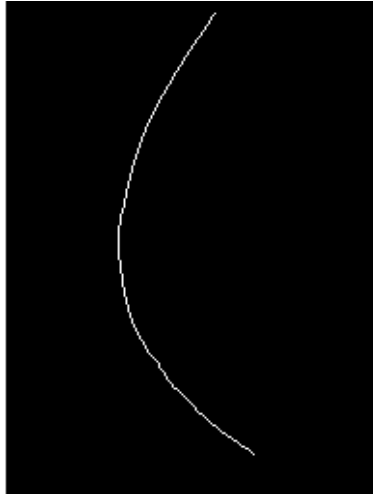
onde α_1 , ΔX_1 e ΔY_1 são respectivamente o ângulo e as distâncias horizontal e vertical entre dois pixels da imagem original, enquanto os termos com índice 2 correspondem a imagem girada em 90° .

A Figura 3.11a apresenta a imagem original, analisada anteriormente, com uma rotação de 90° . Na Figura 3.11b, encontra-se o histograma referente a esta nova curva.

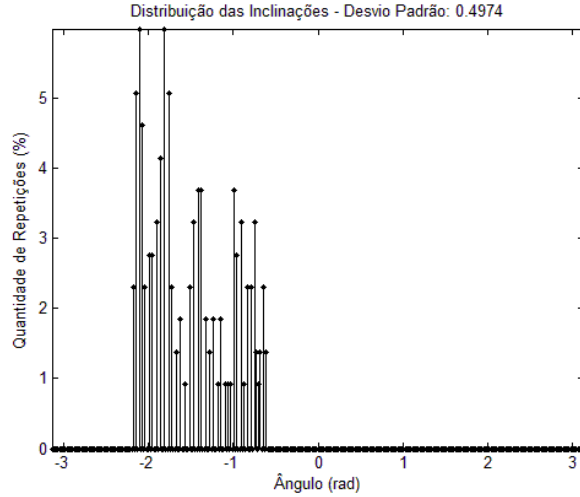
Se compararmos ponto a ponto o histograma da Figura 3.11b com o da Figura 3.10, encontraremos uma grande diferença entre eles. Desta forma, usando este dado para comparar as curvas, um classificador diria que as curvas são diferentes, enquanto sabemos que elas correspondem à mesma imagem, sendo que uma das imagens sofreu rotação de 90° .

Observando os dois histogramas, é possível concluir com facilidade que eles apresentam a mesma forma e o desvio padrão de ambos possui o mesmo valor. Com uma análise um pouco mais detalhada, é possível observar que as torres do segundo histograma estão deslocadas para a esquerda em 90 posições. Ou seja, o impulso que inicialmente estava em 0 radiano, agora está no valor $-\frac{\pi}{2}$ radianos.

Se a foto de mesmo duto for tirada com a posição da câmera girada em alguns graus, encontraremos histogramas diferentes, e esta imagem girada poderá ser atribuída



(a) Curva com rotação de 90°



(b) Histograma com os valores de α_2

Figura 3.11: Inclinações em um segmento com rotação de 90°

a uma classe incorreta. Como o objetivo deste trabalho é analisar e comparar tubulações através de imagens digitais, não é possível permitir que este erro ocorra.

Para corrigir o deslocamento de um histograma, teríamos que saber o ângulo (θ) de rotação que a imagem sofreu, e recalculamos os ângulos de inclinações (α) da seguinte forma:

$$\alpha = \arctan\left(\frac{\Delta Y}{\Delta X}\right) - \theta \quad (3.12)$$

Uma solução para resolver este problema seria calcular a média do histograma e deslocá-lo de forma que seu novo valor médio seja zero. Assim, duas curvas iguais, ou similares, estariam sempre centradas na origem, independentemente de uma das imagens ter sofrido rotação em relação à outra.

3.5.1 DFT dos Segmentos

Se considerarmos o histograma de inclinações, calculado anteriormente, como uma sequência periódica de 360 amostras impulsivas, podemos calcular a Transformada Discreta de Fourier (DFT) desta sequência, que conforme descrito em livros como [11] e [12]

é calculada por

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi kn}{N}} \quad (3.13)$$

onde $x[n]$ corresponde à sequência ou função discreta da qual queremos realizar a transformada discreta, enquanto N é igual ao número de amostras, neste caso 360. O resultado da DFT apresenta uma sequência de números complexos de tamanho N .

Uma propriedade interessante da DFT, que devemos observar, é a de deslocamento no tempo. Quando uma sequência $x[n]$ sofre um atraso de M amostras, a DFT desta sequência apresentará uma variação apenas na fase. Sendo $X_1[k]$ e $X_2[k]$ as DFTs de $x[n]$ e $x[n-M]$, respectivamente, onde k é um número inteiro entre zero e 359, temos:

$$X_1[k] = \sum_{n=0}^{359} x[n] e^{-j\frac{2\pi kn}{360}} \quad (3.14)$$

$$X_2[k] = \sum_{n=0}^{359} x[n-M] e^{-j\frac{2\pi kn}{360}} = \sum_{m=0}^{359} x[m] e^{-j\frac{2\pi k(m+M)}{360}} \quad (3.15)$$

Logo,

$$X_2[k] = e^{-j\frac{2\pi kM}{360}} X_1[k]. \quad (3.16)$$

Desta forma, o módulo de X_2 , é dado por

$$|X_2[k]| = |X_1[k]|. \quad (3.17)$$

Ou seja, o módulo da sequência $x[n]$ é igual ao da sequência $x[n-M]$. Como os módulos de X_2 e X_1 são iguais, podemos comparar as sequências circulares $x[n-M]$ e $x[n]$ utilizando os módulos de suas DFTs. Utilizando o módulo da Transformada Discreta de Fourier de uma sequência circular é possível realizar a comparação entre dois histogramas sem a necessidade de deslocá-los colocando a média em zero. Para qualquer k , $X[k]$ contém informação de todos os pontos da sequência $x[n]$.

A primeira amostra do módulo da Transformada Discreta de Fourier corresponde ao valor DC da sequência $x[n]$. Como no caso de interesse as funções são sempre histogramas com 360 posições, normalizadas para que a soma de todos os valores seja um, o valor DC de qualquer um dos histogramas será sempre um.

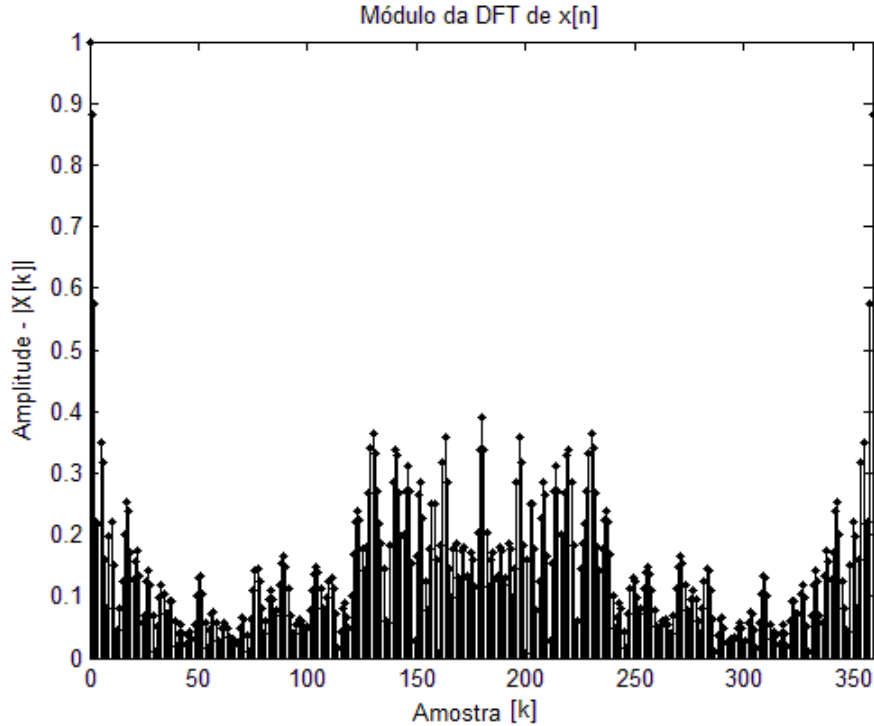


Figura 3.12: Módulo da DFT do Histograma da Figura 3.10

O módulo de uma DFT é uma função par: excluindo a primeira amostra, as demais são espelhadas em relação ao centro. A Figura 3.12 apresenta o módulo da Transformada Discreta de Fourier do Histograma da Figura 3.10, onde observa-se uma simetria com relação ao centro. Como existem 360 valores para k , e em k igual a zero temos o valor DC da sequência $x[n]$, podemos considerar o centro em k igual a 180. Sendo $X[k]$ uma função par, os valores de $X[k]$ para dois k equidistantes de 180 apresentam o mesmo valor.

3.5.2 Comparação das DFTs

O objetivo do algoritmo de comparação de segmentos é classificar uma curva de um duto de acordo com classes pré-estabelecidas. Os passos até a comparação foram explicados anteriormente. Primeiramente calculamos um histograma com os ângulos de inclinação ao longo do segmento que queremos classificar. Em seguida calcula-se o módulo da DFT do histograma, considerando-o como uma sequência periódica.

Supondo que esta operação descrita acima foi repetida para uma série de imagens,

onde as classes das tubulações fotografadas são conhecidas, os módulos das DFTs dos histogramas das inclinações dos segmentos são armazenados na memória do sistema para uma futura comparação com uma borda que desejamos classificar.

A comparação entre dois segmentos é feita da seguinte maneira. Após realizar as etapas descritas no item anterior, calcula-se a distância Euclidiana ponto a ponto, entre $X[k]$ referente ao segmento que desejamos classificar e $Y_i[k]$ referente a um segmento que tem uma classe conhecida. As distâncias encontradas ponto a ponto são somadas. A equação que descreve esta operação é apresentada da seguinte forma:

$$D_i = \sum_{k=A}^B (|X[k]| - |Y_i[k]|)^2 \quad (3.18)$$

onde A e B descrevem um intervalo de amostras que serão usadas para estimar a distância euclidiana entre X e Y_j .

Como a amostra em k igual a zero corresponde ao valor DC, podemos descartá-la da comparação. Também podemos desprezar os valores de k maiores que 180, uma vez que o módulo de uma DFT é uma função par. Assim, é possível fazer uma comparação sem utilizar todos os pontos, atribuindo para A e B os valores um e 180 respectivamente. Podemos reescrever a Equação (3.18) com estes números:

$$D_i = \sum_{k=1}^{180} (|X[k]| - |Y_i[k]|)^2 \quad (3.19)$$

Para encontrar a classe à qual uma borda pertence, devemos calcular o valor de D_i entre o X da borda que queremos classificar e o Y_i de cada um dos segmentos armazenados na memória do sistema. A curva com o menor D_i encontrado é considerada a mais próxima da borda analisada, por isso a borda referente a X recebe a mesma classificação que o segmento do histograma de índice i .

Capítulo 4

Testes e Resultados

4.1 Classes dos Dutos Analisados

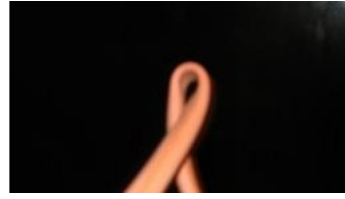
O objetivo deste projeto consiste em classificar determinadas tubulações de acordo com classes pré-estabelecidas. Neste trabalho foram utilizadas cinco classes, conhecidas pelos nomes de ‘Corcova’, ‘Kink’, ‘Loop’, ‘Reto’ e ‘Torção’. Estas nomenclaturas são referentes às estruturas físicas apresentadas ao longo de dutos flexíveis. A Figura 4.1 apresenta um exemplo de cada uma das classes.

- **Corcova** - Ocorre quando uma tubulação apresenta um desnível, com formato de uma curvatura convexa ao longo de sua extensão.
- **Kink** - Consiste de uma dobra muito fechada, parecida como um nó, ao longo da tubulação.
- **Loop** - Quando ao longo da extensão de um duto existe um laço.
- **Reto** - Classe simples, onde uma tubulação está esticada.
- **Torção** - Um tubo torcido ao longo de seu eixo longitudinal.

A classificação é feita comparando as bordas dos dutos, conforme explicado no capítulo anterior. Para isso aplicamos o detector de bordas em diversas imagens, e separamos as principais bordas. Para criar um banco de dados para o sistema, separamos de



(a) Corcova



(b) Kink



(c) Loop



(d) Reto



(e) Torção

Figura 4.1: Classes consideradas

um grupo de imagens as bordas que foram possíveis classificar corretamente utilizando apenas a visão humana, e a armazenamos na memória do sistema.

Como um dos objetivos neste projeto de Visão Computacional é substituir um trabalho de identificação, realizado pelo sistema visual humano, devemos partir do princípio que os dados utilizados no projeto são classificados corretamente por nós, e não apenas pelos métodos computacionais aplicados. Depois de separarmos e armazenarmos as principais bordas é possível comparar uma determinada imagem com as armazenadas.

Para eliminar qualquer segmento que possa gerar uma classificação errada, podemos repetir a classificação de cada uma das bordas já selecionadas, levando em consideração a borda nova que pretendemos acrescentar à memória do sistema. Se a com-

paração de uma nova imagem com as demais da memória não resultar na classificação correta, podemos descartar esta borda, evitando que este segmento possa gerar uma classificação errada.

4.2 Parâmetros Utilizados

Visando encontrar os melhores parâmetros para foram testados mais de 100 diferentes combinações de thresholds e detectores de bordas. O Detector de Bordas Canny apresentou os melhores resultados entre os detectores e alguns parâmetros de entrada apresentaram precisão e continuidade na detecção de bordas para o grupo de imagens de interesse.

O Detector de Bordas de Canny necessita basicamente de três parâmetros de entrada o limiar superior (T_2) e o inferior (T_1), além do desvio padrão (σ) do filtro Gaussiano. Conforme explicado na Seção 3.2 a escolha dos limiares é responsável por detectar as bordas de interesse, ignorando bordas falsas provocadas por ruído e aceitando bordas fracas próximo a bordas mais fortes. Enquanto que um valor maior de σ tende a reduzir a influência do ruído, porém pode prejudicar a detecção de bordas fracas que possam ser de interesse.

A escolha desses parâmetros é fundamental para o bom andamento do processo, pois uma vez que as bordas mais significativas não coincidam com as bordas da tubulação que desejamos classificar, esta borda gerará uma classificação aleatória e não um resultado preciso com base nas informações das bordas do duto.

Na Tabela 4.1 são apresentados alguns parâmetros com os quais, de maneira geral, obtivemos sucesso na detecção de bordas dos dutos. A combinação que apresentou melhores resultados encontra-se na última linha da tabela.

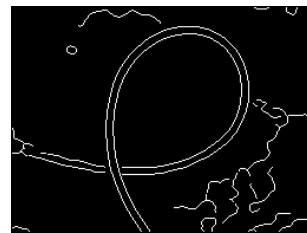
Para facilitar e padronizar o sistema, os valores $T_1 = 0.12$, $T_2 = 0.30$, $\sigma = 5$ foram adotados para o cálculo e detecção de todas as bordas utilizando o Detector de Bordas de Canny. A Imagem 4.2 apresenta uma boa detecção de bordas de um duto com Loop. Na Imagem 4.3 observamos a detecção de bordas de uma tubulação com torção. Nos dois casos foram utilizados os melhores parâmetros encontrados.

Tabela 4.1: Parâmetros do Detector de Borda de Canny

T_1	T_2	σ
0.04	0.10	1.00
0.10	0.40	1.00
0.01	0.04	2.00
0.08	0.20	2.00
0.08	0.20	3.00
0.12	0.30	5.00



(a) Imagem Original

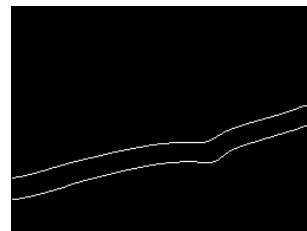


(b) Bordas Encontradas

Figura 4.2: Imagem de Loop segmentada com $T_1 = 0.12$, $T_2 = 0.30$ e $\sigma = 5$



(a) Imagem Original



(b) Bordas Encontradas

Figura 4.3: Imagem de Torção segmentada com $T_1 = 0.12$, $T_2 = 0.30$ e $\sigma = 5$

4.3 Composição da Base de Dados

Inicialmente foram utilizadas 140 imagens, separadas entre as cinco classes consideradas na seção anterior. Algumas imagens precisaram ser descartadas, pois não apresentavam boas condições para o detector de bordas, devido a interferências externas nas imagens, como por exemplo, um fundo xadrez. Estas imagens são submetidas ao Detector

de Bordas de Canny e em seguida têm seus segmentos separados, conforme descrito nas Seções 3.2 e 3.3.

Depois desta pré-seleção, 84 imagens foram aprovadas, gerando 146 segmentos, que podiam ser classificados visualmente, sem auxílio de uma ferramenta de decisão eletrônica. Todos os segmentos selecionados passaram pelo processo de cálculo do histograma de inclinações descrito na Seção 3.4. Depois disso, calculou-se a DFT de cada uma dessas seqüências.

Desta forma, foi criada uma matriz de tamanho 360×146 , onde cada coluna apresenta a DFT do histograma de inclinações de um dado segmento. A Tabela 4.2 apresenta como esta matriz é organizada. Podemos descartar a amostra correspondente ao nível DC da DFT, e toda a metade inferior da matriz, armazenando uma matriz com apenas 180 linhas.

Tabela 4.2: Organização da Matriz com as DFTs

Classe	Corcova	Kink	Loop	Reto	Torção
Coluna Inicial	1	25	49	74	115
Coluna Final	24	48	73	114	146
Total de Segmentos	24	24	25	41	32

Para realizar a comparação entre os segmentos selecionados devemos comparar cada coluna da matriz com as demais. Esta comparação é feita de acordo com os procedimentos descritos na Seção 3.5.2, calculando o quadrado da diferença linha a linha entre duas colunas, e somando todos os resultados. Devemos desconsiderar a comparação de uma coluna com sua própria imagem.

Após realizar o cálculo descrito acima, entre uma coluna X e todas as outras colunas Y_i , encontraremos diversos valores D_i . A coluna i que apresentar a menor distância determinará a classe. Espera-se que a classe encontrada ($D_i = \min(D_i)$) seja a mesma que a classificação visual determinou para este segmento. No caso da comparação de um Kink com todas as colunas da matriz, para que o sistema acerte a classificação, o menor valor encontrado deve corresponder a uma coluna entre 25 e 48.

Classificando computacionalmente as bordas, que foram separadas inicialmente, 127 segmentos receberam a mesma classe que lhe foi atribuída pela inspeção visual humana, o que corresponde a 87% de acertos. A partir destes dados, as colunas correspondentes aos segmentos confundidos foram retiradas da matriz.

Uma segunda rodada de comparações foi realizada, e dois segmentos não foram classificados corretamente. Suas respectivas colunas foram excluídas, e na terceira rodada de comparações, nenhum erro foi apresentado. Desta forma uma segunda matriz foi gerada, organizada de acordo com a Tabela 4.3. Ao todo 21 bordas foram descartadas, dentre as que inicialmente foram classificadas sem dificuldades ou dúvidas pelo sistema visual humano.

Tabela 4.3: Organização da Segunda Matriz com as DFTs

Classe	Corcova	Kink	Loop	Reto	Torção
Coluna Inicial	1	19	37	60	99
Coluna Final	18	36	59	98	125
Total de Segmentos	18	18	23	39	27

4.4 Comparação com Novos Segmentos

Após selecionar uma base de dados para comparação foram realizados testes com imagens que não foram utilizadas na etapa anterior. Ao todo foram aproveitadas 108 imagens, que tiveram suas respectivas classes distinguidas visualmente para realizarmos uma comparação através do sistema projetado.

De cada imagem foram extraídos dois segmentos com mais de 100 pixels de comprimento. No total foram comparados 215 segmentos, o que não corresponde ao dobro das imagens utilizadas, pois uma das imagens, da classe Kink, apresentou apenas uma única borda que não foi descartada pelos critérios de seleção.

4.4.1 Primeira Comparação

Na Tabela 4.4 podemos ver a relação de acertos e erros das 215 bordas comparadas com a base de dados da Tabela 4.2. Primeiramente, as imagens foram separadas em suas respectivas classes, e depois classificadas pelo sistema. A primeira coluna da Tabela 4.4 apresenta a classe real das tubulações. Nas demais colunas estão os resultados obtidos na saída do classificador, separados em relação ao maior e o segundo maior segmento (Nº Segmento 1 ou 2) da imagem.

Tabela 4.4: Resultados da Comparação com a Primeira Base de Dados

Nº Segmento	Resultados Encontrados											
	Corcova		Kink		Loop		Reto		Torção		Total	
	1	2	1	2	1	2	1	2	1	2	1	2
Corcova	15	14	1	1	2	2	0	0	2	3	20	20
Kink	0	7	8	5	13	4	0	0	0	4	21	20
Loop	0	2	1	4	21	14	0	0	0	2	22	22
Reto	1	0	0	0	0	0	19	21	4	3	24	24
Torção	3	2	0	0	0	0	4	3	14	16	21	21

Por exemplo, quando analisamos as imagens referentes à classe de dutos retos, a comparação classificou 19 das 24 imagens corretamente. Porém em um dos casos o classificador indicou uma corcova, enquanto por quatro vezes o duto foi classificado como torção. Para este caso, levamos em consideração apenas a classificação da maior borda da imagem. A classificação das segundas maiores bordas das imagens referentes a dutos retos resultou em 21 dutos retos e três torções. Na última coluna é indicado o total de segmentos classificados para cada grupo de imagens.

O motivo para o cálculo de duas bordas, a princípio, é meramente acadêmico, com a finalidade de comparar a variação entre os dois principais segmentos de uma tubulação. Cada duto apresenta no mínimo duas bordas. No caso ideal o detector de bordas encontraria estes dois segmentos, e ambos deveriam ser classificados como a mesma classe. Verificamos que tal fato não ocorre sempre, principalmente devido a falhas

na segmentação.

A Tabela 4.5 mostra o número de imagens que apresentaram duas bordas classificadas corretamente como a classe do duto naquela imagem. A classe Kink apresenta um grande número de erros. Esta classe possui um histograma de inclinações muito parecido com a classe Loop.

Tabela 4.5: Imagens com dois segmentos classificados corretamente

Classe	Corcova	Kink	Loop	Reto	Torção
Acertos	14	1	14	18	12
Total de Imagens	20	21	22	24	21

Um bom argumento para utilizar dois segmentos de um duto para classificá-lo pode ser baseado no fato da maior borda muitas vezes não ser classificada corretamente, porém a segunda principal sim. Neste caso o sistema de inspeção automática pode alertar para uma anomalia nos padrões esperados para a tubulação.

4.4.2 Segunda Comparação

Na Seção 4.2 é dada a sugestão de retirar da memória do sistema os segmentos classificados incorretamente em relação aos demais do banco de dados. O formato da matriz das DFTs resultantes é ilustrado na Tabela 4.3. O resultado da comparação dos segmentos do grupo das 108 novas imagens de teste é apresentado na Tabela 4.6.

A quantidade de acertos totais, somando primeiro e segundo segmentos, não resultou em uma melhora em relação à comparação feita anteriormente, e apresentada na Tabela 4.4. As classes Reto, Corcova e Torção, apresentaram uma classificação mais correta do maior segmento do que no caso anterior. A diferença é pouco significativa, para o pequeno espaço amostral que estamos analisando.

Os resultados de classificação de uma tubulação, utilizando todas as bordas de interesse da imagem, são apresentados na Tabela 4.7. Se utilizássemos os dois segmentos para determinar a classificação de um duto, mais uma vez não obteríamos bons resultados para a classe Kink. Porém classes como Corcova, Loop e Reto apresentam resultados

Tabela 4.6: Resultados da Comparação com a Segunda Base de Dados

Nº Segmento	Resultados Encontrados											
	Corcova		Kink		Loop		Reto		Torção		Total	
	1	2	1	2	1	2	1	2	1	2	1	2
Corcova	16	15	1	0	2	2	0	0	1	3	20	20
Kink	1	6	8	5	12	4	0	0	0	5	21	20
Loop	0	3	1	3	21	16	0	0	0	0	22	22
Reto	1	0	0	0	0	0	20	20	3	4	24	24
Torção	3	3	0	0	0	0	3	3	15	15	21	21

melhores do que os anteriores (Tabelas 4.4 e 4.5).

As Torções apresentaram 15 classificações corretas utilizando apenas a maior borda ou a segunda maior. Porém somente 12 de cada uma dessas bordas são pares de um duto. Concluimos então que ao todo 18 imagens receberam classificação correta, porém em seis desses casos ocorreu do sistema não concordar com as duas bordas analisadas.

Tabela 4.7: Imagens com todas as bordas classificadas corretamente

Classe	Corcova	Kink	Loop	Reto	Torção
Acertos	15	2	16	18	12
Total de Imagens	20	21	22	24	21

Quando procuramos compreender os erros apresentados na classificação dos segmentos, notamos que na grande maioria dos casos em que os segmentos que não foram classificados corretamente, a resposta à detecção de bordas dessas imagens apresenta baixa qualidade. Muitas bordas apresentam descontinuidades ou interferências que alteram a direção da borda na imagem binária.

Um erro devido a falhas na segmentação, podem ser vistos em um Loop classificado como Torção (Figura 4.4) na primeira classificação realizada. Em geral, as Torções classificadas como Corcova apresentavam curvaturas semelhantes às presentes na classe

Corcova (Figura 4.5), o que dificultou até mesmo a interpretação visual. Outro exemplo de má detecção observado consiste de um duto reto que foi classificado como Corcova (Figura 4.6), pois as bordas não mantiveram uma continuidade tangenciando a tubulação.

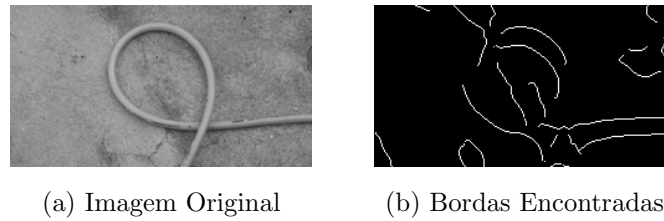


Figura 4.4: Loop classificado como Torção devido a má detecção de bordas

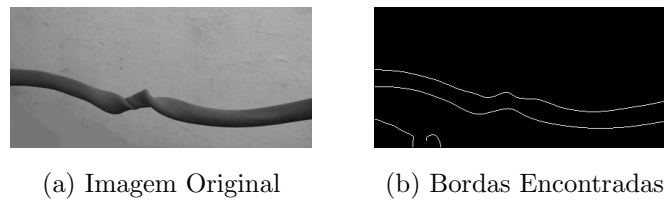


Figura 4.5: Torção classificada como Corcova devido a curvatura no duto

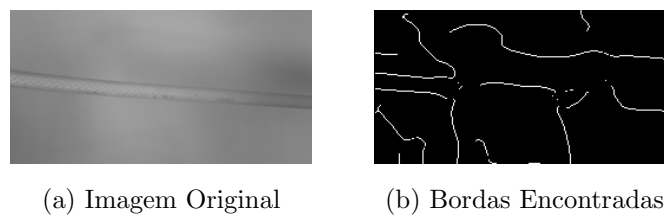


Figura 4.6: Duto Reto classificado como Corcova devido a falha de segmentação

Capítulo 5

Conclusão

Uma boa segmentação de imagens é fundamental para o sucesso de um projeto Visão Computacional. No caso específico deste trabalho, onde a detecção de bordas fornece o principal objeto para ser analisado e classificado pelo sistema, a boa execução desta operação praticamente define se a classificação de uma tubulação será efetuada corretamente.

Observamos que o principal motivo para erros na inspeção automática foi a análise incorreta de bordas que receberam interferência em sua detecção. O segmento submetido à classificação não apresentava as características corretas do duto que desejávamos classificar. Em geral, quando a resposta ao filtro de Canny [1] apresentou segmentos coerentes com as bordas da tubulação presente na imagem, os resultados do classificador estavam de acordo com o esperado.

Este projeto utilizou imagens ilustrativas para testar a funcionalidade do algoritmo que desejávamos realizar. Devido ao pequeno número de imagens disponíveis, e algumas vezes devido à baixa qualidade destas, a realização de testes mais profundos, capazes de determinar a real precisão do método implementado, ficou comprometida. Foi necessário se basear em dois tipos de comparações. A primeira foi feita entre as imagens utilizadas no bando de dados do sistema. E a segunda foi feita com um novo grupo de imagens, geradas para testar o método proposto. Pouco mais de 100 imagens foram utilizadas, enquanto a memória do sistema continha uma base de dados em torno de 140 segmentos.

Devemos observar que o objetivo deste trabalho é criar um sistema de Visão Com-

putacional, capaz de realizar uma inspeção automática. Partimos do princípio de que as imagens utilizadas estão em boas condições, capazes de gerar bons resultados quando apresentadas ao detector de bordas de Canny. O ideal seria que as imagens fossem inicialmente pré-processadas, com a finalidade de reduzir a interferência de ruídos e objetos externos à região de interesse no momento da filtragem para detectar as bordas.

A realização do algoritmo proposto foi considerada satisfatória. Conseguimos comparar e analisar a curvatura de tubulações através deste método. Os resultados estão corretos e a inspeção é feita de forma unicamente computacional, quando as condições são propícias. Os problemas apresentados ao longo do projeto indicam que antes de realizar qualquer alteração nos procedimentos do algoritmo, precisamos automatizar um sistema capaz de fornecer imagens em condições ótimas para este classificador. Assim, a realização dos processamentos descritos neste trabalho resultará em um máximo de sucesso.

Para trabalhos futuros poderíamos sugerir a criação de uma etapa de ligação de bordas através de um processamento local e da análise de similaridade dos pixels vizinhos. A análise de uma pequena região em torno da borda poderia encontrar uma continuidade dessa borda, resolvendo um problema de descontinuidade que acarreta em má classificação do duto. Outra possível abordagem seria o projeto de um filtro de pré-processamento para deixar a imagem em condições de obter bordas mais longas e precisas.

Referências Bibliográficas

- [1] CANNY, J., “A Computational Approach to Edge Detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 8, NO. 6, pp. 679–698, 1986.
- [2] JAIN, A. K., *Fundamentals of Digital Image Processing*. Upper Saddle River, NJ, Estados Unidos, Prentice-Hall, 1989.
- [3] PEDRINI, H., SCHWARTZ, W. R., *Análise de Imagens Digitais - Princípios Algoritmos e Aplicações*. 1 ed. São Paulo, SP, Thomson Learning, 2008.
- [4] GONZALEZ, R. C., WOODS, R. E., *Processamento de Imagens Digitais*. 1 ed. São Paulo, SP, Editora Edgard Blücher, 2006.
- [5] ROBERTS, L. G., “Machine Perception of Three Dimensional Solids”. In: *Optical and electro-optical information processing*, pp. 159–197, Cambridge, MA, Estados Unidos, 1965.
- [6] PREWITT, J. M. S., *Picture Processing and Psychopictorics*, chapter Object enhancement and extraction, Academic Press, pp. 75–149, 1970.
- [7] SOBEL, I., *Machine Vision for Three-Dimensional Scenes*, chapter An Isotropic 3x3 Image Gradient Operator, Academic Press, pp. 376–379, 1990.
- [8] KRISCH, R. A., “Computer Determination of the Constituent Structure of Biological Images”, *Computers and Biomedical Research*, v. 4, NO. 3, pp. 315–328, 1971.
- [9] ROBINSON, G. S., “Edge Detection by Compass Gradient Masks”, *Computer and Graphics Image Processing*, v. 6, NO. 5, pp. 492–501, 1977.

- [10] FREI, W., CHEN, C., “Fast Boundary Detection: A Generalization and New Algorithm”, *IEEE Transactions on Computers*, v. 26, NO. 10, pp. 988–998, 1977.
- [11] DINIZ, P. S. R., SILVA, E. A. B. D., NETTO, S. L., *Processamento Digital de Sinais - Projeto e Análise de Sistemas*, chapter Transformadas Discretas, Porto Alegre, RS, Bookman, pp. 101–156, 2004.
- [12] MITRA, S. K., *Digital Signal Processing - Computer Based Approach*, 2 ed., chapter Discrete-Time Signals in the Transform-Domain, McGraw-Hill, pp. 117–179.