



**FACULDADE DE TECNOLOGIA SENAI CIMATEC
ESPECIALIZAÇÃO EM AUTOMAÇÃO, CONTROLE E ROBÓTICA**

LAUÊ RAMI SOUZA COSTA DE JESUS

**DESENVOLVIMENTO E IMPLEMENTAÇÃO DE UM SISTEMA PARA O
RECONHECIMENTO DE PADRÕES DE CARACTERES DE MOSTRADORES
DIGITAIS DE AMBIENTES INDUSTRIAIS**

Salvador
2017

LAUÊ RAMI SOUZA COSTA DE JESUS

**DESENVOLVIMENTO E IMPLEMENTAÇÃO DE UM SISTEMA PARA O
RECONHECIMENTO DE PADRÕES DE CÂRACTERES DE MOSTRADORES
DIGITAIS DE AMBIENTES INDUSTRIAIS**

Monografia apresentada à Faculdade de Tecnologia SENAI CIMATEC como requisito parcial para a obtenção do Título de Especialista em Automação, Controle e Robótica.

Professor Orientador: Oberdan Rocha Pinheiro

Salvador
2017

Ficha catalográfica elaborada pela Biblioteca da Faculdade de Tecnologia SENAI CIMATEC

J58d Jesus, Lauê Rami Souza Costa de

Desenvolvimento e implementação de um sistema para o reconhecimento de padrões de caracteres de mostradores digitais de ambientes industriais / Lauê Rami Souza Costa de Jesus. – Salvador, 2017.

64 f. : il. color.

Orientador: Prof. Dr. Oberdan Rocha Pinheiro.

Monografia (Especialização em Automação, Controle e Robótica) – Programa de Pós-Graduação, Faculdade de Tecnologia SENAI CIMATEC, Salvador, 2017.

Inclui referências.

1. Reconhecimento de padrões de caracteres. 2. OpenCV. 3. Tesseract OCR.
4. Processamento digital de imagens. I. Faculdade de Tecnologia SENAI CIMATEC. II. Pinheiro, Oberdan Rocha. IV. Título.

CDD: 629.8

LAUÊ RAMI SOUZA COSTA DE JESUS

**DESENVOLVIMENTO E IMPLEMENTAÇÃO DE UM SISTEMA PARA O
RECONHECIMENTO DE PADRÕES DE CARACTERES DE MOSTRADORES
DIGITAIS DE AMBIENTES INDUSTRIAIS**

Projeto Final de Curso aprovado com nota 9,0 (nove) como requisito de Especialista em Automação, Controle e Robótica, tendo sido julgado pela Banca Examinadora formada pelos Professores:

Dr. Oberdan Rocha Pinheiro – OrientadorDoutor
Faculdade de Tecnologia SENAI CIMATEC

Msc. Milton Bastos de Souza - Professor
Faculdade de Tecnologia SENAI CIMATEC

Salvador, 16 de maio de 2017

AGRADECIMENTOS

Agradeço a Deus por ter me dado forças para superar as dificuldades e nunca desistir.

Ao SENAI-CIMATEC pela oportunidade de crescimento e apoio sempre concedido.

Ao meu orientador Oberdan Rocha Pinheiro pela compreensão, compromisso e dedicação que conduziu o processo de orientação para a construção desse trabalho.

RESUMO

Este trabalho apresenta o desenvolvimento e implementação de um sistema para o reconhecimento de padrões de caracteres de mostradores digitais de ambientes industriais. Nesta monografia são detalhados os estudos sobre as etapas fundamentais do processamento digital de imagens desde a aquisição de imagens até o reconhecimento de objetos, reconhecimento óptico de caracteres, técnicas e aplicações da biblioteca OpenCV e aplicação da biblioteca Tesseract OCR. A metodologia de desenvolvimento deste trabalho mostra como o embasamento teórico é aplicado na elaboração do sistema. Serão abordadas as ferramentas utilizadas no desenvolvimento, bem como a arquitetura definida. Como resultados, serão apresentadas imagens com a representação das simulações geradas a partir do código elaborado na linguagem C++. Todo o processo desde aquisição das imagens até o reconhecimento de padrões de caracteres será abordado.

Palavras-chave: Reconhecimento de Padrões de Caracteres. OpenCV. Tesseract OCR. Processamento Digital de Imagens.

ABSTRACT

This work presents the development and implementation of a characters patterns recognition system of digital displays of industrial environments. In this report are detailed studies on the fundamental stages of image processing, from image acquisition to object recognition, optical character recognition, techniques and applications of the OpenCV library and application of the Tesseract OCR library. The development methodology of this work shows how the theoretical basis is applied in the elaboration of the system. In this report will be present the tools used to development, as well as the defined architecture. The results will be explain with images of the representation of the simulations generated from the code elaborated in the C ++ language. The entire process from the acquisition of images to the characters patterns recognition will be addressed.

Key words: Characters Patterns Recognition. OpenCV. Tesseract OCR. Digital Image Processing.

LISTA DE ILUSTRAÇÕES

Figura 1 - Taxol - Agente anticâncer	14
Figura 2 - Aquisição de Imagens	15
Figura 3 - Etapas fundamentais no processamento digital de imagens	17
Figura 4 - Processo de Amostragem 1º parte	18
Figura 5 - Processo de Amostragem 2º parte	18
Figura 6 - Processo de Quantização	19
Figura 7 - Salt-and-pepper noise	21
Figura 8 - Resultado da filtragem mediana	21
Figura 9 - Função gaussiana	22
Figura 10 - Aplicação da filtragem gaussiana	23
Figura 11 - Operação	23
Figura 12 – Tiepoints	25
Figura 13 - Conjunto A	27
Figura 14 - Conjunto B	27
Figura 15 - Dilatação de A por B	28
Figura 16 - Operação de Translação	28
Figura 17 - Imagem Fonte	29
Figura 18 - Operação de Abertura	30
Figura 19 - Imagem Fonte	30
Figura 20 - Operação de Fechamento	30
Figura 21 - Máscara 3x3	31
Figura 22 - Máscaras 3x3	32
Figura 23 - Imagem Fonte	34
Figura 24 - Detecção de Borda	34
Figura 25 – Histograma	35
Figura 26 - Arquitetura do Sistema	39
Figura 27 - Fluxograma de Processos	43
Figura 28 - Novus N1540 à esquerda, Orion Versa Star Pro à direita	54
Figura 29 - Tons de Cinza – N1540 à esquerda, Orion Versa Star Pro à direita	55
Figura 30 - Limiarização – N1540 à esquerda, Orion Versa Star Pro à direita	55
Figura 31 - Filtragem mediana equipamento Orion Versa Star Pro	56

Figura 32 - Gaussiana – N1540 à esquerda, Orion Versa Star Pro à direita	56
Figura 33 - Dilatação N1540 à esquerda, Orion Versa Star Pro à direita	57
Figura 34 - Contornos N1540 à esquerda, Orion Versa Star Pro à direita	57
Figura 35 - Retângulos N1540 à esquerda, Orion Versa Star Pro à direita	58
Figura 36 - Objetos de Interesse na Imagem equipamento N1540	58
Figura 37 - Objetos de Interesse na Imagem equipamento Orion Versa Star Pro	59
Figura 38 - 1º Operação de fechamento equipamento N1540	59
Figura 39 - 2º Operação de fechamento equipamento N1540	59
Figura 40 - Sequência de operações equipamento Orion Versa Star Pro	60
Figura 41 - Resultado após 5 iterações equipamento Orion Versa Star Pro	60
Figura 42 - Reconhecimento - N1540 à esquerda Orion Versa Star Pro à direita	61

LISTA DE ABREVIATURAS E SIGLAS

CCD	Charge-Coupled Device
CDT	C/C++ Development Tooling
GCC	GNU Compiler Collection
HP	Hewlett-Packard
IDE	Integrated Development Environment
JPEG	Joint Photographic Experts Group
LCD	Liquid Crystal Display
MinGW	Minimalist GNU for Windows
OCR	Optical Character Recognition
PNG	Portable Network Graphics
RGB	Red-Green-Blue
UTF-8	8 - bit Unicode Transformation Format

SUMÁRIO

INTRODUÇÃO	11
CAPÍTULO I - REFERENCIAL TEÓRICO PARA A RESOLUÇÃO DO PROBLEMA	12
1.1 PROCESSAMENTO DIGITAL DE IMAGENS	12
1.1.1 Exemplos de Aplicações de Processamento de Imagens	13
1.1.2 Componentes de um Sistema de Processamento de Imagens	14
1.1.3 Etapas fundamentais no Processamento Digital de Imagens	15
1.2 AMOSTRAGEM E QUANTIZAÇÃO DA IMAGEM	17
1.2.1 Representação de Imagens Digitais	19
1.3 MELHORIA DE IMAGEM	20
1.3.1 Filtragem Mediana	20
1.3.2 Filtragem Gaussiana	21
1.3.3 Melhoria de Imagem usando operações Lógicas e Aritméticas	23
1.4 RESTAURAÇÃO DE IMAGEM	24
1.4.1 Transformações Geométricas	24
1.5 PROCESSAMENTO MORFÓLOGICO DE IMAGEM	26
1.5.1 Dilatação	27
1.5.2 Erosão	28
1.5.3 Abertura e Fechamento	29
1.6 SEGMENTAÇÃO DE IMAGEM	31
1.6.1 Detecção de Descontinuidade	31
1.6.2 Detecção de Ponto	32
1.6.3 Detecção de Linha	32
1.6.4 Detecção de Borda	33
1.6.5 Thresholding	35
1.6.6 Reconhecimento de Objetos	35
1.6.6.1 Reconhecimento Óptico de Caracteres	36
CAPÍTULO II – METODOLOGIA DO DESENVOLVIMENTO DO PROJETO	38
2.1 DEFINIÇÃO DE ARQUITETURA DO SISTEMA	38
2.2 OPENCV	39

2.3 TESSERACT OCR	40
2.4 LINGUAGEM DE PROGRAMAÇÃO C++	41
2.5 ECLIPSE CDT	41
CAPÍTULO III – DESENVOLVIMENTO DO PROJETO	42
3.1 DESENVOLVIMENTO DO SISTEMA	42
3.1.1 Definição dos Parâmetros dos Equipamentos Industriais	43
3.1.2 Processo de Aquisição da Imagem	44
3.1.3 Binarização da Imagem	45
3.1.4 Procedimento de Melhoria da Imagem	46
3.1.5 Procedimentos Específicos dos Equipamentos Industriais	47
3.1.6 Reconhecimento de Padrões dos Caracteres	52
3.2 RESULTADOS	53
3.2.1 Processo de Aquisição da Imagem	53
3.2.2 Binarização da Imagem	54
3.2.3 Procedimento de Melhoria de Imagem	55
3.2.4 Procedimentos Específicos dos Equipamentos Industriais	57
3.2.5 Reconhecimento de Padrões dos Caracteres	60
CONSIDERAÇÕES FINAIS	62
REFERÊNCIAS	63

INTRODUÇÃO

A automação industrial tem como objetivo criar mecanismos, como softwares ou equipamentos específicos, que serão aplicados em uma determinada máquina ou processo industrial, visando à melhora da produtividade, diminuição dos custos e a qualidade de processos tidos como repetitivos.

Nesse contexto, atualmente é possível observar o aumento do uso de componentes de hardware e sistemas complexos de Visão Computacional em diversos setores industriais. A visão computacional tenta detectar os problemas e minimizar os erros gerados no processo de produção, aumentando a qualidade dos produtos finais. Hoje em dia, o uso de inspeção óptica usando diferentes tipos de câmera, como infravermelho, câmeras 3D, dentre outras, depende dos requisitos, tais como detectar efeitos de superfície. E algoritmos complexos são usados em setores nas indústrias para diferentes propósitos, tais como detecção de defeitos, reconhecimento e classificação.

Este projeto tem como objetivo geral desenvolver um sistema capaz de fazer a leitura de um mostrador digital de um equipamento industrial, utilizando como entrada para o sistema, uma imagem obtida a partir de uma câmera. Este procedimento visa o reconhecimento de padrões de caracteres. A partir do desenvolvimento e implementação desse projeto, a leitura dos dispositivos que antes era realizada por pessoas, será realizada por um sistema automatizado, proporcionando uma manipulação dos dados de forma direta entre o computador e o equipamento.

No capítulo I deste trabalho será abordada a fundamentação teórica coletada visando à aprendizagem de métodos, técnicas e processos para obtenção de cada um dos resultados. No capítulo II será apresentada a metodologia de desenvolvimento do projeto, desde a análise da arquitetura geral até as ferramentas utilizadas como base para o desenvolvimento do sistema. No capítulo III será explicado o processo de desenvolvimento e implementação do sistema, detalhando as técnicas de Computação Visual utilizadas, os procedimentos e os artifícios desenvolvidos para solucionar os eventuais detalhes do projeto; ao final do capítulo serão apresentados os resultados obtidos.

CAPÍTULO I – REFERENCIAL TEÓRICO PARA A RESOLUÇÃO DO PROBLEMA

Neste capítulo serão abordados os assuntos utilizados como base para o desenvolvimento do projeto. Conceitos básicos de processamento digital de imagens são pré-requisitos para obtenção da arquitetura do sistema. Métodos e técnicas avançadas da área de processamento digital de imagens também serão apresentados. As etapas fundamentais no processamento digital de imagens serão citadas, no entanto apenas as etapas que foram utilizadas na elaboração do projeto serão detalhadas.

1.1 PROCESSAMENTO DIGITAL DE IMAGENS

Segundo Gonzalez (2002), uma imagem pode ser definida como uma função bidimensional, $f(x, y)$, onde x e y são coordenadas espaciais (planas), e a amplitude de f em qualquer par de coordenadas (x, y) é chamada intensidade ou nível de cinza da imagem nesse ponto. Quando x , y , e os valores de amplitude de f são todos finitos, quantidades discretas, tem-se uma imagem digital.

O campo de processamento de imagem digital refere-se ao processamento de imagens digitais por meio de um computador digital. Uma imagem digital é composta por um número finito de elementos, cada um dos quais tem uma localização particular e valor. Estes elementos são referidos como elementos de imagem. Pixel é o termo mais utilizado para denotar os elementos de uma imagem digital (GONZALEZ, 2002).

No campo de processamento de imagem digital é possível considerar três tipos de processos informatizados: processos de baixo, médio e alto nível. Processos de baixo nível envolvem operações primitivas, como o pré-processamento de imagens para reduzir o ruído, o aumento do contraste e a nitidez da imagem. Um processo de baixo nível é caracterizado pelo fato de que as entradas e as saídas são imagens. O processamento de nível médio em imagens envolve tarefas como a segmentação, a descrição desses objetos para reduzi-los a uma forma adequada para processamento computacional e a classificação de objetos individuais. Um processo de nível médio é caracterizado geralmente pelo fato de que suas saídas são atributos extraídos dessas imagens (por exemplo, arestas, contornos e a identidade dos objetos individuais). Finalmente, o

processamento de alto nível envolve um conjunto de objetos reconhecidos, como na análise de imagens (GONZALEZ, 2002).

1.1.1 Exemplos de Aplicações de Processamento de Imagens

A partir do surgimento dos primeiros computadores digitais e o início do programa espacial, conhecida também como corrida espacial, foram utilizadas técnicas computacionais de aprimoramento de imagens, quando imagens transmitidas por uma sonda espacial, eram processadas por computador para corrigir vários tipos de distorção inerentes à câmera acoplada à sonda. Estas técnicas serviram de base para métodos aprimorados de realce e restauração de imagens de outros programas espaciais posteriores (MARQUES, 1999).

Nos dias atuais, a área de processamento de imagens apresenta uma expressiva quantidade de aplicações que estão implementadas em vários setores da sociedade. O uso de imagens no diagnóstico médico pode ser citado como um exemplo bastante utilizado nos dias atuais, detecção de diversos tipos de câncer, como pode ser visto na Figura 1, prevenção de lesões físicas de jogadores de futebol, entre outras aplicações. A capacidade de processar automaticamente imagens obtidas de microscópios, por exemplo, contando o número de células de certo tipo presentes em uma imagem. O processamento e a interpretação automática de imagens captadas por satélites auxiliam os trabalhos nas áreas de Geografia (MARQUES, 1999).

Na área de Meteorologia, por exemplo, a previsão do clima teve um expressivo avanço referente à precisão das informações. Robôs com visão artificial em tarefas tais como controle de qualidade em linhas de produção aumenta a cada ano, num cenário de crescente automação industrial (MARQUES, 1999).

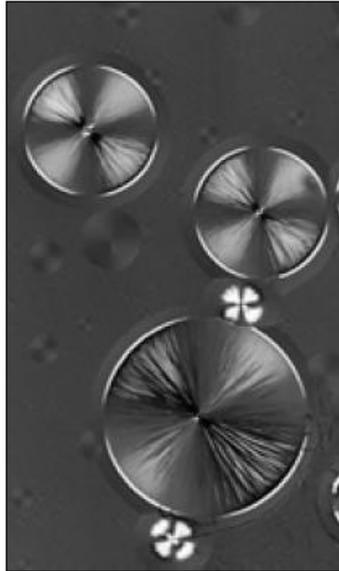


Figura 1 - Taxol - Agente anticâncer (fonte: GONZALEZ, 2002)

1.1.2 Componentes de um Sistema de Processamento de Imagens

Segundo Jahne (2005) um sistema de processamento e aquisição de imagens de propósito geral consiste de quatro elementos essenciais:

1. Um sistema de aquisição de imagens. Pode ser uma câmera *CCD*, um *scanner* ou um gravador de vídeo, como pode ser visto na Figura 2.
2. Um dispositivo conhecido como *frame grabber* para converter o sinal elétrico (normalmente um sinal de vídeo analógico) do sistema de aquisição de imagens para uma imagem digital que pode ser armazenada.
3. Um computador pessoal ou uma estação de trabalho que forneça poder de processamento.
4. Um *software* de processamento de imagens que ofereça as ferramentas para manipular e analisar as imagens.

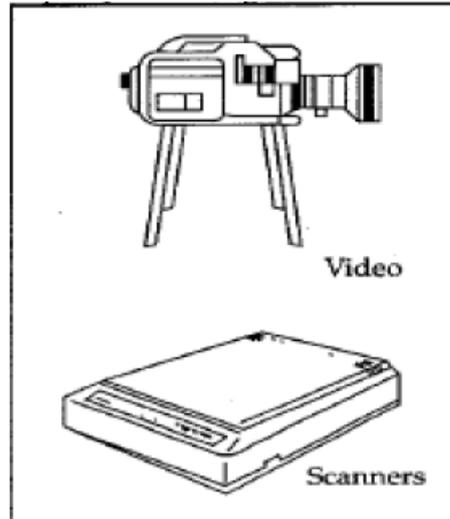


Figura 2 - Aquisição de Imagens (Fonte: Gonzalez, 2002)

1.1.3 Etapas fundamentais no Processamento Digital de Imagens

Como podem ser visualizadas na Figura 3, as etapas fundamentais no Processamento Digital de Imagens podem ser organizadas da seguinte forma (GONZALEZ, 2002):

1. Aquisição de Imagens. No tópico 1.1.2 foram apresentados os componentes que tornam possível a execução desta etapa. Um dispositivo físico sensível a uma banda do espectro de energia eletromagnética produzirá um sinal elétrico, como saída proporcional ao nível de energia percebida, em seguida um digitalizador realizará a conversão da saída elétrica do dispositivo físico para a forma digital.
2. Melhoria de Imagem. Basicamente, as técnicas de aprimoramento de imagem têm como objetivo destacar determinadas características de interesse de uma imagem.
3. Restauração da imagem. As técnicas de restauração de imagem são baseadas em modelos matemáticos ou probabilísticos de degradação da imagem.
4. Processamento de imagem colorida. Esta área abrange conceitos fundamentais em modelos de cores e processamento básico de cores em um

domínio digital, a cor pode ser usada como a base para extração de características de interesse de uma imagem.

5. *Wavelets* e processamento multi-resolução. *Wavelets* se apresenta como uma base para a representação de imagens em vários graus de resolução.
6. Compressão. A compressão refere-se a técnicas para reduzir o armazenamento necessário para salvar uma imagem, ou a largura de banda necessária para transmitir a imagem.
7. Processamento Morfológico. O processamento morfológico trata de ferramentas para extrair componentes de imagem que são úteis na representação e descrição da forma.
8. Segmentação. Os procedimentos de segmentação dividem uma imagem em seus componentes ou objetos. A segmentação é uma das etapas mais complexas no processamento de imagens digital. Quanto melhor for o algoritmo de segmentação, maior a chance de sucesso no reconhecimento da imagem.
9. Representação e Descrição. A escolha de uma representação é apenas parte da solução para transformar dados brutos em uma forma adequada para o processamento subsequente do computador. Um método deve ser especificado para descrever os dados de modo que os recursos de interesse sejam destacados.
10. Reconhecimento de Objetos. O reconhecimento é o processo que atribui um rótulo a um objeto com base nos seus descritores.

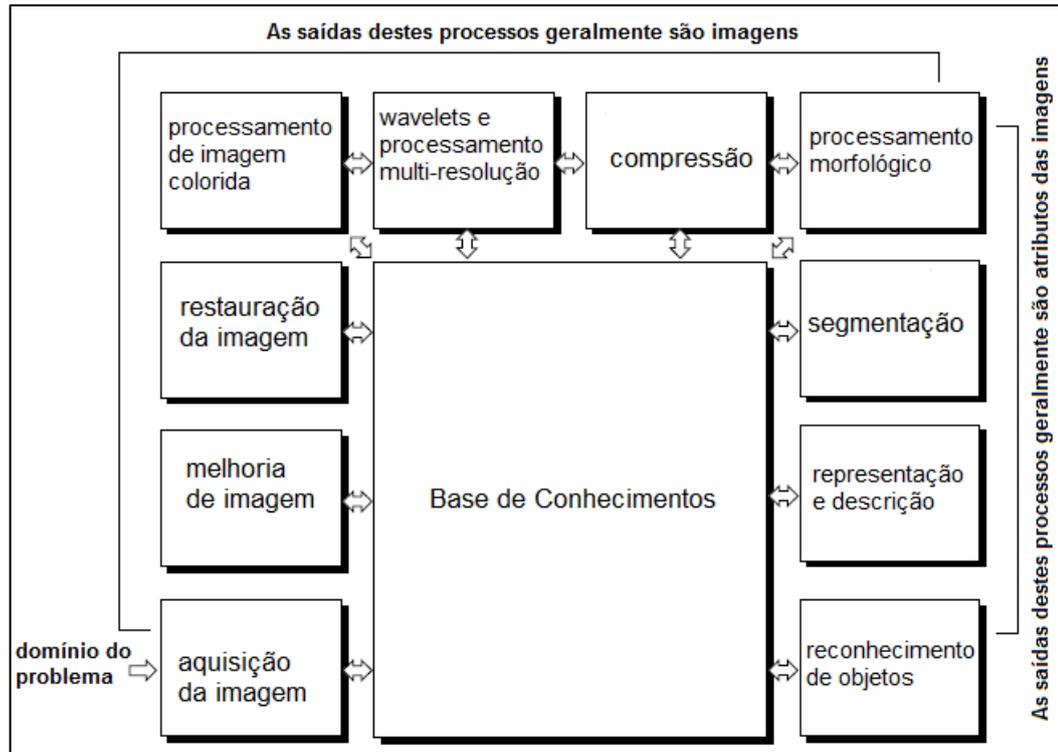


Figura 3 - Etapas fundamentais no processamento digital de imagens (Fonte: GONZALEZ, 2002)

A visualização dos resultados do processamento de imagem pode ocorrer na saída de qualquer etapa apresentada na Figura 3. É importante ressaltar que nem todas as aplicações de processamento de imagem requerem a complexidade das interações apresentadas na Figura 3.

1.2 AMOSTRAGEM E QUANTIZAÇÃO DA IMAGEM

Para criar uma imagem digital é preciso converter os dados sensíveis contínuos em formato digital e isso envolve dois processos que são chamados de Amostragem e Quantização (GONZALEZ, 2002).

Uma imagem pode ser contínua em relação às coordenadas x e y , bem como em amplitude. Para convertê-la em forma digital, tem-se que amostrar a função em ambas as coordenadas e amplitude. A digitalização dos valores das coordenadas é chamada de amostragem. A digitalização dos valores das amplitudes é chamada de quantização.

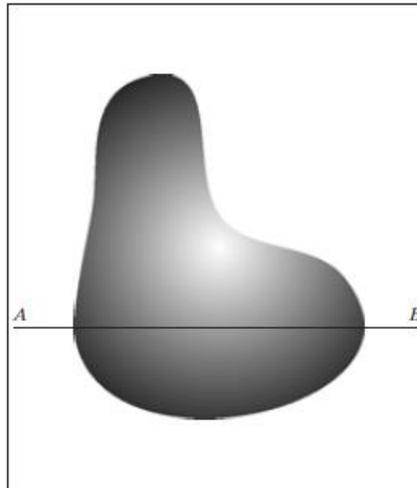


Figura 4 - Processo de Amostragem 1º parte (Fonte: GONZALEZ, 2002)

O processo de amostragem tem como ponto de partida um sinal analógico e terá como resultado uma sequência de amostras discretas. Na Figura 4 é possível perceber uma linha que vai do ponto A para o ponto B.

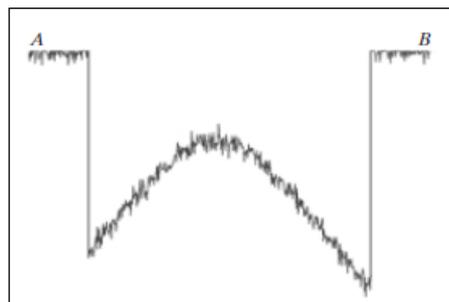


Figura 5 - Processo de Amostragem 2º parte (Fonte: GONZALEZ, 2002)

A partir da linha AB representada na Figura 5, retirada da imagem contínua da Figura 4, pode-se analisar o processo de amostragem e quantização. Em seguida, tem-se de fato o processo referente à amostragem e quantização (Figura 6), neste processo é possível visualizar o resultado como uma sequência de amostras discretas. Um sinal contínuo original pode ser aproximado por um sinal construído de amplitudes discretas selecionadas de um conjunto disponível, baseadas no mínimo erro. A quantização pode ser definida como a aproximação do valor obtido pelo processo de amostragem, para o nível mais próximo dentre um número finito de níveis de quantização. Na Figura 6 é possível visualizar o processo de quantização realizado na linha AB.

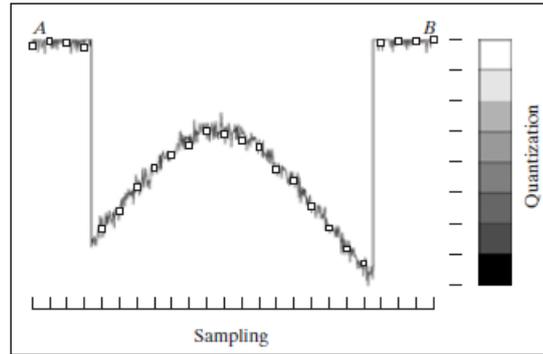


Figura 6 - Processo de Quantização (Fonte: GONZALEZ, 2002)

1.2.1 Representação de Imagens Digitais

O resultado do processo de amostragem e quantização é uma matriz de números reais (GONZALEZ, 2002). Assumindo que uma imagem $f(x, y)$ é amostrada, tem-se que o resultado da imagem digital possui M linhas e N colunas. Os valores das coordenadas na origem são $(x, y) = (0, 0)$. Os próximos valores das coordenadas ao longo da primeira linha da imagem são representadas como $(x, y) = (0, 1)$.

$$f(x, y) = \begin{matrix} f(0, 0) & f(0, 1) & \dots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & \dots & f(1, N - 1) \\ \vdots & \vdots & & \vdots \\ f(M - 1, 0) & f(M - 1, 1) & \dots & f(M - 1, N - 1) \end{matrix} \quad \text{Eq.(1)}$$

O lado direito da equação (1) é por definição uma imagem digital. Cada elemento desta matriz é chamado de elemento de imagem ou *pixel* (GONZALEZ, 2002).

Seja Z e R o conjunto de números inteiros reais e o conjunto de números reais, respectivamente. O processo de amostragem pode ser visto como um particionamento do plano xy em uma grade, com as coordenadas do centro de cada grade sendo um par de elementos do produto cartesiano Z^2 , que é o conjunto de todos os pares ordenados de elementos (z_i, z_j) , com z_i e z_j sendo inteiros de Z . Portanto, $f(x, y)$ é uma imagem digital se (x, y) são inteiros de Z^2 e f é uma função que atribui um valor de nível de cinza (ou seja, um número real do conjunto de números reais, R) para cada par de coordenadas distintas (x, y) . Esta atribuição funcional é o processo de quantização descrito na seção 1.2. Se os níveis de cinza

também são números inteiros, Z substitui R e uma imagem digital torna-se uma função 2-D cujas coordenadas e valores de amplitude são inteiros (GONZALEZ, 2002).

1.3 MELHORIA DE IMAGEM

O principal objetivo da melhoria de imagem é processar uma imagem com o objetivo de torná-la mais adequada que a imagem original para uma aplicação específica (GONZALEZ, 2002).

A partir do processo de amostragem e quantização é produzida uma imagem resultante que pode apresentar características indesejadas para determinada aplicação. Por exemplo, o contraste da imagem, a presença de ruídos, entre outras características podem fazer com que haja a necessidade de um aprimoramento na qualidade da imagem. As técnicas de aprimoramento utilizadas para gerar uma imagem digitalizada de melhor qualidade são denominadas de baixo nível, porque trabalham diretamente com os valores de intensidade dos pixels (MARQUES, 1999).

1.3.1 Filtragem Mediana

A filtragem mediana é um processo não linear útil na redução do ruído impulsivo ou *salt-and-pepper* (sal e pimenta). Também é útil na preservação das bordas em uma imagem, ao mesmo tempo em que reduz o ruído aleatório. Na filtragem mediana, uma janela desliza ao longo da imagem, e o valor de intensidade mediano dos pixels dentro da janela substitui a intensidade do pixel que está sendo processado. Por exemplo, quando os valores dos pixels dentro de uma janela são 5, 6, 35, 10, e 15, e o pixel que está sendo processado tem o valor de 35, este valor é substituído por 10, que é a mediana dos 5 valores (EKSTROM, 1984).

A filtragem mediana suaviza a imagem e, portanto, é útil para reduzir o ruído. O filtro mediano pode preservar descontinuidades em uma função degrau e pode suavizar alguns pixels cujos valores diferem de seus arredores sem afetar os outros pixels.

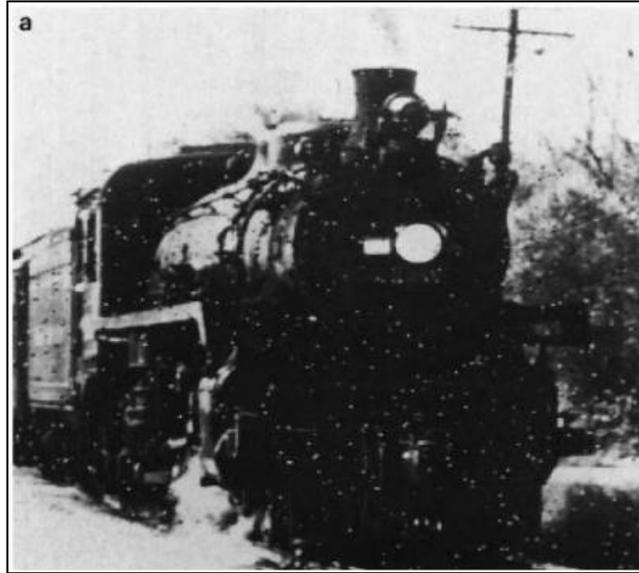


Figura 7 - Salt-and-pepper noise (fonte: EKSTROM, 1984)

A Figura 7 apresenta uma imagem degradada com a presença do ruído *salt-and-pepper*. Aplicando a filtragem mediana com uma janela 3 X 3, tem-se como resultado a imagem da Figura 8.

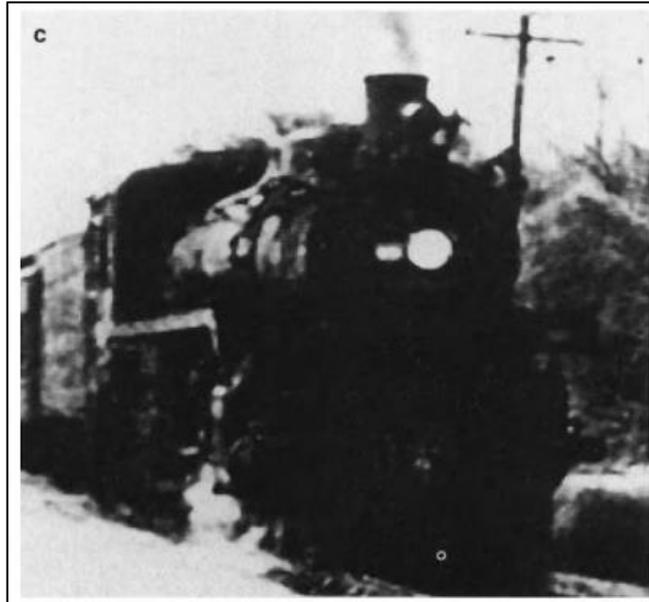


Figura 8 - Resultado da filtragem mediana (fonte: EKSTROM, 1984)

1.3.2 Filtragem Gaussiana

Filtro gaussiano é uma classe de filtros de suavização lineares com os pesos escolhidos de acordo com a forma de uma função de Gauss (JAIN, 1995). O filtro de

suavização gaussiano é um bom filtro para remoção de ruído elaborado a partir de uma distribuição normal. A função de Gauss de média nula em uma dimensão é:

$$g(x) = e^{-\frac{x^2}{2\sigma^2}} \quad \text{Eq.(2)}$$

Onde o desvio padrão de Gauss σ determina a largura da gaussiana. Para o processamento de imagem, a função bidimensional gaussiana discreta de média nula é (JAIN, 1995):

$$g[i, j] = e^{-\frac{(i^2+j^2)}{2\sigma^2}} \quad \text{Eq.(3)}$$

Esta função é usada como um filtro de suavização. Um gráfico da função bidimensional gaussiana de média zero é mostrado na Figura 9.

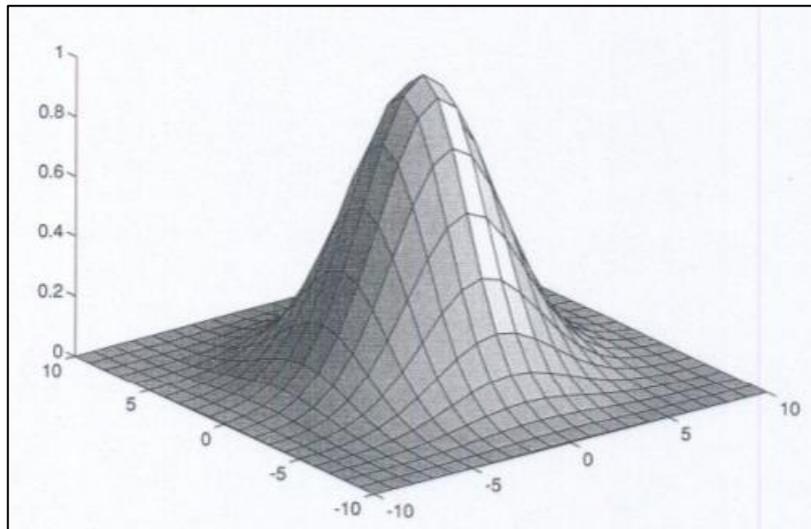


Figura 9 - Função gaussiana (fonte: JAIN, 1995)

A aplicação da filtragem gaussiana é efetiva quando se refere à suavização de imagens. Na Figura 10, a filtragem gaussiana foi aplicada na imagem do lado esquerdo e o resultado é a imagem suavizada do lado direito.



Figura 10 - Aplicação da filtragem gaussiana (fonte: JAIN, 1995)

1.3.3 Melhoria de Imagem usando operações Lógicas e Aritméticas

As operações lógicas e aritméticas envolvendo imagens são realizadas baseando-se pixel-a-pixel, entre duas ou mais imagens (isto exclui a operação lógica *NOT*, o que é realizado em uma única imagem). Como um exemplo, a subtração de dois resultados de imagens (GONZALEZ, 2002).

Ao lidar com operações lógicas em imagens em escala de cinza, os valores de pixel são processados como sequências de números binários. Por exemplo, realizando a operação *NOT* sobre um pixel de oito bits com a intensidade que representa a cor preta, produz um pixel com a intensidade que representa a cor branca. Os valores intermediários são processados da mesma maneira, mudando todos os bits 1 para bits 0 e vice-versa.

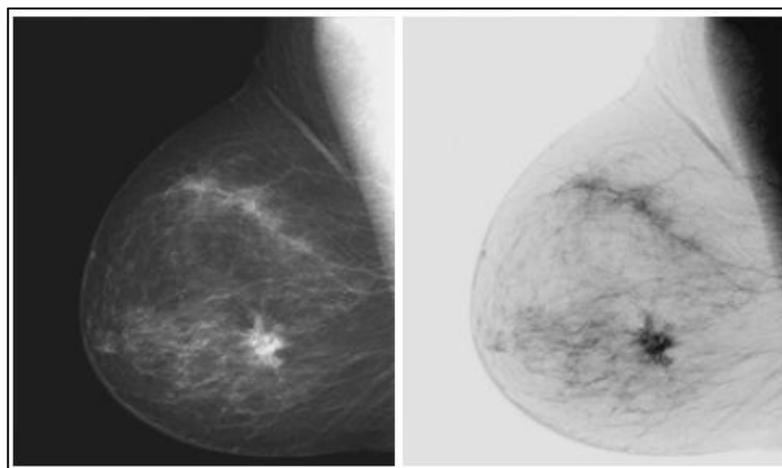


Figura 11 - Operação (Fonte: GONZALEZ, 2002)

Na Figura 11 é possível visualizar a operação lógica *NOT* aplicada a uma imagem gerada a partir de um exame de mamografia.

1.4 RESTAURAÇÃO DE IMAGEM

A restauração é um processo que tenta reconstruir ou recuperar uma imagem que foi degradada, usando o conhecimento prévio do fenômeno da degradação. Semelhante à técnica de melhoria de imagem, a função da técnica de restauração é melhorar a imagem. As técnicas de restauração baseiam-se na modelagem da degradação utilizando conhecimento prévio e na aplicação do processo inverso para restaurar a imagem original (ANNADURAI, 2006).

1.4.1 Transformações Geométricas

As técnicas de restauração podem ser aplicadas as imagens que já sofreram distorções geométricas com o intuito de modificar as relações espaciais entre pixels e restaurar a imagem (ANNADURAI, 2006).

As duas operações básicas em uma transformação geométrica são:

- Uma transformação espacial que descreve os procedimentos de rearranjo para pixels no plano da imagem.
- A interpolação de nível de cinza que trata das atribuições de níveis de cinza a pixels na imagem transformada espacialmente.

Suponhamos que uma imagem f com as coordenadas de *pixels* (x, y) sofre uma distorção geométrica para produzir uma imagem g com as coordenadas (x', y') . Esta expressão pode ser expressa como:

$$x' = r(x, y) \quad \text{Eq.(4)}$$

e

$$y' = s(x, y) \quad \text{Eq.(5)}$$

onde $r(x, y)$ e $s(x, y)$ são transformações espaciais que produzem a imagem distorcida geometricamente $g(x', y')$. Por exemplo, se $r(x, y) = x/2$ e $s(x, y) = y/2$, a

“distorção” é simplesmente o encolhimento do tamanho de $f(x, y)$ pela metade em ambas as direções espaciais. Um método frequentemente usado para formular a realocação espacial dos pixels é o uso de *tiepoints* (pontos de ligação), que são um subconjunto de pixels cuja localidade na entrada (imagem distorcida) e na saída (imagem corrigida) é conhecida precisamente (ANNADURAI, 2006).

A Figura 12 mostra regiões quadrilaterais numa imagem distorcida e na correspondente imagem corrigida. Os vértices dos quadriláteros são *tiepoints* (pontos de ligação) correspondentes.

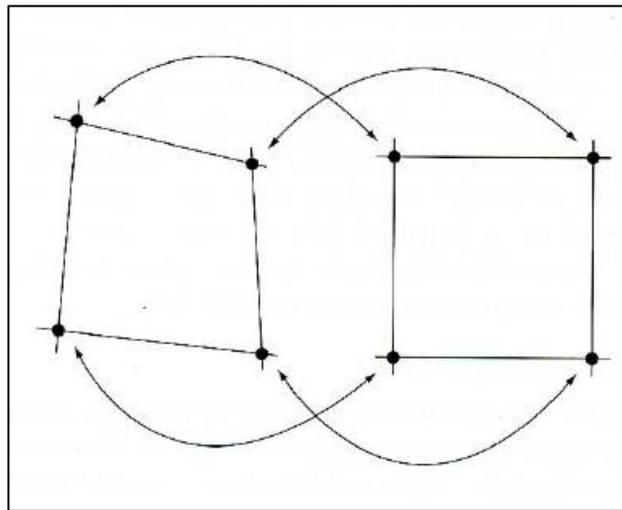


Figura 12 – Tiepoints (Fonte: GONZALEZ, 2002)

Suponhamos que o processo de distorção geométrica dentro das regiões quadrilaterais é modelado por um par de equações de modo que:

$$r(x, y) = c_1x + c_2y + c_3xy + c_4 \quad \text{Eq.(6)}$$

e

$$s(x, y) = c_5x + c_6y + c_7xy + c_8 \quad \text{Eq.(7)}$$

Portanto, a partir das equações (4) e (5),

$$x' = c_1x + c_2y + c_3xy + c_4 \quad \text{Eq.(8)}$$

e

$$y' = c_5x + c_6y + c_7xy + c_8 \quad \text{Eq.(9)}$$

Desde que existam um total de oito *tiepoints* (pontos de ligações) conhecidos, estas equações podem ser resolvidas para os oito coeficientes c_i , $i = 1, 2, \dots, 8$. Os coeficientes constituem o modelo de distorção geométrica utilizado para transformar todos os pixels dentro da região quadrilateral definida pelos *tiepoints* (pontos de ligação) utilizados para obter os coeficientes. Em geral, bastantes *tiepoints* (pontos de ligação) são necessários para gerar um conjunto de quadriláteros que cobrem toda a imagem, com cada quadrilátero tendo seu próprio conjunto de coeficientes (ANNADURAI, 2006).

Uma vez que se têm os coeficientes, o procedimento usado para gerar a imagem restaurada não é difícil. Se nós quisermos encontrar os valores da imagem não distorcida em qualquer ponto (x_0, y_0) , nós simplesmente precisamos conhecer onde na imagem distorcida $f(x_0, y_0)$ foi mapeado. Isto nós descobrimos pela substituição de (x_0, y_0) nas equações (8) e (9) para obter as coordenadas (x'_0, y'_0) . O valor do ponto na imagem não distorcida que foi mapeada para (x'_0, y'_0) é $g(x'_0, y'_0)$. Portanto nós obtemos o valor da imagem restaurada simplesmente deixando $\hat{f}(x_0, y_0) = g(x'_0, y'_0)$.

1.5 PROCESSAMENTO MORFOLÓGICO DE IMAGEM

Pode-se denominar o processo morfológico de imagem como uma ferramenta para extrair componentes de imagem que são úteis na representação e descrição da forma da região (GONZALEZ, 2002). O processamento morfológico de imagem pode ser aplicado como técnica que visa atingir características de realce, filtragem, segmentação, detecção de bordas, afinamento, entre outras (MARQUES, 1999).

Para um melhor entendimento sobre as técnicas utilizadas no processo morfológico de imagem será útil obter informações sobre os conceitos do processo de convolução numa imagem digital. A ideia desta operação é utilizar uma máscara com dimensão d , onde cada posição da máscara possui um determinado valor. Em seguida coloca-se a máscara com sua posição central sobre a imagem na posição (x, y) e, para cada posição da máscara executa-se o produto do valor da máscara pelo valor do *pixel*, faz-se a somatória destes valores obtidos na multiplicação e substitui-se o valor da posição (x, y) por este resultado (MARENGONI, 2009).

1.5.1 Dilatação

Com A e B (Figura 13 e Figura 14) como conjuntos em Z^2 (ver seção 1.2.1), a dilatação de A por B é então o conjunto de todos os deslocamentos, z (ver seção 1.2.1), tais que B e A se sobrepõem por pelo menos um elemento. O conjunto B comumente é definido como o elemento estruturante na dilatação, bem como nas outras operações morfológicas.

Embora a dilatação seja baseada em operações de conjuntos, enquanto a convolução é baseada em operações aritméticas, o processo básico de "lançar" B sobre sua origem e depois deslocá-la sucessivamente de modo que ele deslize sobre o conjunto (imagem) A , é análogo ao processo de convolução (GONZALEZ, 2002).

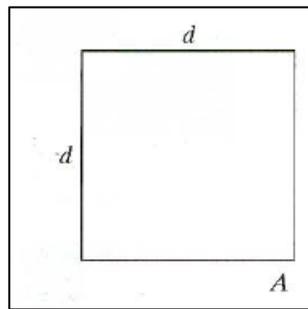


Figura 13 - Conjunto A (Fonte: GONZALEZ, 2002)

Na figura 13 pode ser visualizado o conjunto A .

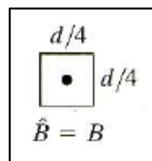


Figura 14 - Conjunto B (Fonte: GONZALEZ, 2002)

Na Figura 14 pode ser visualizado o conjunto B , que se refere ao elemento estruturante e sua reflexão (o ponto preto denota a origem do elemento). Neste caso o elemento estruturante e sua reflexão são iguais porque B é simétrico com relação à origem. O conceito da operação de reflexão é: Reflete todos os elementos de B sobre a origem deste conjunto.

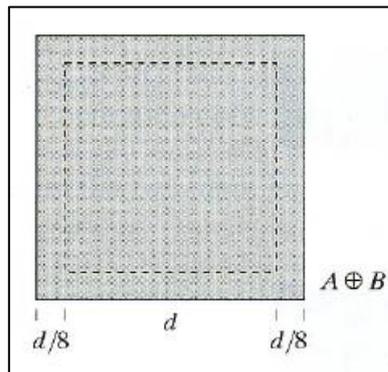


Figura 15 - Dilatação de A por B (Fonte: GONZALEZ, 2002)

Na Figura 15 pode ser visualizada a dilatação de A por B . A linha pontilhada mostra o conjunto original para referência, e a linha sólida mostra o limite para além do qual quaisquer deslocamentos adicionais da origem da reflexão de B por z fariam com que a intersecção da reflexão de B e do conjunto A ficasse vazia. Portanto, todos os pontos dentro desta fronteira constituem a dilatação de A por B .

1.5.2 Erosão

Para os conjuntos A e B em \mathbb{Z}^2 , a erosão de A por B é o conjunto de todos os pontos z (ver seção 1.2.1), tais que B , transladado por z , está contido em A . Utilizando o mesmo conjunto A , apresentado na Figura 13, e o mesmo conjunto B , apresentado na Figura 14. Na Figura 16 pode ser visualizado o processo que é similar ao apresentado na Figura 15. Como antes, o conjunto A é mostrado com uma linha pontilhada para referência na Figura 16. O conceito da operação de translação é: Translada a origem do conjunto A , para o ponto z (GONZALEZ, 2002).

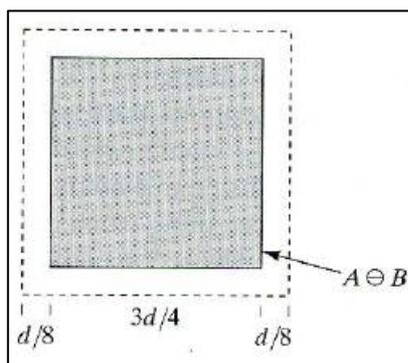


Figura 16 - Operação de Translação (Fonte: GONZALEZ, 2002)

A fronteira da região sombreada mostra o limite além do qual outro deslocamento da origem de B faria com que este conjunto não estivesse mais completamente contido em A . Portanto, a região sombreada dentro da fronteira constitui a erosão de A por B .

1.5.3 Abertura e Fechamento

Como foi mostrada anteriormente, a dilatação expande e a erosão encolhe uma imagem. A operação de abertura geralmente suaviza o contorno de um objeto, rompe istmos estreitos e elimina protruções finas. A operação de fechamento tende a suavizar seções de contornos, mas ao contrário de abertura, geralmente, ele elimina golfos longos e finos, elimina pequenos furos, e preenche lacunas no contorno (GONZALEZ, 2002).

Utilizando o mesmo conjunto A , apresentado na Figura 13, e o mesmo conjunto B , apresentado na Figura 14. Tem-se que a abertura do conjunto A , pelo elemento estruturante B é a operação de erosão de A por B , seguida pela dilatação do resultado por B . Similarmente, a operação de fechamento do conjunto A , pelo elemento estruturante B é simplesmente a dilatação de A por B , seguida da erosão do resultado por B .

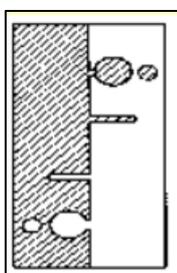


Figura 17 - Imagem Fonte (Fonte: LOPES et al., 2017)

Na Figura 17 é apresentada uma imagem a qual será aplicada a operação de abertura.

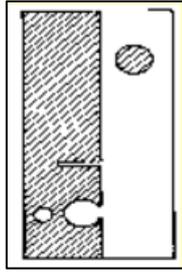


Figura 18 - Operação de Abertura (Fonte: LOPES et al., 2017)

Na figura 18 é apresentada a imagem resultante da operação de abertura, aplicada sobre a imagem da Figura 17. É possível perceber há quebra de istmos e eliminação de protusões finas.

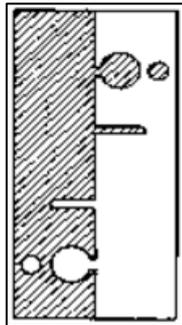


Figura 19 - Imagem Fonte (Fonte: LOPES et al., 2017)

Na Figura 19 é apresentada uma imagem a qual será aplicada a operação de fechamento.

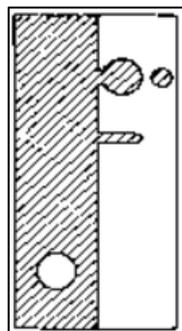


Figura 20 - Operação de Fechamento (Fonte: LOPES et al., 2017)

Na figura 20 é apresentada a imagem resultante da operação de fechamento, aplicada sobre a imagem da Figura 19. É possível perceber que há a eliminação de golfos e preenchimento de lacunas no contorno.

1.6 SEGMENTAÇÃO DE IMAGEM

A Segmentação subdivide uma imagem em suas regiões ou objetos constituintes. A Segmentação deve parar quando os objetos de interesse em um aplicativo forem isolados. Os algoritmos de segmentação de imagens geralmente são baseados em uma das duas propriedades básicas de valores de intensidade: descontinuidade e similaridade.

Na primeira categoria, a abordagem é dividir uma imagem com base em mudanças abruptas de intensidade, como bordas em uma imagem. As abordagens principais na segunda categoria são baseadas na partição de uma imagem em regiões que são semelhantes de acordo com um conjunto de critérios predefinidos. O limiar, o crescimento da região e a divisão e fusão de regiões são exemplos de métodos nessa categoria (GONZALEZ, 2002).

1.6.1 Detecção de Descontinuidade

A maneira mais comum de procurar descontinuidade é executar uma máscara através da imagem da maneira descrita na seção 1.5.

Para uma máscara 3 X 3, como pode ser vista na Figura 21, este procedimento envolve a computação da soma dos produtos dos coeficientes com os níveis de cinza contidos na região abrangida pela máscara (GONZALEZ, 2002).

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Figura 21 - Máscara 3x3 (Fonte: GONZALEZ, 2002)

A resposta da máscara em qualquer ponto na imagem é dada por:

$$R = w_1z_1 + w_2z_2 + \dots + w_9z_9 \quad \text{Eq.(10)}$$

Onde z_i é o nível de cinza do *pixel* associado com o coeficiente da máscara w_i . Como de costume, a resposta da máscara é definida em relação à sua localização central.

1.6.2 Detecção de Ponto

A detecção de pontos isolados numa imagem é um princípio direto. Nós dizemos que um ponto foi detectado no local em que a máscara é centralizada se:

$$|R| \geq T \quad \text{Eq.(11)}$$

Onde T é um limiar não negativo; e R é dado pela equação (10). Basicamente, esta formulação mede a diferença do peso entre o ponto central e seus vizinhos. A ideia é que um ponto isolado (um ponto cujo nível de cinza é significativamente diferente do seu fundo e que está localizado numa área totalmente ou parcialmente homogênea) será bastante diferente do seu ambiente, e assim ser facilmente detectável por este tipo de máscara (GONZALEZ, 2002).

1.6.3 Detecção de Linha

-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
Horizontal			+45°			Vertical			-45°		

Figura 22 - Máscaras 3x3 (Fonte: GONZALEZ, 2002)

Considerando as máscaras mostradas na Figura 22, caso a primeira máscara fosse movida em torno de uma imagem, ela responderia mais fortemente a linhas (um pixel de espessura) orientadas horizontalmente. Com um fundo constante, a resposta máxima teria resultado quando a linha passasse através da linha do meio

da máscara. A segunda máscara na Figura 22 responde melhor a linhas orientadas a $+45^\circ$; a terceira máscara a linhas verticais; e a quarta máscara para linhas na direção -45° . Vale ressaltar que os coeficientes em cada máscara somam zero, indicando uma resposta zero das máscaras em áreas de nível de cinza constante (GONZALEZ, 2002).

1.6.4 Detecção de Borda

A detecção de bordas é a abordagem mais comum para detectar descontinuidades significativas no nível de cinza. Uma borda é um conjunto de pixels conectados que se encontram na fronteira entre duas regiões. As bordas são regiões da imagem onde ocorre uma mudança de intensidade em certo intervalo do espaço, em determinada direção. Isto corresponde a regiões de alta derivada espacial, que contêm alta frequência espacial (GONZALEZ, 2002).

Um operador que é sensível às variações abruptas de intensidade operará como um detector de bordas. Um operador de derivada faz exatamente esta função. Uma interpretação de derivada seria a taxa de mudança de uma função – a taxa de mudança dos níveis de cinza em uma imagem é maior perto das bordas e menor em áreas constantes. Ao obter os valores de intensidade da imagem e calcular os pontos onde a derivada é um ponto de máximo, tem-se a marcação das bordas (QUEIROZ, 2001).

Dado que as imagens são representadas em duas dimensões, é importante considerar mudanças nos níveis de cinza em muitas direções. Por esta razão, derivadas parciais das imagens são usadas, com as respectivas direções X e Y . Uma estimativa da direção atual da borda pode ser obtida usando as derivadas X e Y como os componentes da direção ao longo dos eixos, computando-se a soma dos resultados parciais. O operador envolvido é o gradiente: se a imagem é vista como uma função de duas variáveis $f(x,y)$, então o gradiente é definido como (QUEIROZ, 2001):

$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) \quad \text{Eq.(12)}$$

A magnitude do gradiente é dada por:

$$(|\nabla f(x, y)|)^2 = \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 \quad \text{Eq.(13)}$$

Conforme a equação 13, a utilização da função magnitude do gradiente na detecção de bordas é computacionalmente muito custosa. O que se faz na prática é a aproximação da equação acima para a forma:

$$\nabla f \approx |\text{Gradiente}_x| + |\text{Gradiente}_y| \quad \text{Eq.(14)}$$

Geralmente, a magnitude do gradiente tem um valor alto na borda e baixo no interior da região, o que possibilita a determinação do contorno de uma imagem (QUEIROZ, 2001).

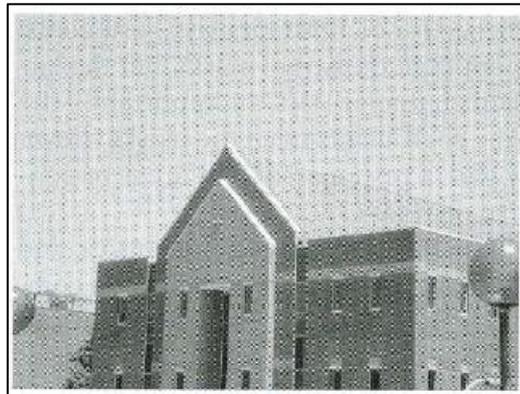


Figura 23 - Imagem Fonte (Fonte: GONZALEZ, 2002)

Na Figura 23 é mostrada uma imagem original a qual será aplicada a operação de detecção de borda.

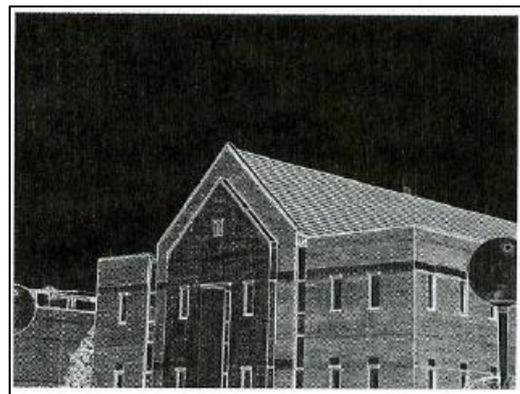


Figura 24 - Detecção de Borda (Fonte: GONZALEZ, 2002)

Na Figura 24 é mostrado o resultado da operação de detecção de borda, neste caso o gradiente da imagem, $|G_x| + |G_y|$.

1.6.5 Thresholding

Suponha que o histograma de nível de cinza mostrado na Figura 25 corresponde a uma imagem, $f(x, y)$, composta de objetos claros sobre um fundo escuro, de modo que os *pixels* do objeto e do fundo têm níveis de cinza agrupados em dois modos dominantes. Uma maneira de extrair os objetos do fundo é selecionar um *threshold* (limiar) T que separa esses modos. Então qualquer ponto (x, y) para qual $f(x, y) > T$ é chamado de ponto do objeto; de outra forma, o ponto é chamado de ponto de fundo (GONZALEZ, 2002).

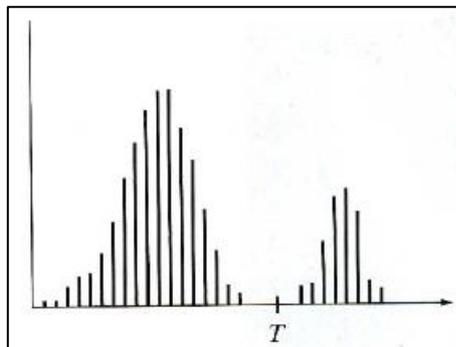


Figura 25 – Histograma (Fonte: GONZALEZ, 2002)

1.6.6 Reconhecimento de Objetos

O reconhecimento de objetos acontece quando é possível diferenciar um objeto em uma determinada imagem com clareza suficiente que sua classe possa ser discriminada, por exemplo, automóvel, pessoa, embarcação, entre outros (CRUZ, 2014).

As abordagens para o reconhecimento de padrões podem ser divididas em duas principais áreas, decisão-teórica e estrutural. A primeira categoria trata de padrões descritos usando descritores quantitativos, como comprimento, área e textura. A segunda categoria trata de padrões descritos usando descritores

qualitativos. O tema do reconhecimento é o conceito de "aprendizagem" de padrões de amostra (GONZALEZ, 2002).

As abordagens para o reconhecimento de objetos baseados em teoria da decisão incluem áreas como redes neurais, correlação, classificador de distância mínima, classificadores estatísticos. Para a abordagem do reconhecimento de objetos baseado em estrutura, têm-se as técnicas baseadas em forma (CRUZ, 2014).

1.6.6.1 Reconhecimento Óptico de Caracteres

Identificar texto em uma imagem faz parte de um processo comumente chamado de OCR (*Optical Character Recognition*) e pode ser dividido nas seguintes etapas (GODOY, 2016):

- Pré-processamento e segmentação de texto: Durante esta etapa, o computador deve aprender a lidar com o ruído e a rotação da imagem (distorção), identificar quais áreas podem ser um texto.
- Identificação de texto: Este é um processo usado para identificar cada letra em um texto.

Para a realização desta etapa, podem ser usadas ferramentas que implementam os procedimentos de reconhecimento de padrões, métodos de características/classificação e classificadores adaptativos (SZELISKI, 2010).

Tesseract é um motor OCR de código aberto que foi desenvolvido pela HP entre 1984 e 1994. O reconhecimento então procede como um processo de duas passagens. Na primeira passagem, uma tentativa é feita para reconhecer cada palavra por sua vez. Cada palavra que é satisfatória é passada para um classificador adaptativo como dados de treinamento. O classificador adaptativo então tem a chance de reconhecer com mais precisão o texto mais para baixo na página (SMITH, 2007).

Uma vez que o classificador adaptativo pode ter aprendido algo útil muito tarde para fazer uma contribuição perto do topo da página, uma segunda passagem é executada sobre a página, em que as palavras que não foram reconhecidas bem o

suficiente são reconhecidas novamente. Parte do processo de reconhecimento para qualquer mecanismo de reconhecimento de caracteres é identificar como uma palavra deve ser segmentada em caracteres (SMITH, 2007).

CAPÍTULO II – METODOLOGIA DE DESENVOLVIMENTO DO PROJETO

A metodologia de desenvolvimento para este projeto inclui a análise e projeto da arquitetura do sistema, com o intuito de definir quais os recursos mais adequados para a implementação do sistema. O objetivo geral do projeto é implementar um sistema que utilize uma câmera para fazer a leitura de um mostrador digital de um equipamento industrial, para fins de reconhecimento de padrões de caracteres. Portanto, é necessária a integração de uma câmera na arquitetura do projeto.

Para que o resultado da leitura de um mostrador digital de um equipamento industrial seja alcançado, o sistema deve fornecer ferramentas para aquisição da imagem através da câmera, bem como para o processamento digital da imagem e para o reconhecimento de padrões de caracteres. Baseado nessas informações, os procedimentos realizados incluem o estudo de materiais, livros e artigos científicos para obter embasamento teórico nas áreas de visão computacional, linguagem de programação C++ e reconhecimento óptico de caracteres.

Após os estudos e análises foram definidos quais recursos seriam utilizados para a elaboração do sistema. No contexto da visão computacional, a biblioteca OpenCV foi definida como o recurso a ser utilizada, levando-se em consideração as técnicas e funções oferecidas, o ponto de vista financeiro (software livre), a vasta documentação disponível e a integração com outras ferramentas.

Para a utilização de funções e métodos que a OpenCV não oferece, bem como a integração entre diferentes recursos, a IDE Eclipse CDT e a linguagem de programação C++ foram definidas como ferramentas para edição, desenvolvimento de código e integração entre diferentes etapas do sistema. Para o Reconhecimento Óptico de Caracteres foi definida a ferramenta Tesseract OCR desenvolvida pela HP. Tesseract OCR é uma ferramenta de código aberto e oferece diversas funções que facilitam o processo de reconhecimento de padrões de caracteres.

2.1 DEFINIÇÃO DA ARQUITETURA DO SISTEMA

A definição da arquitetura do sistema é apresentada na figura 26. A arquitetura é dividida em fases, onde cada fase implementa uma ou mais operações referentes a processamento digital de imagens.

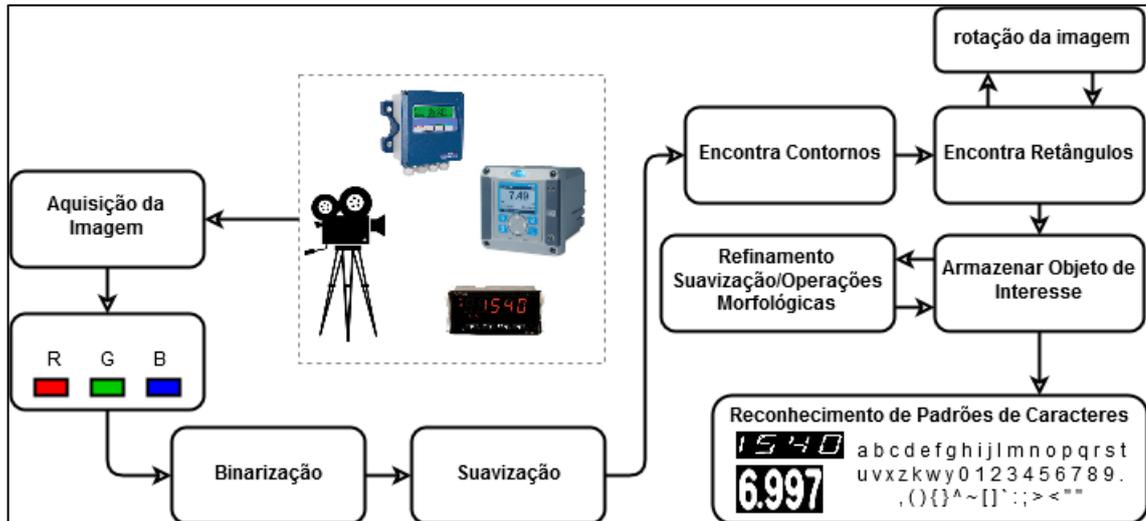


Figura 26 - Arquitetura do Sistema (Fonte: Próprio Autor)

2.2 OPENCV

OpenCV é uma biblioteca de visão computacional de fonte aberta disponível em <http://SourceForge.net/projects/opencvlibrary>. A biblioteca é escrita em C e C++, e executada em Linux, Windows e Mac OS X. Há desenvolvimento ativo em interfaces para *Python*, *Ruby*, *Matlab* e outros idiomas. A biblioteca OpenCV contém mais de 500 funções que abrangem muitas áreas de visão computacional, incluindo inspeção de produtos, imagens médicas, segurança, interface com o usuário, calibração de câmera, visão estéreo e robótica (STROUSTRUP, 2013).

OpenCV é destinado a fornecer as ferramentas básicas necessárias para resolver problemas de visão computacional. Em alguns casos, as funcionalidades de alto nível na biblioteca serão suficientes para resolver os problemas mais complexos na visão computacional. Mesmo quando este não é o caso, os componentes básicos na biblioteca são completos o suficiente para permitir a criação de uma solução completa para qualquer problema de visão de computacional (STROUSTRUP, 2013).

Entre as principais bibliotecas da OpenCV podemos incluir (STROUSTRUP, 2013):

- CXCORE

Contém estrutura de dados, álgebra de matrizes, transformações de dados, persistência de objetos, gerenciamento de memória, tratamento

de erros, carregamento dinâmico de código, bem como desenho, texto e matemática básica.

- CV
Contém processamento de imagem, análise da estrutura de imagem, movimento e rastreamento, reconhecimento de padrões, e calibração de câmera.
- *Machine Learning* (ML)
Contém muitos clusters, classificação e funções de análises de dados.
- *HighGUI*
Contém a interface de usuário GUI e o armazenamento e recuperação de imagem/vídeo.
- CVCAM
Interface da câmera.

2.3 TESSERACT OCR

Tesseract é um motor OCR de código aberto originalmente desenvolvido pela *Hewlett-Packard Laboratories, Bristol* e *Hewlett-Packard Co*. Ele tem todas as licenças de código sob a licença *Apache* e está hospedado no *GitHub* em <https://github.com/tesseract-ocr>. É considerado um dos motores de OCR mais precisos que está disponível. Ele pode ler uma grande variedade de formatos de imagem e pode converter texto escrito em mais de 60 idiomas (GODOY, 2016).

Tesseract OCR já está integrado ao OpenCV 3.0. A ferramenta fornece um módulo totalmente treinável, podendo lidar com quaisquer caracteres na codificação UTF - 8 (8 - bit *Unicode Transformation Format*).

Tesseract OCR assume que sua entrada é uma imagem binária com regiões poligonais de texto definidas. O primeiro passo do processamento é uma análise de componentes conectados, em que os contornos dos componentes são armazenados. Os contornos são reunidos como bolhas, que por sua vez são organizadas em linhas de texto. As linhas de texto são divididas em palavras de forma diferente de acordo com o tipo de espaçamento de caracteres.

O reconhecimento então procede como um processo de duas passagens. Dentro da primeira passagem, é realizada uma tentativa para reconhecer cada

palavra por vez. Cada palavra que é satisfatória é passada para um classificador adaptativo como dados de treinamento. Então, o classificador adaptativo obtém uma chance de reconhecer o texto na parte inferior da página (SMITH, 2007).

Uma vez que o classificador adaptativo não tenha reconhecido a informação desejada, uma segunda passagem é executada sobre a página, com objetivo de aplicar o procedimento novamente nas palavras que não foram reconhecidas bem o suficiente.

2.4 LINGUAGEM DE PROGRAMAÇÃO C++

C++ é uma extensão da linguagem de programação C (STROUSTRUP, 2013). A linguagem C foi desenvolvida para escrever o sistema operacional Unix, no entanto é amplamente utilizada em software em nível de sistema e desenvolvimento de sistemas embarcados. C++ fornece características da programação orientada a objetos. A linguagem C++ é muito próxima à linguagem C, pois programas escritos em C++ permitem a utilização de códigos escritos em linguagem C. A linguagem C++ é amplamente usada na indústria para desenvolvimento de software comercial.

2.5 ECLIPSE CDT

O Projeto CDT oferece um Ambiente de Desenvolvimento Integrado C e C ++ Integrado baseado na plataforma Eclipse. Alguns recursos incluem o suporte para criação de projetos e compilação, editor de código com destaque de sintaxe, ferramentas de depuração visual, incluindo memórias, registradores, entre outros.

O GNU Compiler Collection (GCC) é um conjunto de compiladores de linguagens de programação produzido pelo projeto GNU e responsável por compilar o código-fonte em binários executáveis para diversas plataformas. O MinGW é a versão do compilador GCC para Windows.

O Eclipse CDT é uma ferramenta que proporciona a integração entre o compilador MinGW, a biblioteca OpenCV e a biblioteca Tesseract.

CAPÍTULO III – DESENVOLVIMENTO DO PROJETO

Como citado anteriormente, este projeto tem como objetivo o desenvolvimento de um sistema capaz de efetuar a leitura de um mostrador digital de um equipamento industrial, utilizando como entrada para o sistema, uma imagem obtida a partir de uma câmera, visando o reconhecimento de padrões dos caracteres.

Os conceitos de processamento digital de imagens, bem como as técnicas e os recursos necessários para o desenvolvimento deste sistema foram apresentados nos capítulos anteriores. Neste capítulo serão apresentadas as soluções encontradas para o desenvolvimento do sistema. Os resultados obtidos a partir das simulações efetuadas na ferramenta Eclipse CDT também serão apresentados.

3.1 DESENVOLVIMENTO DO SISTEMA

Para o desenvolvimento do sistema apresentado na figura abaixo, foram utilizados os seguintes recursos: a biblioteca OpenCV, a biblioteca Tesseract, a ferramenta de desenvolvimento Eclipse CDT e a linguagem de programação C++. O sistema operacional utilizado para as instalações das ferramentas foi o *Windows*.

O conjunto de bibliotecas OpenCV para visão computacional pode ser obtido no site <http://www.opencv.org>. Os arquivos estão disponíveis para *download* gratuito, pois a biblioteca OpenCV é código aberto. A ferramenta referente ao ambiente de desenvolvimento para o código C/C++ é o Eclipse CDT, esta ferramenta é código aberto, portanto é possível realizar o *download* gratuito no site do projeto Eclipse. Para compilar os binários do conjunto de bibliotecas OpenCV, bem como a biblioteca Tesseract, foi necessário a instalação do compilador MinGW-w64. Para a biblioteca Tesseract OCR trabalhar com imagens é necessária a instalação das bibliotecas Leptonica, libJpeg, libPng e libTiff.

Para realizar a integração dos recursos foi necessária a configuração das variáveis de ambiente no *Windows*, bem como a configuração interna do Eclipse, adicionando as bibliotecas OpenCV e Tesseract, e o compilador MinGW-w64. Após a integração foi possível escrever códigos em C/C++ e importar as funções tanto da biblioteca OpenCV quanto da biblioteca Tesseract.

A partir da figura 27, será explicado como as funções da biblioteca OpenCV e da biblioteca Tesseract foram utilizadas para o desenvolvimento do sistema.

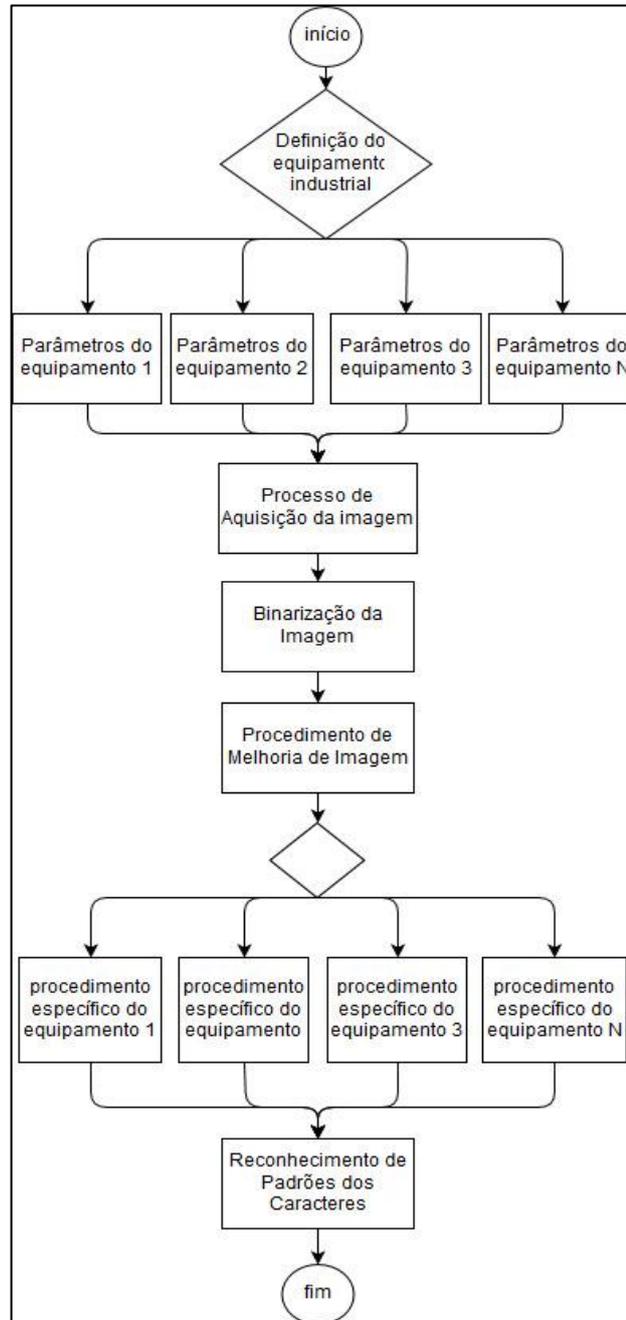


Figura 27 - Fluxograma de Processos (Fonte: Próprio Autor)

3.1.1 Definição dos Parâmetros dos Equipamentos Industriais

Como é possível observar na figura 27, o primeiro passo é a realização da escolha do equipamento industrial a ser utilizado no sistema. A necessidade dessa

escolha baseia-se no fato de que cada equipamento industrial possui uma característica que o difere dos outros equipamentos. A cor, o tamanho, a quantidade de ruído na imagem obtida a partir do equipamento industrial são características que podem variar de um equipamento para outro. Portanto, as funções do sistema foram desenvolvidas de forma que as suas entradas fossem parametrizáveis, desta forma o conceito de reuso de código pode ser aplicado em diversas partes do sistema.

Neste projeto foram utilizadas imagens obtidas a partir de dois equipamentos industriais, com características diferentes, portanto foi necessária a definição de parâmetros de entradas para cada imagem. Por exemplo, para uma imagem não foi necessária à aplicação da filtragem mediana, no entanto para a outra imagem a filtragem mediana foi fundamental para o processo de melhoria da imagem. A aplicação da filtragem gaussiana também pode ser definida, bem como o elemento estruturante e a operação morfológica a ser utilizada. No processo para a aplicação do *threshold* da imagem é necessária à definição dos limiares mínimo e máximo, bem como do algoritmo que será utilizado no processo.

A biblioteca Tesseract OCR oferece diferentes opções de dados obtidos a partir do treinamento de uma determinada linguagem. Por exemplo, o pacote de dados de treinamento “eng” que se refere ao pacote de linguagem Inglês, e o pacote de dados de treinamento “deu” que se refere ao pacote de linguagem Alemão. Ainda é possível a realização de um treinamento para uma nova linguagem, porém não faz parte do escopo deste projeto. Portanto, o pacote de dados de treinamento utilizado para cada imagem neste projeto foi diferente, sendo definidos através de parâmetros.

Como se pode perceber, uma das características do sistema desenvolvido é a flexibilidade, pois novas imagens referentes a diferentes equipamentos industriais podem ser utilizadas, no entanto a definição de parâmetros para cada imagem se faz necessária.

3.1.2 Processo de Aquisição da Imagem

Para a realização do processo de aquisição da imagem é necessário a importação da biblioteca *highgui.hpp* oriunda da OpenCV (BRADSKI, 2008). Esta

definição é realizada na classe principal `ACR_projeto_final`, onde está implementado o método `main`.

A classe `VideoCapture` tem como característica a captura de vídeo a partir arquivos de vídeos, sequência de imagens ou câmeras. A classe oferece API C++ para capturar vídeos de câmeras ou para ler arquivo de vídeo e sequência de imagens. Esta classe recebe como parâmetro de entrada um `ID` (número de identificação) referente ao dispositivo de captura de vídeo disponível. Para realizar a leitura de um `frame` (quadro) de imagem, pode-se utilizar o método `read`, representado pelo operador (`>>`), este método obtém, decodifica e retorna o próximo quadro de vídeo.

Após um quadro de vídeo ser obtido é necessário que estas informações sejam armazenadas numa estrutura para que o processamento de melhoria de imagem seja realizado posteriormente. A classe `Mat` representa uma matriz n -dimensional, que pode ser usada para armazenar vetor e matriz real ou complexa, imagem em tons de cinza ou em cores, entre outras características. Para a utilização desta estrutura de dados é necessário a importação da biblioteca `core.hpp` oriunda da OpenCV. Portanto, após a obtenção do quadro de vídeo, o mesmo é armazenado numa estrutura do tipo `Mat`.

3.1.3 Binarização da Imagem

Para realizar o processo de melhoria da imagem é indicada a utilização da imagem em tons de cinza, pois facilita a aplicação de técnicas para o tratamento da imagem. No entanto, a imagem obtida no procedimento anterior é colorida (BGR), então se faz necessário à conversão para tons de cinza.

A função `cvtColor` converte uma imagem de entrada de um espaço de cor para outro (LAGANIERE, 2011). No caso de uma transformação de/para o espaço de cor RGB, a ordem dos canais devem ser especificadas (RGB ou BGR). Esta função recebe como parâmetro de entrada o código de conversão do espaço de cor, no caso deste projeto é definido o código de conversão `CV_BGR2GRAY`. Para utilização desta função é necessário a importação da biblioteca `imgproc.hpp` oriunda da OpenCV.

Após a obtenção da imagem em tons de cinza é realizada a aplicação do processo de limiarização da imagem. A função *threshold* é capaz de realizar a limiarização de nível fixo a uma matriz de canal único (LAGANIERE, 2011). A função é normalmente usada para obter uma imagem de dois níveis (binária) ou para remover um ruído.

Como foi citado anteriormente é necessária à definição dos limiares mínimo e máximo, portanto a função recebe tais entradas como parâmetros. Outro parâmetro de entrada que pode ser definido é o tipo de limiarização, que neste projeto utiliza o `THRESH_BINARY`. Basicamente é feita uma comparação das intensidades dos pixels da imagem com o valor do limiar definido, caso a intensidade do pixel seja maior que o valor do limiar, a intensidade do pixel será substituída pela intensidade máxima que o pixel poderá ter. Caso a intensidade do pixel seja menor que o valor do limiar definido, a intensidade do pixel será substituída por zero. Para a utilização desta função é necessário a importação da biblioteca *imgproc.hpp* oriunda da OpenCV.

3.1.4 Procedimento de Melhoria de Imagem

Esta etapa do sistema tem como objetivo destacar determinadas características de interesse de uma imagem. Para que seja possível a obtenção deste resultado é necessária à remoção de ruídos que estão presentes na imagem. Para a remoção de ruídos pode ser utilizados filtros, e a utilização de cada tipo de filtro depende do tipo do ruído, como já foi explicado anteriormente no capítulo I. As filtragens mediana e gaussiana foram utilizadas nesse projeto como técnicas de remoção dos ruídos na imagem.

Este procedimento recebe a imagem tratada após ter sido aplicado o processo de limiarização. A função *medianBlur* tem como característica a suavização da imagem recebida usando a filtragem mediana. Esta função recebe como parâmetro de entrada o tamanho da abertura linear. O filtro mediano percorre cada pixel da imagem e substitui cada pixel pela mediana de seus pixels vizinhos (localizados em uma vizinhança quadrada ao redor do pixel avaliado). Para utilizar esta função é necessário a importação da biblioteca *imgproc.hpp* oriunda da OpenCV (BRADSKI, 2008).

A função *GaussianBlur* tem como característica a filtragem gaussiana que é feita convolvendo cada ponto na matriz de entrada com um *kernel* gaussiano e, em seguida, somando-os todos para produzir a matriz de saída. Esta função recebe como parâmetro de entrada o tamanho do *kernel* gaussiano. Para utilizar esta função é necessário a importação da biblioteca *imgproc.hpp* oriunda da OpenCV.

Após a aplicação dos filtros era esperado que os ruídos presentes na imagem tivessem sido removidos. Para que a imagem esteja pronta para ser enviada para próxima etapa do sistema é necessário que a área de interesse esteja em evidência. Uma técnica que pode melhorar a imagem neste sentido é a dilatação da imagem. A função *dilate* tem como característica realizar a dilatação da imagem de origem usando um elemento estruturante especificado que determina a forma de uma vizinhança de pixel. Esta função recebe como parâmetro de entrada o número de iterações, ou seja, a quantidade de vezes que o processo de dilatação é aplicado na imagem.

A função *dilate* também recebe um parâmetro de entrada que é gerado a partir de outra função, a *getStructuringElement*. A função *getStructuringElement* retorna um elemento estruturante do tamanho e forma especificados para operações morfológicas. As opções para o tamanho e forma do elemento estruturante são úteis porque permitem usar o elemento estruturante mais adequado para obter determinada área de interesse na imagem.

3.1.5 Procedimentos Específicos dos Equipamentos Industriais

O objetivo principal desta etapa do sistema é conseguir identificar e obter os objetos de interesse na imagem, ou seja, o processo de segmentação da imagem é aplicado nesta etapa e deve parar quando os objetos de interesse em um aplicativo forem isolados. Cada equipamento industrial tem mostradores digitais que possuem características diferentes, o formato dos caracteres apresentados no display, o tamanho de cada caractere, o espaçamento entre os caracteres. Portanto, cada equipamento industrial podem utilizar diferentes técnicas para a realização desta etapa. Nesta etapa a imagem é recebida após a aplicação da operação de dilatação. A partir deste ponto é possível localizar os objetos de interesse na imagem.

A função *findContours* recupera os contornos da imagem binária, que são úteis para análise de formas e detecção e reconhecimento de objetos. Cada contorno detectado é armazenado num vetor de pontos. É possível determinar através de um parâmetro de entrada o modo de recuperação dos contornos, por exemplo, apenas os contornos externos podem ser recuperados. Outro parâmetro de entrada define qual método de aproximação será utilizado na geração dos contornos. Neste projeto foram definidos o modo de recuperação `RETR_LIST` (recupera todos os contornos sem estabelecer uma conexão hierárquica) e o método de aproximação `CHAIN_APPROX_SIMPLE` (comprime segmentos horizontais, verticais e diagonais e deixa apenas seus pontos finais, por exemplo, um contorno retangular para cima é codificado com quatro pontos).

Em seguida, a função *drawContours* desenha os contornos na imagem, para que isso seja possível a função recebe como parâmetro de entrada o vetor de pontos que contém os contornos identificados. É possível definir também a cor de cada contorno desenhado na imagem. Para utilizar as funções *findContours* e *drawContours* é necessário a importação da biblioteca *imgproc.hpp* oriunda da OpenCV.

Após obtermos uma imagem com os contornos identificados, pode-se utilizar uma função para identificar o retângulo mínimo de rotação de cada contorno. OpenCV fornece uma função chamada *minAreaRect* para esta operação. Esta função calcula e retorna o retângulo de delimitação de área mínima (possivelmente rotacionado) para um conjunto de pontos especificado. Esta função recebe como parâmetro de entrada o vetor de pontos com os contornos armazenados.

A utilização da função *minAreaRect* neste projeto é fundamental porque a partir do retângulo de delimitação de área mínima referente a cada contorno, é possível definir se o retângulo obtido será descartado (não é objeto de interesse na imagem) ou será armazenado (é objeto de interesse na imagem). Critérios como proporção, tamanho, largura, ângulo, podem ser usados para definir se determinado retângulo deve ser descartado ou não. Para a utilização da função *minAreaRect* é necessário a importação da biblioteca *imgproc.hpp* oriunda da OpenCV.

Cada retângulo obtido deve ser armazenado numa estrutura de dados adequada. A classe *RotatedRect* representa retângulos rotacionados num plano.

Cada retângulo é especificado por um ponto central, comprimento de cada lado e ângulo de rotação em graus.

Após a obtenção dos retângulos, as informações referentes às localizações dos retângulos devem ser transferidas para a imagem resultante da função *drawContours*. Para que isso seja possível, devem-se obter as referências dos pontos dos retângulos e armazená-las numa estrutura de dados adequada. A classe *Point2f* representa pontos 2D especificados pelas coordenadas *x* e *y*, o tipo de dado armazenado é o *float*. Em seguida, os retângulos serão desenhados na imagem através da função *line*. Esta função recebe como parâmetro de entradas os pontos (*Point2f*) que armazenam as referências para os pontos dos retângulos. Para a utilização das classes *RotatedRect*, *Point2f* e da função *line* é necessário a importação da biblioteca *core.hpp* oriunda da OpenCV.

Como foi citado anteriormente, a função *findContours* retorna todos os contornos identificados na imagem, e como o modo de recuperação *RETR_LIST* retorna todos os tipos de contornos, é possível que para um mesmo objeto de interesse na imagem sejam retornados múltiplos contornos. No entanto, o sistema utiliza uma estratégia que verifica se o centro de massa (ponto central) de um retângulo identificado está a uma distância mínima do centro de massa de outro retângulo. Caso esteja dentro desta distância mínima, os dois retângulos são comparados em relação ao tamanho, o maior retângulo é preservado e o menor será descartado.

O objetivo do sistema é o reconhecimento de padrões de caracteres a partir da leitura de um mostrador digital de um equipamento industrial. Nesta etapa do processo, os retângulos que não foram descartados representam a localização dos caracteres presentes no mostrador digital do equipamento industrial. Portanto, a partir deste ponto será utilizada a imagem resultante do processo de limiarização, pois a imagem até este ponto não sofrera operações como a dilatação, que dificulta o reconhecimento do caractere pela biblioteca Tesseract OCR.

O procedimento a partir deste ponto será extrair e ajustar a inclinação dos caracteres identificados na imagem limiarizada. Caso os caracteres identificados não apresentem inclinação, não se faz necessário o procedimento de ajuste de inclinação. Caso um caractere esteja inclinado, o reconhecimento do padrão do caractere ficará comprometido. Portanto, é necessário verificar o ângulo do

retângulo correspondente ao caractere. A informação do ângulo é um atributo da classe *RotatedRect*. O ângulo pode ser inferior a 45 graus, isso significa que o caractere está alinhado verticalmente, por esse motivo é necessário adicionar 90 graus ao ângulo de rotação e alternar as propriedades de largura e altura.

A função *warpAffine* aplica a transformação afim numa imagem, ou seja, aplicação do processo de rotação da imagem. Esta função transforma uma imagem usando uma matriz específica, chamada de matriz de rotação. A matriz de rotação é obtida a partir da função *getRotationMatrix2D*, que tem como parâmetros de entrada o centro de massa (ponto central) e o ângulo obtido a partir retângulo identificado. Além da matriz de rotação, a função *warpAffine* recebe como parâmetro de entrada o método de interpolação.

O próximo passo é cortar a área do retângulo de delimitação de área mínima referente a cada contorno. Este procedimento é possível utilizando a função *getRectSubPix*. Esta função recupera um retângulo de pixel de uma imagem com precisão de sub-pixel, ou seja, retorna uma imagem cortada. Esta função recebe como parâmetro de entrada a imagem limiarizada e o retângulo de delimitação de área mínima referente a um contorno. No entanto, uma borda deverá ser adicionada a imagem retornada pela função *getRectSubPix*.

A função *copyMakeBorder* forma uma borda em torno de uma imagem. Isso é importante porque a etapa de classificação dos objetos geralmente espera uma margem ao redor do texto. Esta função recebe como parâmetro de entrada a imagem retornada da função *getRectSubPix* e a espessura da borda em torno da parte superior, inferior, esquerda e direita. Para a utilização das funções *warpAffine*, *getRotationMatrix2D*, *getRectSubPix* e *copyMakeBorder* é necessário a importação da biblioteca *imgproc.hpp* oriunda da OpenCV.

Nesta etapa do sistema é possível ter várias imagens recortadas enfatizando cada um dos caracteres presentes no mostrador digital ou ter uma única imagem recortada com um conjunto de caracteres. Portanto, o passo seguinte é salvar a imagem tratada até este ponto, num arquivo específico com formato da imagem definido. Para realizar este procedimento foi utilizada a função *imwrite* que tem como característica salvar uma imagem em um arquivo específico. Esta função recebe como parâmetro de entrada o nome do arquivo, a imagem retornada da função *copyMakeBorder* e o formato específico da imagem (JPEG, PNG, entre outros). Para

a utilização da função *imwrite* é necessário a importação da biblioteca *highgui.hpp* oriunda da OpenCV.

A melhoria obtida na imagem até esta etapa do sistema pode ser suficiente ou não, isto pode variar a depender da qualidade da imagem obtida a partir dos equipamentos industriais. Caso a imagem processada até este ponto esteja com uma qualidade considerada suficiente, o próximo passo a ser executado refere-se ao reconhecimento de padrões de caracteres. Caso a imagem processada até este ponto esteja com uma qualidade considerada insuficiente, o próximo passo será a repetição de alguns procedimentos já executados. Os procedimentos a serem executados variam de uma imagem do equipamento industrial para outro.

Algumas funções como *medianBlur*, *GaussianBlur*, *dilate*, *findContours*, *drawContours*, *minAreaRect*, *line*, *getRectSubPix* podem ser executadas novamente sobre a imagem anteriormente processada com objetivo de melhorar a qualidade, aumentando o destaque da área de interesse na imagem. Outras funções que não foram executadas anteriormente também podem ser utilizadas nesta nova etapa. Por exemplo, neste projeto foram utilizadas as funções *erode* e *morphologyEx*.

A função *erode* aplica a operação de erosão a uma imagem utilizando um específico elemento estruturante. Esta função recebe como parâmetro de entrada o número de iterações, ou seja, a quantidade de vezes que o processo de erosão é aplicado na imagem. O processo para obtenção do elemento estruturante é idêntico ao citado anteriormente para função *dilate*.

A função *morphologyEx* realiza transformações morfológicas avançadas usando a dilatação e a erosão como operações básicas. Esta função recebe como parâmetro de entrada o elemento estruturante, o número de iterações que a operação é aplicada e o tipo da operação morfológica. Os tipos de operação morfológicas utilizadas neste sistema foram a operações de abertura e fechamento. No caso deste sistema, estas operações são úteis para situações em que os caracteres são mostrados em displays de sete segmentos, devido ao espaço presente entre os segmentos o reconhecimento de padrões dos caracteres fica comprometido. A operação de fechamento pode ajustar estes espaços entre os segmentos. Para a utilização das funções *erode* e *morphologyEx* é necessário a importação da biblioteca *imgproc.hpp* oriunda da OpenCV.

3.1.6 Reconhecimento de Padrões dos Caracteres

O objetivo dessa etapa do sistema consiste em identificar cada letra ou dígito presente no mostrador digital. Nesta etapa do sistema tem-se um ou mais arquivos com extensões (JPEG) que armazenam a imagem tratada pelos procedimentos aplicados anteriormente. Como já foi explicada anteriormente, a biblioteca Tesseract OCR é responsável por realizar o reconhecimento de padrões dos caracteres.

Para utilizar as funções da biblioteca Tesseract OCR é necessário a importação da biblioteca *baseapi.h* oriunda da Tesseract. Em seguida, um objeto *TessBaseAPI* deverá ser instanciado e representará o motor Tesseract OCR. Este objeto fornece as funções necessárias para a realização do reconhecimento de padrões de caracteres.

Antes de utilizar qualquer função é preciso inicializar o motor Tesseract OCR e isto é feito pela função *init*. Esta função tem como parâmetro de entrada o caminho de dados para o *tessdata*, ou seja, onde estão localizados os arquivos das linguagens instaladas. Outro parâmetro de entrada é o código da linguagem utilizada, por exemplo, *eng* para Inglês, *por* para Português, *hin* para Hindu, entre outras. Caso seja realizado um treinamento para reconhecer uma nova linguagem (Tesseract fornece este recurso) será definido um código para essa linguagem.

Neste projeto, foi realizado um treinamento para a geração de uma nova linguagem, com o objetivo de facilitar o reconhecimento de caracteres apresentados em displays de sete segmentos. Não faz parte do escopo do projeto demonstrar o passo a passo da realização do treinamento de uma nova linguagem no Tesseract OCR, no entanto pode-se dizer que uma imagem com os caracteres da nova linguagem é parâmetro de entrada para a realização do treinamento, bem como um arquivo de texto com as letras e os dígitos referentes aos caracteres da imagem.

Após o motor Tesseract OCR ser inicializado é preciso que a imagem utilizada para a operação de reconhecimento de padrões de caracteres seja lida. A função *pixRead* abre uma imagem com a biblioteca Leptonica, portanto, para utilizar essa função é necessário a importação da biblioteca Leptonica. As informações da imagem lida são armazenadas numa estrutura de dados chamada *Pix* (oriunda da *baseapi.h*).

O próximo passo é fornecer a imagem lida para que o motor Tesseract OCR realize o reconhecimento. Isto pode ser realizado utilizando a função *SetImage*, esta função recebe como parâmetro de entrada a estrutura de dados *Pix* que armazena a imagem lida anteriormente.

A última função executada *GetUTF8Text* retornará o caractere/texto reconhecido, codificado como UTF8. Após a obtenção do texto reconhecido, o mesmo será exibido no terminal de simulação.

3.2 RESULTADOS

Nesta seção serão apresentados os resultados obtidos a partir da simulação do sistema. Os resultados serão apresentados através das etapas do sistema, onde cada etapa proporciona uma imagem como resultado final. O resultado obtido para última etapa será a exibição do texto reconhecido no terminal de simulação.

3.2.1 Processo de Aquisição da Imagem

Para a realização dos testes com o intuito de validar o funcionamento do sistema foram definidos dois equipamentos industriais. A partir dos equipamentos definidos foram obtidas as imagens utilizadas como entrada para o sistema. Vale ressaltar que duas abordagens de verificação foram utilizadas, com o intuito de obter cobertura para a maioria dos equipamentos industriais. Uma abordagem foi utilizar um equipamento industrial com mostradores digitais do tipo display de sete segmentos e a outra foi utilizar um equipamento industrial com mostradores digitais do tipo display LCD.

O *Thermo Scientific Orion Versa Star Pro* é um equipamento industrial com a capacidade de realizar determinadas medições como:

- Medição de pH
- Concentração de íons
- Temperatura
- Condutividade

O *Thermo Scientific Orion Versa Star Pro* fornece um display do tipo LCD (*Liquid Crystal Display*).

O N1540 da empresa *Novus* é um indicador de processo que possui uma ampla lista de tipos de entrada como termopares, termoresistência e sinais lineares de tensão e correntes elétricas, que permitem ao dispositivo indicar as mais diversas variáveis nos mais diversos processos. O N1540 fornece um display de sete segmentos.

Referente à etapa de aquisição de imagem, o primeiro passo é a inicialização da câmera para obtenção de um quadro da imagem. A figura 28 apresenta as imagens obtidas a partir dos equipamentos *Novus N1540* e *Orion Versa Star Pro*.



Figura 28 - Novus N1540 à esquerda, Orion Versa Star Pro à direita (Fonte: Próprio Autor)

3.2.2 Binarização da Imagem

Como explicado na seção 3.1.3 o processo de binarização da imagem é dividido em duas partes. Na primeira parte ocorre a conversão da imagem colorida (como pode ser vista na figura 28) para uma imagem em tons de cinza. O resultado para esta operação pode ser visualizado na figura 29.



Figura 29 - Tons de Cinza – N1540 à esquerda, Orion Versa Star Pro à direita (Fonte: Próprio Autor)

A segunda parte do processo de binarização da imagem refere-se à limiarização das imagens apresentadas na figura 29. Com *threshold* definido com o valor igual a 100, obtêm-se os resultados apresentados na figura 30.

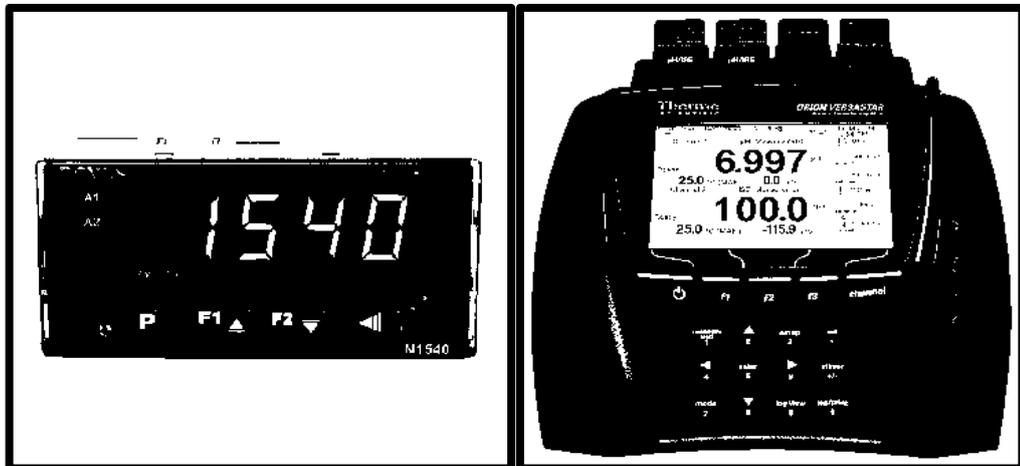


Figura 30 - Limiarização – N1540 à esquerda, Orion Versa Star Pro à direita (Fonte: Próprio Autor)

3.2.3 Procedimento de Melhoria de Imagem

Como apresentado na seção 3.1.4, esta etapa do sistema efetua a suavização das imagens com objetivo de eliminar/reduzir os ruídos presentes na imagem. Este procedimento é definido em três partes, na primeira parte um filtro mediano é aplicada a imagem, no entanto sua utilização depende de um parâmetro de entrada. Para o equipamento N1540 a filtragem mediana não foi necessária nesta etapa. O resultado da aplicação da filtragem mediana para o equipamento Orion Versa Star Pro é apresentado na figura 31.

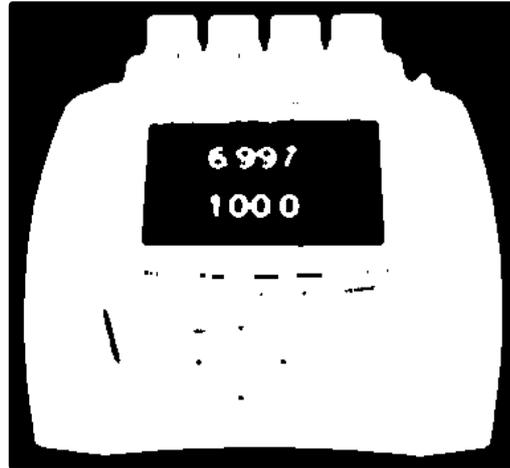


Figura 31 - Filtragem mediana equipamento Orion Versa Star Pro (Fonte: Próprio Autor)

Na segunda parte a filtragem gaussiana é aplicada, no entanto a sua utilização depende de um parâmetro de entrada. O resultado da aplicação da filtragem gaussiana é apresentado na figura 32.

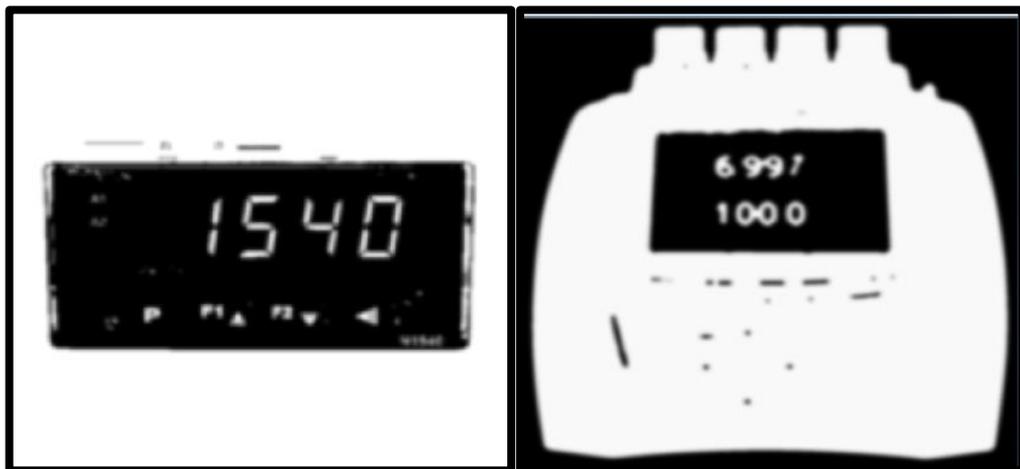


Figura 32 - Gaussiana – N1540 à esquerda, Orion Versa Star Pro à direita (Fonte: Próprio Autor)

Na terceira parte a operação de dilatação é aplicada na imagem, com o objetivo de destacar o objeto de interesse na imagem. O resultado desta operação pode ser visualizado na figura 33.

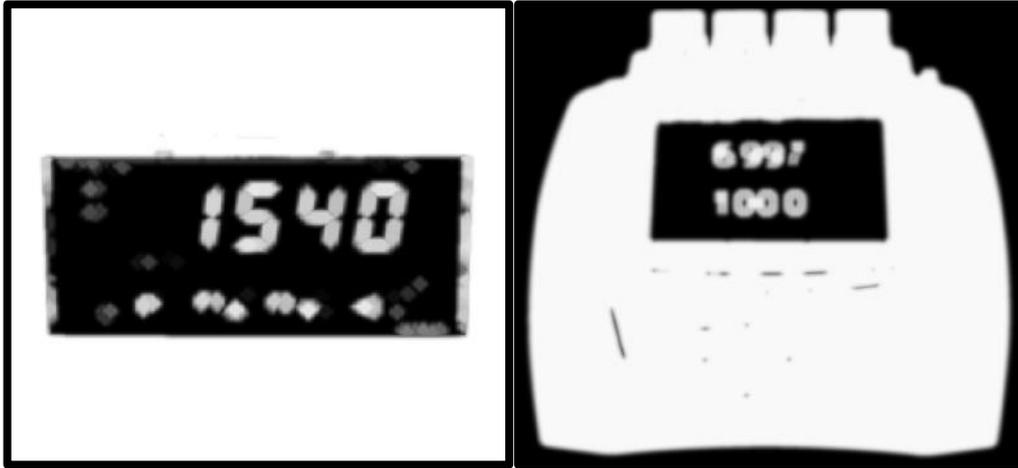


Figura 33 - Dilatação N1540 à esquerda, Orion Versa Star Pro à direita (Fonte: Próprio Autor)

3.2.4 Procedimentos Específicos dos Equipamentos Industriais

Como apresentado na seção 3.1.5, a etapa de procedimento específico dos equipamentos industriais pode apresentar variação na quantidade de operações utilizadas, bem como nas técnicas aplicadas. No entanto, algumas operações foram utilizadas nos dois equipamentos industriais apresentados. A primeira operação refere-se à detecção de contornos na imagem, o resultado desta operação é apresentado na figura 34.

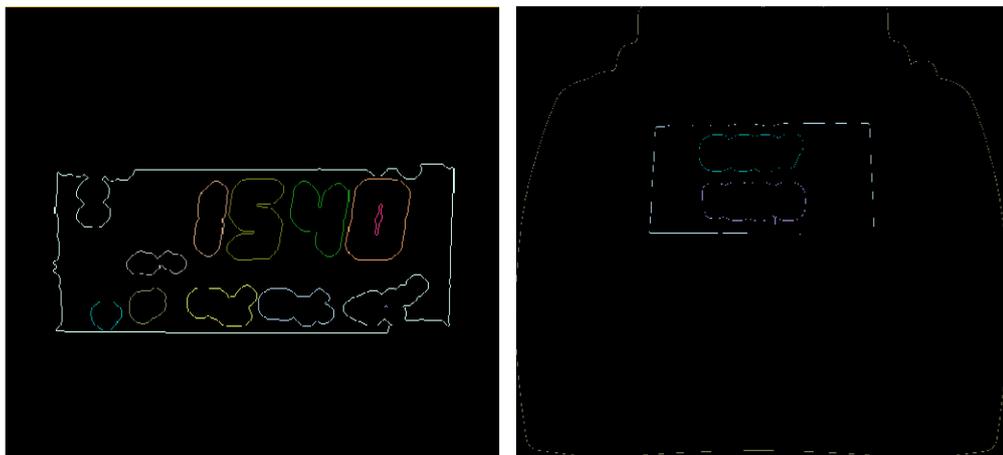


Figura 34 - Contornos N1540 à esquerda, Orion Versa Star Pro à direita (Fonte: Próprio Autor)

Após a detecção de contornos, apenas o contornos referentes aos objetos de interesse serão obtidos, isto é feito identificando as regiões com os retângulos

delimitadores de área mínima. Os retângulos (na cor verde) podem ser visualizados na figura 35.

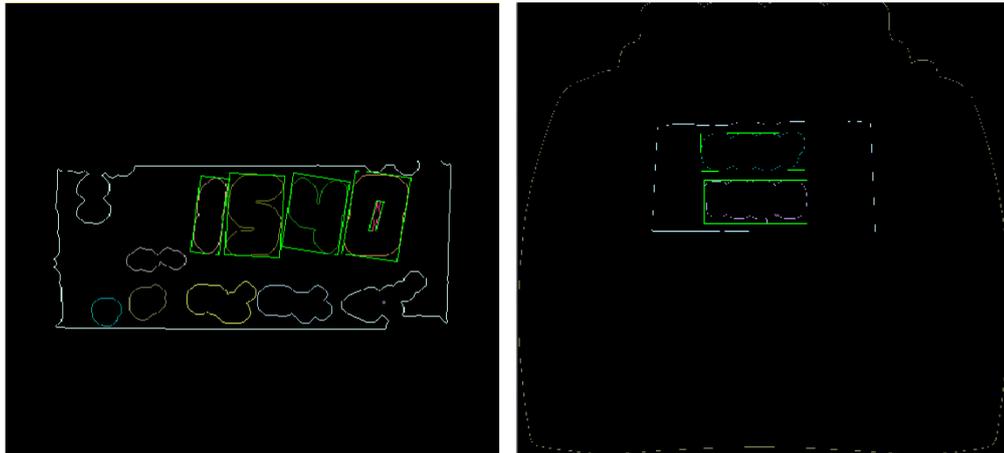


Figura 35 - Retângulos N1540 à esquerda, Orion Versa Star Pro à direita (Fonte: Próprio Autor)

Com as regiões de interesse das imagens identificadas, o passo seguinte é recortar as áreas definidas e armazenar numa estrutura de dados adequada. A partir da figura 35 percebe-se que 4 retângulos foram identificados para o equipamento N1540 e 2 retângulos foram identificados para o equipamento Orion Versa Star Pro. Isto evidencia que estratégias diferentes foram utilizadas para obter a área de interesse na imagem.

Para o equipamento N1540 as áreas definidas que foram recortadas da imagem processada são apresentadas na figura 36.



Figura 36 - Objetos de Interesse na Imagem equipamento N1540 (Fonte: Próprio Autor)

Para o equipamento Orion Versa Star Pro as áreas definidas que foram recortadas da imagem processada são apresentadas na figura 37.

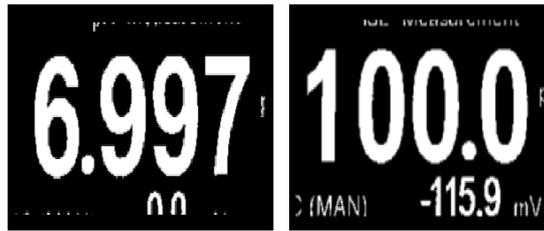


Figura 37 - Objetos de Interesse na Imagem equipamento Orion Versa Star Pro (Fonte: Próprio Autor)

Com a obtenção dos objetos de interesse na imagem é possível efetuar técnicas de suavização, operações morfológicas, entre outras operações que visam a melhoria da região de interesse na imagem. Para o equipamento N1540 são aplicadas duas operações morfológicas na imagem, a primeira refere-se à operação de fechamento utilizando o elemento estruturante da forma elipse. O resultado desta operação pode ser visualizado na figura 38.



Figura 38 - 1º Operação de fechamento equipamento N1540 (Fonte: Próprio Autor)

A segunda operação refere-se à operação de fechamento utilizando o elemento estruturante da forma retângulo. O resultado desta operação pode ser visualizado na figura 39.



Figura 39 - 2º Operação de fechamento equipamento N1540 (Fonte: Próprio Autor)

Para o equipamento Orion Versa Star Pro são definidas 5 operações incluindo técnicas de suavização e operações morfológicas, no entanto essas operações são aplicadas 5 vezes com o objetivo de reduzir ao máximo a quantidade de ruído na

região de interesse na imagem. O resultado da primeira iteração é apresentado na figura 40.

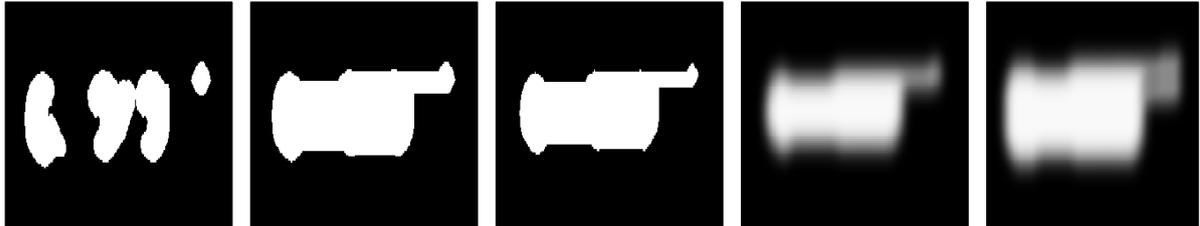


Figura 40 - Sequência de operações equipamento Orion Versa Star Pro (Fonte: Próprio Autor)

A sequência de operações apresentada na figura 40, no sentido da esquerda para direita: Filtragem mediana, operação de fechamento, operação de erosão, filtragem gaussiana e operação de dilatação. Após todas as iterações serem efetuadas tem-se como resultado as imagens da figura 41.



Figura 41 - Resultado após 5 iterações equipamento Orion Versa Star Pro (Fonte: Próprio Autor)

3.2.5 Reconhecimento de Padrões dos Caracteres

Como apresentado na seção 3.1.6, para a realização do reconhecimento de padrões de caracteres foi utilizada a biblioteca Tesseract OCR. Todas as etapas de processamento anteriores visavam entregar como entrada para a Tesseract, uma imagem em boas condições para o reconhecimento de padrões de caracteres.

O resultado do reconhecimento de padrões de caracteres efetuado pela Tesseract OCR para ambos equipamentos é apresentado na figura 42.

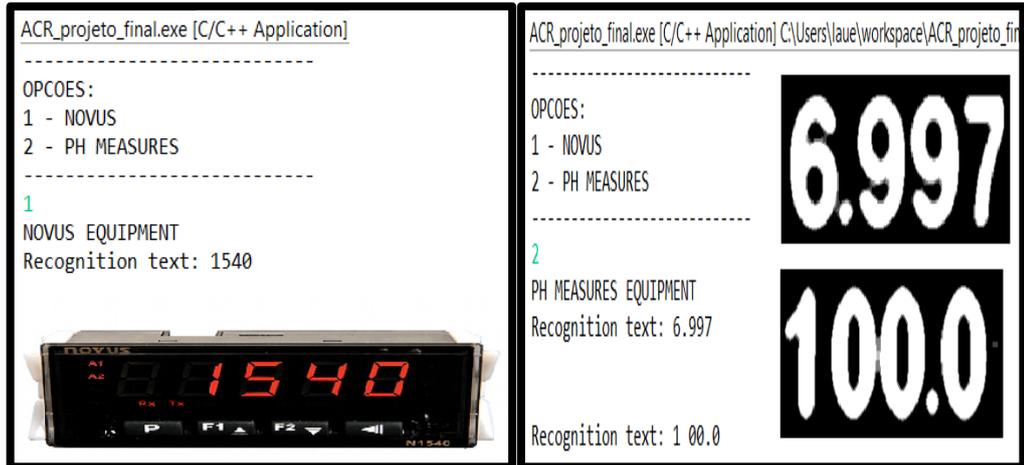


Figura 42 - Reconhecimento - N1540 à esquerda Orion Versa Star Pro à direita (Fonte: Próprio Autor)

CONSIDERAÇÕES FINAIS

O projeto de conclusão de curso da especialização em Automação, Controle e Robótica apresentou como proposta a elaboração de um sistema para o reconhecimento de padrões de caracteres de mostradores digitais de ambientes industriais. O projeto abordou diferentes áreas como processamento digital de imagens, algoritmos e programação, visão computacional e integrou determinados conceitos para aplicação na automação industrial.

O desenvolvimento do projeto atingiu os resultados esperados com a elaboração e implementação de uma arquitetura para o sistema. A integração entre o processo de codificação no ambiente de programação usando linguagem C++, a biblioteca OpenCV e a biblioteca Tesseract OCR foi obtida com êxito. O sistema apresentou a característica de flexibilidade, pois novos métodos e funções voltados para diferentes mostradores digitais de equipamentos industriais podem ser adicionados.

A metodologia de desenvolvimento adotada mostrou-se adequada para a implementação do sistema, proporcionando a aplicação dos estudos teóricos apresentados na fase de codificação das etapas do sistema. A utilização de ferramentas de código aberto, como Eclipse CDT, OpenCV e Tesseract OCR, proporcionou a criação de um sistema voltado para automação industrial de baixo custo.

Os resultados obtidos a partir das simulações realizadas no ambiente de desenvolvimento Eclipse CDT tiveram seus objetivos atingidos, pois o reconhecimento de padrões de caracteres foi realizado sem apresentar problemas. Os resultados foram gerados a partir de diferentes mostradores digitais de equipamentos industriais, comprovando a flexibilidade do sistema e demonstrando a possibilidade de adicionar novos requisitos.

Trabalhos futuros que aperfeiçoem o sistema inclui agregar novos módulos ou componentes de processamento digital de imagens e a refatoração do código visando melhorar a estrutura interna e conseqüentemente a eficiência.

REFERÊNCIAS BIBLIOGRÁFICAS

- ANNADURAI, S. **Fundamentals of Digital Image Processing**. 1st edition. Pearson India, 2006.
- BRADSKI, Gary; KAEHLER, Adrian. **Learning OpenCV**. 1st edition. O'Reilly Media, 2008.
- CRUZ, Juliano Elias Cardoso. **Reconhecimento de objetos em imagens orbitais com o uso de abordagens do tipo descritor-classificador**. São José dos Campos, 2014, INPE.
- EKSTROM, Michael P. **Digital Image Processing Techniques**. 1st edition. Orlando: Academic Press, 1984.
- GODOY, Vinícius; ESCRIVÁ, David Millan; JOSHI, Prateek. **OpenCV By Example**. Packt Publishing, 2016.
- GONZALEZ, Rafael C. **Digital Image Processing**. Second Edition. New Jersey: Prentice Hall, 2002.
- HALTERMAN, L. Richard. **Fundamentals of C++ Programming**. Southern Adventist University, 2017.
- JAHNE, Bernd. **Digital Image Processing**. 6th edition. Berlin: Springer – Verlag Berlin Heidelberg, 2005.
- JAIN, Ramesh. **Machine Vision**. New York: McGraw-Hill, 1995.
- LAGANIERE, Robert. **OpenCV 2 Computer Vision Application Programming Cookbook**. 1st edition. Birmingham: Packt Publishing, 2011.
- LOPES, Eymar et al. **Teoria : Processamento de Imagens**. Divisão de Processamento de Imagens, INPE, 2017.
- MARENGONI, Maurício; STRINGHINI, Denise. **Tutorial: Introdução à Visão Computacional usando OpenCV**. São Paulo, 2009, Universidade Presbiteriana Mackenzie, Faculdade de Computação e Informática e Pós Graduação em Engenharia Elétrica.
- MARQUES, Ogê Filho. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999.
- QUEIROZ, J. E. R.; GOMES, H. M. **Introdução ao Processamento Digital de Imagens**. Campina Grande, 2001, Departamento de Sistemas e Computação, UFCG.
- SMITH, Ray. **An Overview of the Tesseract OCR**. Google Inc, 2007, IEEE.

STROUSTRUP, Bjarne. **The C++ Programming Language**. Fourth Edition. New Jersey: Pearson Education, 2013.

SZELISKI, Richard. **Computer Vision: Algorithms and Applications**. Springer, 2010.