



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

**Roberto Espinheira da Costa Bomfim**

**Visão Computacional aplicada ao Controle de Qualidade de  
Casulos do Bicho-da-Seda: *Subsistema de Segmentação de  
Imagens***

Salvador  
Março, 2015

Roberto Espinheira da Costa Bomfim

**Visão Computacional aplicada ao Controle de Qualidade de  
Casulos do Bicho-da-Seda: *Subsistema de Segmentação de  
Imagens***

Trabalho apresentado ao Curso de Especialização em  
Automação e Controle da Faculdade de Tecnologia SENAI  
CIMATEC como parte dos requisitos para a obtenção do  
título de Especialista em Automação e Controle.

Orientador: Bruno Guimarães Costa Alves

Salvador  
Março, 2015

Ficha catalográfica elaborada pela Biblioteca da Faculdade de Tecnologia SENAI CIMATEC

B695v Bomfim, Roberto Espinheira da Costa

Visão computacional aplicada ao controle de qualidade de casulos do bicho-da-seda: subsistema de segmentação de imagens / Roberto Espinheira da Costa Bomfim – Salvador, 2015.

131 f. : il. color.

Orientador: MSc. Bruno Guimarães Costa Alves.

Monografia (Especialização em Automação e Controle) – Programa de Pós-Graduação, Faculdade de Tecnologia SENAI CIMATEC, Salvador, 2015.

Inclui referências.

1. Visão computacional. 2. Produção de seda. 3. Segmentação de imagens. 4. Processamento de imagens. 5. Casulo do bicho-da-seda. I. Faculdade de Tecnologia SENAI CIMATEC. II. Alves, Bruno Guimarães Costa. III. Título.

CDD 629.8

*Este trabalho é dedicado a todos aqueles que lutam, cada qual ao seu modo, por um mundo menos desigual. Se a gente canta em coro é mais forte o som da nossa voz!*

# Agradecimentos

Agradeço primeiramente aos amigos e familiares, que contribuem diariamente para o meu crescimento pessoal (crescimento este que considero mais importante que qualquer crescimento profissional). Sou grato também ao meu orientador Bruno Alves, idealizador do projeto, pela sua disponibilidade, paciência e pela confiança no meu trabalho. Finalmente, agradeço à Fiação de Seda BRATAC por ter disponibilizado as informações necessárias à realização do trabalho e ao SENAI CIMATEC, pela oportunidade de formação e crescimento profissional.

# Resumo

Ao longo da cadeia de produção da seda, os casulos do bicho-da-seda passam por sucessivas etapas de controle de qualidade, normalmente envolvendo inspeções visuais feitas por pessoas. No entanto, o cansaço físico e psicológico provocado pela execução repetitiva de tal tarefa traz prejuízos tanto à saúde dos trabalhadores quanto à qualidade do produto final. Diante desse contexto, o objetivo do projeto a longo prazo é automatizar o processo de detecção, classificação e retirada dos casulos que não atendam aos critérios de qualidade. O objetivo deste trabalho, por sua vez, é *analisar os diferentes métodos de segmentação de imagens existentes, com foco na detecção de casulos do bicho-da-seda*. A metodologia utilizada inclui a construção de um banco de imagens dos casulos e a implementação e comparação de 3 métodos diferentes de segmentação de imagens, escolhidos com base no cruzamento entre as características do problema e as características dos algoritmos disponíveis na literatura. Ao final, os resultados dos 3 algoritmos de segmentação são apresentados e é feita uma análise comparativa de seus respectivos desempenhos, vantagens e desvantagens.

**Palavras-chaves:** casulo do bicho-da-seda, produção de seda, segmentação de imagens, processamento de imagens, visão computacional.

# Abstract

Along the silk production chain, silkworm's cocoons pass through successive quality control steps, usually involving human visual inspections. However, the physical and psychological fatigue caused by repetitive execution of such a task can impair workers' health and the quality of the final product. In this context, the long-term objective of the project is to automate the whole process of detection, classification and removal of cocoons that don't meet the quality criteria. The objective of this work, in turn, is to *analyze the different existing image segmentation methods, focusing on the detection of silkworm's cocoons*. The methodology includes the generation of a cocoons' image bank and the implementation and comparison of 3 different image segmentation methods, chosen by a comparison of the characteristics of the problem and the characteristics of the algorithms available in the literature. In the end, the results of the 3 segmentation algorithms are presented and a comparative analysis of their performance, advantages and disadvantages is made.

**Key-words:** silkworm cocoon, silk production, image segmentation, image processing, computer vision.

# Lista de ilustrações

Figura 1 – Controle de qualidade dos casulos em uma linha de produção de fios de seda	17
Figura 2 – Exemplos de novas aplicações da seda	25
Figura 3 – Principais produtores mundiais de seda	26
Figura 4 – Evolução da produção de seda no Brasil ao longo dos anos	27
Figura 5 – Processo de produção da seda e suas respectivas entradas e saídas	28
Figura 6 – Ciclo de vida do Bicho-da-Seda ( <i>Bombyx mori</i> )	29
Figura 7 – Processo de produção da fibra têxtil da seda	31
Figura 8 – Exemplos de defeitos comuns em casulos do bicho-da-seda.	34
Figura 9 – Quadro de defeitos comuns em casulos do bicho-da-seda	34
Figura 10 – Relação existente entre a área de processamento de imagens e disciplinas correlatas	36
Figura 11 – Imagens obtidas a partir de diferentes tipos de sensores	38
Figura 12 – Imagens do Pulsar de Caranguejo obtidas em diferentes regiões do espectro eletromagnético	38
Figura 13 – Convenção utilizada na representação espacial de uma imagem	39
Figura 14 – Imagem de uma arara-canindé representada em 2 espaços de cores diferentes.	39
Figura 15 – Representação gráfica do espaço de cores RGB	40
Figura 16 – Exemplo da aplicação de operações morfológicas utilizando elementos estruturantes de formatos distintos	42
Figura 17 – Exemplo da aplicação de um processo de equalização de histograma	43
Figura 18 – Exemplo da aplicação de um algoritmo de correção de iluminação feito em MATLAB	44
Figura 19 – Exemplo da aplicação de um algoritmo de detecção de bordas	44
Figura 20 – Diagrama dos principais processos aplicados a imagens	45
Figura 21 – Diagrama de blocos das principais etapas realizadas por um sistema de visão computacional	49
Figura 22 – Diagrama de um sistema de visão computacional genérico voltado a aplicações industriais	51
Figura 23 – Especificações de projeto de um sistema de visão computacional	55
Figura 24 – Resultado de um processo de segmentação realizado em uma imagem simples	56
Figura 25 – Histograma de duas imagens em escala de cinza compostas por objetos claros em um plano de fundo escuro	60
Figura 26 – Elementos de uma paisagem utilizados na construção da analogia feita com o método de segmentação por <i>watershed</i> (Bacia Hidrográfica).	63

Figura 27 – Visualização da progressão do algoritmo de segmentação por <i>watershed</i> aplicado a uma imagem artificial. . . . .	64
Figura 28 – Evolução das aplicações possíveis da transformada de Hough . . . . .	66
Figura 29 – Aplicação da transformada de Hough a uma reta no espaço cartesiano - Caso 01. . . . .	67
Figura 30 – Aplicação da transformada de Hough a um ponto no espaço cartesiano - Caso 02. . . . .	68
Figura 31 – Parâmetros $\rho$ e $\theta$ para representação de uma reta em coordenadas polares. . . . .	68
Figura 32 – Aplicação da transformada de Hough a uma reta no espaço cartesiano - Caso 03. . . . .	69
Figura 33 – Aplicação da transformada de Hough a um ponto no espaço cartesiano - Caso 04. . . . .	69
Figura 34 – Exemplo de aplicação da transformada de Hough. . . . .	70
Figura 35 – Representação de uma forma geométrica arbitrária para definição dos parâmetros da GHT . . . . .	73
Figura 36 – Exemplos de imagens dos casulos do bicho-da-seda . . . . .	77
Figura 37 – Exemplo de imagens de cada um dos cenários de teste escolhidos . . . . .	80
Figura 38 – Imagens dos casulos bons utilizadas nos algoritmos . . . . .	81
Figura 39 – Imagens dos casulos defeituosos utilizadas nos algoritmos . . . . .	82
Figura 40 – Fluxograma do algoritmo de segmentação - Método do <i>Threshold</i> . . . . .	84
Figura 41 – Exemplo da evolução da imagem ao longo do processo de segmentação - Método do <i>Threshold</i> . . . . .	86
Figura 42 – Resultado da segmentação do Subconjunto 01 do conjunto de casulos bons - Método do <i>Threshold</i> . . . . .	87
Figura 43 – Resultado da segmentação do Subconjunto 01 do conjunto de casulos defeituosos - Método do <i>Threshold</i> . . . . .	88
Figura 44 – Fluxograma do algoritmo de segmentação - Método do <i>Watershed</i> . . . . .	92
Figura 45 – Exemplo da evolução da imagem ao longo do processo de segmentação - Método do <i>Watershed</i> . . . . .	95
Figura 46 – Resultado da segmentação do conjunto de casulos bons - Método do <i>Watershed</i> . . . . .	96
Figura 47 – Resultado da segmentação do conjunto de casulos defeituosos - Método do <i>Watershed</i> . . . . .	97
Figura 48 – Exemplo de curva da família das superelipses (para $n > 2$ ) . . . . .	103
Figura 49 – Influência do parâmetro $n$ na curva da superelipse . . . . .	103
Figura 50 – Definição dos pontos e parâmetros da elipse utilizados no algoritmo . . . . .	104
Figura 51 – Fluxograma do algoritmo de segmentação - Método da Transformada de Hough . . . . .	107
Figura 52 – Exemplo da evolução da imagem ao longo do processo de segmentação - Método da Transformada de Hough . . . . .	109

Figura 53 – Resultado da segmentação do conjunto de casulos bons - Método da Transformada de Hough . . . . .	110
Figura 54 – Resultado da segmentação do conjunto de casulos defeituosos - Método da Transformada de Hough . . . . .	111

# Lista de tabelas

Tabela 1 – Etapas de projeto de um sistema de visão computacional . . . . .	53
Tabela 2 – Principais dificuldades inerentes ao projeto de um sistema de visão computacional . . . . .	54
Tabela 3 – Resumo dos resultados da segmentação - Método do <i>Threshold</i> . . . . .	90
Tabela 4 – Resumo dos resultados da segmentação - Método do <i>Watershed</i> . . . . .	100
Tabela 5 – Resumo dos resultados da segmentação - Método da <i>Transformada de Hough</i>	114
Tabela 6 – Resumo dos resultados da segmentação - Comparação entre os métodos utilizados . . . . .	116
Tabela 9 – <i>Ranking</i> mundial de produção de casulos do bicho-da-seda . . . . .	129
Tabela 10 – <i>Ranking</i> mundial de produção de seda crua . . . . .	130

# Lista de abreviaturas e siglas

$\mu$ C	MicroControlador
1D, 2D, 3D	1, 2 ou 3 Dimensões
API	<i>Application Programming Interface</i>
ASIC	<i>Application-Specific Integrated Circuit</i>
CIMATEC	Centro Integrado de Manufatura e Tecnologia
CTMV	<i>Consulting Team Machine Vision</i>
DSP	<i>Digital Signal Processor</i>
FAO	<i>Food and Agriculture Organization of the United Nations</i>
FAOSTAT	<i>Food and Agriculture Organization Corporate Statistical Database</i>
FIEB	Federação das Indústrias do Estado da Bahia
FPGA	<i>Field-Programmable Gate Array</i>
(G)HT	<i>(Generalized) Hough Transform</i>
GIMP	<i>GNU Image Manipulation Program</i>
LDA	Linha Divisora de Águas
LER	Lesão por Efeito Repetitivo
LUT	<i>Look-Up Table</i>
PC	<i>Personal Computer</i>
RANSAC	<i>Random Sample Consensus</i>
SENAI	Serviço Nacional de Aprendizagem Industrial

*Continua na próxima página...*

### ***Extensões de Arquivos de Imagem***

BMP	<i>Bitmap</i>
JP(E)G	<i>Joint Photographic Experts Group</i>
PNG	<i>Portable Network Graphics</i>
TIFF	<i>Tagged Image File Format</i>

### ***Espaços de Cor\****

CIE	<i>Commission Internationale de l'Eclairage</i>
CMY(K)	<i>Cyan, Magenta Yellow and Key (Black)</i>
HSV	<i>Hue, Saturation and Value</i>
RGB	<i>Red, Green and Blue</i>

---

\*Informações sobre as diferentes aplicações de cada espaço de cor no Tópico 7.9.4.3 de ([HORNBERG, 2006](#)).

# Sumário

<b>1</b>	<b>Introdução</b>	<b>16</b>
1.1	Contexto do Trabalho	16
1.2	Definição do Problema	17
1.3	Objetivo	18
1.4	Motivação	18
1.5	Importância da Pesquisa	19
1.6	Limites e Limitações	20
1.7	Hipóteses	21
1.8	Aspectos Metodológicos	22
1.9	Organização da Monografia	23
<b>2</b>	<b>Produção da Seda</b>	<b>24</b>
2.1	Introdução	24
2.2	A Seda	24
2.2.1	Contexto Econômico Mundial	26
2.2.2	Contexto Econômico Nacional	27
2.3	Processo de Produção da Seda	28
2.3.1	Etapa 1: Processo de Produção dos Casulos do Bicho-da-Seda	29
2.3.2	Etapas 2 e 3: Processo de Produção da Fibra Têxtil da Seda	31
2.3.3	Etapa 4: Processo de Produção do Tecido de Seda	33
2.4	Controle de Qualidade dos Casulos	33
<b>3</b>	<b>Processamento de Imagens &amp; Visão Computacional</b>	<b>35</b>
3.1	Introdução	35
3.2	Relação entre as Disciplinas	35
3.2.1	Classificação de acordo com a Entrada e a Saída do Processo	36
3.2.2	Classificação de acordo com o Nível do Processo	37
3.3	Processamento de Imagens	38
3.3.1	Processamento de Imagens Digitais	39
3.3.2	Operações Básicas	41
3.3.3	Principais Processos	45
3.4	Visão Computacional	48
3.4.1	Sistemas de Visão Computacional aplicados à Indústria	50
3.4.2	Projeto de Sistemas de Visão Computacional	53
<b>4</b>	<b>Segmentação de Imagens</b>	<b>56</b>

4.1	Introdução . . . . .	56
4.2	Processo de Segmentação de Imagens . . . . .	56
4.2.1	Conceito . . . . .	56
4.2.2	Definição Matemática . . . . .	57
4.2.3	Principais Métodos . . . . .	58
4.2.4	Métodos Utilizados neste Trabalho . . . . .	59
4.3	Segmentação pelo Método do <i>Threshold</i> . . . . .	60
4.3.1	Conceito . . . . .	60
4.3.2	Escolha do Limiar . . . . .	61
4.4	Segmentação pelo Método do <i>Watershed</i> . . . . .	63
4.4.1	Conceito . . . . .	63
4.4.2	Utilização de Marcadores . . . . .	65
4.5	Segmentação pelo Método da Transformada de Hough . . . . .	66
4.5.1	Conceito . . . . .	66
4.5.2	Detecção de Linhas . . . . .	67
4.5.3	Detecção de Círculos . . . . .	71
4.5.4	Detecção de Elipses . . . . .	72
4.5.5	Detecção de Formas Arbitrárias . . . . .	73
4.5.6	Variações da Transformada de Hough . . . . .	74
<b>5</b>	<b>Desenvolvimento da Pesquisa . . . . .</b>	<b>76</b>
5.1	Introdução . . . . .	76
5.2	Escolha dos Métodos de Segmentação de Imagens . . . . .	76
5.2.1	Características do Problema . . . . .	76
5.2.2	Métodos Escolhidos . . . . .	78
5.3	Metodologia & Cenários de Teste . . . . .	79
5.4	Segmentação pelo Método do <i>Threshold</i> . . . . .	84
5.4.1	Algoritmo . . . . .	84
5.4.2	Resultados & Discussão . . . . .	86
5.5	Segmentação pelo Método do <i>Watershed</i> . . . . .	92
5.5.1	Algoritmo . . . . .	92
5.5.2	Resultados & Discussão . . . . .	95
5.6	Segmentação pelo Método da Transformada de Hough . . . . .	102
5.6.1	Modelagem do Problema . . . . .	102
5.6.2	Algoritmo . . . . .	104
5.6.3	Resultados & Discussão . . . . .	109
5.7	Comparação dos Resultados . . . . .	116
<b>6</b>	<b>Considerações finais . . . . .</b>	<b>117</b>
6.1	Conclusões . . . . .	117

6.2 Contribuições . . . . .	118
6.3 Atividades Futuras de Pesquisa . . . . .	118
<b>Referências . . . . .</b>	<b>120</b>
<b>Apêndices . . . . .</b>	<b>123</b>
<b>APÊNDICE A Dicionários . . . . .</b>	<b>124</b>
A.1 Produção da Seda . . . . .	125
A.2 Processamento de Imagens & Visão Computacional . . . . .	126
<b>Anexos . . . . .</b>	<b>128</b>
<b>ANEXO A Informações de Produção . . . . .</b>	<b>129</b>

# 1 Introdução

## 1.1 Contexto do Trabalho

Atualmente, a empresa Fiação de Seda BRATAC (situada em Londrina/PR) é cliente do SENAI CIMATEC, possuindo um projeto em execução na área de Microeletrônica & Sistemas Embarcados (FIEP, 2013).

Durante o período de execução do projeto, as pessoas envolvidas puderam se familiarizar com o processo de produção da seda, o que permitiu a identificação de outras oportunidades de trabalhos em conjunto, como a que motivou a realização deste trabalho.

Conforme será detalhado no Capítulo 2, ao longo da cadeia de produção da seda existem 3 situações onde é feito o controle de qualidade dos casulos (BRATAC, 2002). São elas:

### **Entrepastos**

O controle de qualidade é efetuado antes da comercialização dos casulos, numa transação que envolve o agricultor e a indústria de fiação (produtor de casulos e cliente, respectivamente).

### **Preparação para a Produção**

Já na indústria de fiação, após a retirada do armazém e antes de serem conduzidos à linha de produção, os casulos passam novamente por um controle de qualidade.

### **Linha de Produção**

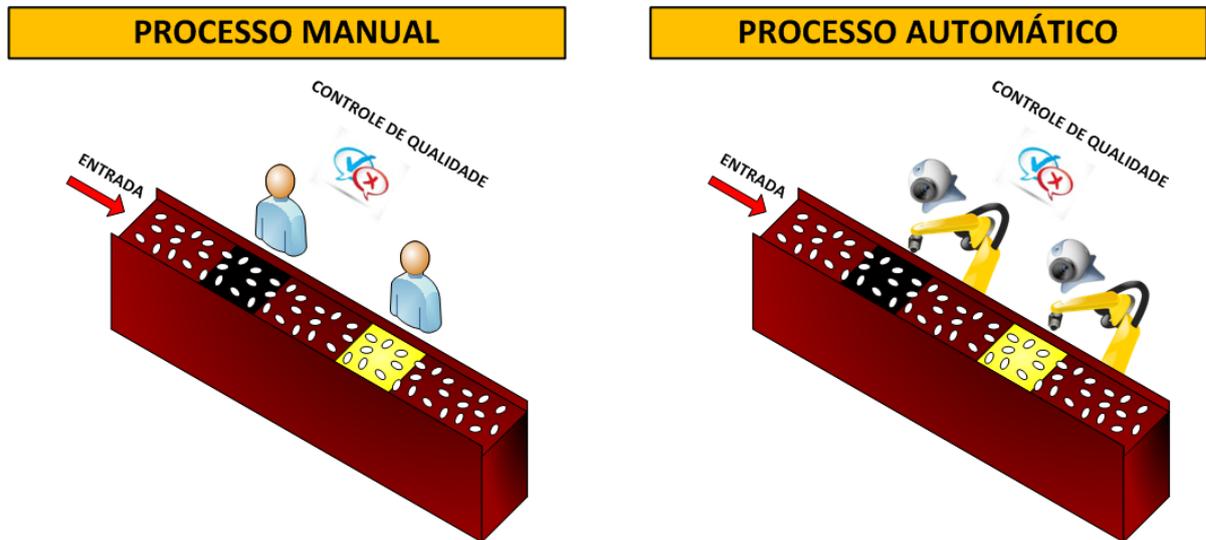
O controle de qualidade é efetuado em diversos pontos ao longo da linha de produção.

Após observar a maneira que este controle de qualidade é feito atualmente, identificou-se a possibilidade de *automatizar o processo de detecção, classificação e retirada dos casulos que não atendam aos critérios de qualidade (e.g. tamanho, forma, cor e textura dentro de um padrão pré-estabelecido)*.

Para tal, pode-se pensar no desenvolvimento de uma solução que utilize, por exemplo, visão computacional (para localizar os casulos), inteligência artificial (para classificá-los) e sistemas mecatrônicos (para remover os casulos defeituosos).

Para melhor ilustrar a diferença entre o processo de controle de qualidade dos casulos ocorrendo de forma manual e automática, uma comparação entre os métodos é feita na Figura 1:

Figura 1 – Controle de qualidade dos casulos em uma linha de produção de fios de seda



Fonte: Autoria própria

## 1.2 Definição do Problema

Diante do contexto apresentado, é importante destacar a complexidade do projeto como um todo (envolvendo áreas diversas como Sistemas Embarcados, Robótica, Automação, Processamento de Imagens, Visão Computacional e Desenvolvimento de Produtos).

Considerando, ainda, a limitação do tempo de execução do projeto por parte do discente, o escopo do trabalho restringiu-se à parte do projeto ligada ao processamento de imagens e visão computacional, mais especificamente à etapa de segmentação das imagens.

Apresenta-se então, enquanto problema norteador deste trabalho, a seguinte questão:

*“Quais os métodos de segmentação de imagens mais adequados à detecção de casulos do bicho-da-seda?”*

### 1.3 Objetivo

O objetivo geral deste trabalho é *analisar os diferentes métodos de segmentação de imagens existentes, com foco na detecção de casulos do bicho-da-seda.*

De forma mais detalhada e partindo do objetivo geral, pode-se definir os objetivos específicos como sendo:

- Criar um banco de imagens dos casulos do bicho-da-seda;
- Analisar, de maneira sistemática, as características do problema;
- Comparar os diferentes métodos de segmentação de imagens (com base em suas principais características);
- Identificar os métodos de segmentação de imagens mais adequados à resolução do problema;
- Implementar os algoritmos identificados e comparar seus respectivos desempenhos.

### 1.4 Motivação

Para o discente, a principal motivação foi ter um primeiro contato com o campo de estudos de processamento de imagens e visão computacional, sempre com foco em aplicações industriais. Apesar de serem consideradas ferramentas extremamente versáteis e poderosas para sistemas de automação, as áreas citadas não foram abordadas durante a graduação ou pós-graduação do discente. Sendo assim, objetiva-se corrigir esta lacuna de formação.

Do ponto de vista do SENAI CIMATEC, a finalidade do trabalho é:<sup>1</sup>

- Analisar a viabilidade de execução do projeto;
- Fidelizar o cliente (através da execução de outro projeto em conjunto);
- Aumentar o portfólio de projetos (sobretudo aqueles na área de processamento de imagens e visão computacional).

Por último, do ponto de vista da empresa, os principais interesses serão elencados na seção posterior (Seção 1.5).

---

<sup>1</sup>Os dois últimos itens derivam do primeiro, caso a viabilidade do projeto seja comprovada.

## 1.5 Importância da Pesquisa

Conforme citado anteriormente, este trabalho constitui o primeiro passo na direção de um projeto maior, que encontra-se atualmente em estágio embrionário. Em outras palavras, os resultados obtidos servirão de base para o desenvolvimento de um sistema de automação do processo de detecção, classificação e retirada dos casulos que não atendam aos critérios de qualidade.

Uma vez que o referido sistema de automação esteja pronto, sua implantação na indústria trará uma série de benefícios à empresa, dentre os quais pode-se destacar (SILVEIRA, 2014):

- Aumento da produtividade;
- Redução de custos;
- Registro das informações de produção (e, conseqüentemente, maior controle sobre o processo);
- Melhoria da qualidade (através da padronização e repetibilidade)<sup>2</sup>;
- Redução dos problemas de ergonomia dos trabalhadores (cansaço visual, lesões na coluna, LER).

---

<sup>2</sup>Considerando o ótimo desempenho alcançado por seres humanos em tarefas gerais de inspeção visual, falar em melhoria da qualidade advinda de sistemas de automação baseados em visão computacional parece ser contra o senso comum.

Como colocado em (DEMANT; STREICHER-ABEL; GARNICA, 2013), o problema é que a comparação entre pessoas e máquinas normalmente é feita considerando apenas o desempenho máximo das pessoas. Entretanto, é fato que ninguém é capaz de manter o mesmo desempenho ao longo de um dia inteiro de trabalho, devido tanto a questões físicas (cansaço físico e visual) quanto a questões psicológicas (problemas de desmotivação e concentração, normalmente causados pela monotonia do trabalho). Sendo assim, os sistemas automatizados de inspeção visual são capazes de igualar e, em alguns casos, até mesmo ultrapassar o desempenho médio de um operador.

Finalmente, num sistema deste tipo as imagens utilizadas podem ser geradas através de sensores das mais variadas espécies (infravermelho, ultravioleta, raios X, ultrassônico), capazes de obter informações invisíveis ao olho humano.

## 1.6 Limites e Limitações

Nesta seção são apresentados os **limites** e **limitações** definidos antes e durante o desenvolvimento da pesquisa. Os **limites** são aspectos que estabelecem o escopo da pesquisa e as **limitações** são problemas enfrentados durante a pesquisa que levam a tomadas de decisão que podem, inclusive, mudar o direcionamento da pesquisa.

Entre os **limites** da pesquisa, pode-se destacar:

### **Prova de Conceito x Desenvolvimento de Produto**

É preciso deixar claro que *não é objetivo deste trabalho desenvolver um produto*. Em primeiro lugar, acredita-se que o desenvolvimento de um sistema dessa natureza deve ser feito de maneira integrada, visto que as decisões relativas a uma parte do projeto podem influenciar em outras partes do mesmo. Somando-se a isso alguns fatores como nível de imaturidade atual do projeto, dificuldade de acesso à informações junto a empresa e a limitação de tempo por parte do discente, optou-se pela realização de uma prova de conceito.

### **Intervenções no Processo**

O estudo foi feito considerando que, em princípio, não haveria nenhuma intervenção no processo. Esse tipo de restrição limita as alternativas de projeto, inviabilizando ou dificultando algumas soluções.

### **Utilização de Imagens Estáticas**

Os algoritmos foram elaborados e testados com base apenas em imagens estáticas. Apesar disso não invalidar os resultados encontrados, é importante salientar que em uma aplicação industrial essa abordagem pode ser suficiente (controle de qualidade nos entrepostos) ou pode ser necessário fazer adaptações para que o algoritmo trabalhe em tempo real (controle de qualidade na linha de produção).

### **Utilização do MATLAB**

Os algoritmos foram elaborados e testados em MATLAB. Deste modo, para utilizar os algoritmos desenvolvidos em um sistema industrial, será necessário avaliar qual a melhor solução global e, caso necessário, traduzir o código para uma outra linguagem. No caso da escolha por um sistema embarcado dedicado, por exemplo, será necessário utilizar uma linguagem compatível (e.g. C, C++, Python etc).

Já entre as **limitações** da pesquisa, tem-se:

### Redução do Escopo

Inicialmente, o escopo do trabalho abrangia as etapas de segmentação e classificação das imagens. Entretanto, devido à complexidade encontrada no problema de segmentação, o trabalho foi dividido em duas partes. Sendo assim, este trabalho limitou-se à etapa de segmentação de imagens e um outro trabalho, feito em paralelo por outro discente, analisou o problema de classificação.

### Métodos de Segmentação

Conforme será detalhado e justificado no Capítulo 4, optou-se por implementar 3 métodos de segmentação dentre os diversos existentes (sendo eles os métodos *Threshold*, *Watershed* e Transformada de Hough).

### Configuração dos Parâmetros

Para que os algoritmos desenvolvidos funcionem adequadamente, é necessário configurar seus parâmetros para uma determinada situação (i.e. nível de iluminação, distância dos objetos etc.). Em princípio, este tipo de premissa pode parecer deveras restritivo para uma aplicação industrial, mas (DEMANT; STREICHER-ABEL; GARNICA, 2013) faz uma analogia bastante interessante sobre esta questão: se ninguém usa uma furadeira equipada com uma broca de furar madeira para trabalhar com concreto armado, será que é razoável esperar que um algoritmo sempre funcione, sem que seus parâmetros tenham que ser reconfigurados?

## 1.7 Hipóteses

No campo de processamento de imagens e visão computacional, costuma-se dizer que cada problema é único, possuindo particularidades que devem ser levadas em conta no processo de busca por uma solução. Considerando, ainda, que existe uma miríade de combinações entre métodos de segmentação e etapas de pré-processamento (que podem conduzir a resultados completamente diferentes), torna-se uma tarefa não trivial determinar quais métodos de segmentação de imagem podem ser utilizados na detecção de casulos do bicho-da-seda.

Abaixo, segue algumas hipóteses que serão testadas neste trabalho:

1. Os métodos clássicos de segmentação de imagem podem ser utilizados na detecção de casulos do bicho-da-seda.
2. O desempenho dos métodos implementados é suficiente para utilização em uma aplicação industrial.

## 1.8 Aspectos Metodológicos

Esta seção apresenta, de maneira global, a metodologia utilizada no presente estudo. Para facilitar a compreensão, os detalhes da execução de cada uma dessas etapas serão fornecidos apenas no Capítulo 5, antes da apresentação dos resultados.

1. **Problema Norteador** - Quais os métodos de segmentação de imagem mais adequados à detecção de casulos do bicho-da-seda?
2. **Aquisição de Dados** - Obter um conjunto de imagens dos casulos do bicho-da-seda, que seja representativo, para utilização nos algoritmos;
3. **Análise das Características do Problema** - Levantar e analisar as características específicas do problema, que podem influenciar na escolha de uma solução;
4. **Análise das Características das Técnicas de Segmentação de Imagens** - Comparar as principais técnicas de segmentação de imagens existentes na literatura;
5. **Escolha de 3 Técnicas de Segmentação de Imagens** - Cruzar as informações obtidas nos itens 3) e 4) para determinar as técnicas de segmentação de imagem a serem implementadas;
6. **Implementação dos Algoritmos** - Utilizar o MATLAB e suas ferramentas (*Toolbox* de Processamento de Imagens e *Toolbox* de Visão Computacional, APIs, etc) para agilizar o desenvolvimento dos algoritmos;
7. **Organização dos Dados** - Separar as imagens obtidas (amostras) em grupos, de acordo com as necessidades de cada um dos testes;
8. **Análise dos Resultados** - Fazer uma análise comparativa dos resultados obtidos para cada método (com base nos respectivos desempenhos).

## 1.9 Organização da Monografia

Este documento é composto por 6 capítulos e está estruturado da seguinte forma:

- **Capítulo 1 - Introdução:** Contextualiza o âmbito no qual a pesquisa proposta está inserida. Explicita, portanto, a definição do problema, os objetivos e justificativas da pesquisa e a maneira como esta monografia está estruturada;
- **Capítulo 2 - Produção da Seda:** Apresenta as principais informações obtidas ao longo da revisão bibliográfica do processo de produção da seda. Deste modo, fornece ao leitor uma visão geral do contexto global em que o trabalho está inserido;
- **Capítulo 3 - Processamento de Imagens & Visão Computacional:** Aborda os conceitos básicos das áreas de Processamento de Imagens & Visão Computacional. Visa nivelar o conhecimento do leitor e estabelecer uma linguagem comum com o mesmo;
- **Capítulo 4 - Segmentação de Imagens:** Faz uma análise comparativa dos diferentes métodos de Segmentação de Imagens com base em suas principais características. Tem como objetivo identificar os métodos de segmentação de imagens mais adequados à resolução do problema;
- **Capítulo 5 - Desenvolvimento da Pesquisa:** Expõe o desenvolvimento do trabalho e a análise dos resultados encontrados;
- **Capítulo 6 - Considerações Finais:** Apresenta as conclusões, contribuições do trabalho e algumas sugestões de atividades de pesquisa a serem desenvolvidas no futuro.

## 2 Produção da Seda

### 2.1 Introdução

Este capítulo visa apresentar, de maneira concisa e simplificada, algumas informações a respeito da produção da seda. Nele foram compilados os principais conhecimentos obtidos ao longo da revisão bibliográfica do tema, de modo a fornecer ao leitor uma visão geral do contexto global em que o trabalho está inserido.

Primeiramente, apresenta-se informações gerais da seda, como suas características físicas, importância e principais usos. Em seguida, analisa-se o processo de produção da seda, que começa com a criação das mariposas, passa pela produção do fio de seda, pela manufatura da fibra têxtil da seda e termina com a fabricação dos tecidos de seda. Por último, algumas características do processo de controle de qualidade dos casulos são evidenciadas.

### 2.2 A Seda

A seda é um filamento de origem animal, secretado sobretudo por algumas espécies de insetos e artrópodes para construção de seus casulos e/ou teias ([Encyclopedia Britannica, 2014b](#)).

Apesar da seda produzida pela maioria dos animais não ser prática para a confecção de tecidos, as larvas de algumas espécies de mariposas produzem casulos que podem ser utilizados de maneira economicamente viável para a manufatura de fibras têxteis. O tipo de seda mais comum é obtido a partir do casulo da espécie *Bombyx mori* (lagarta da amoreira), que foi domesticada pelo homem e recebeu a alcunha de bicho-da-seda ([The Textile Museum, 1997](#)).

O processo de criação do bicho-da-seda voltado para a produção da seda é conhecido como **sericultura**. Evidências históricas apontam que a seda surgiu na China<sup>1</sup> e seu processo de produção, por ser altamente sigiloso, foi sendo lentamente difundido ao redor do mundo, sendo introduzido na Europa somente em 553 d.C. ([International Sericultural Commission, 2013](#)).

A seda possui uma série de virtudes com as quais nenhum outro tecido até hoje, seja ele natural ou artificial, pode competir. Essas particularidades lhe renderam o merecido título de “Rainha dos Tecidos”. Qualidades diferenciadas como brilho natural, afinidade inerente para tingimentos, cores vibrantes, alta absorção, baixa densidade e excelente panejamento fizeram da seda um tecido altamente desejável desde a sua descoberta ([Texere Silk, 2012](#)).

---

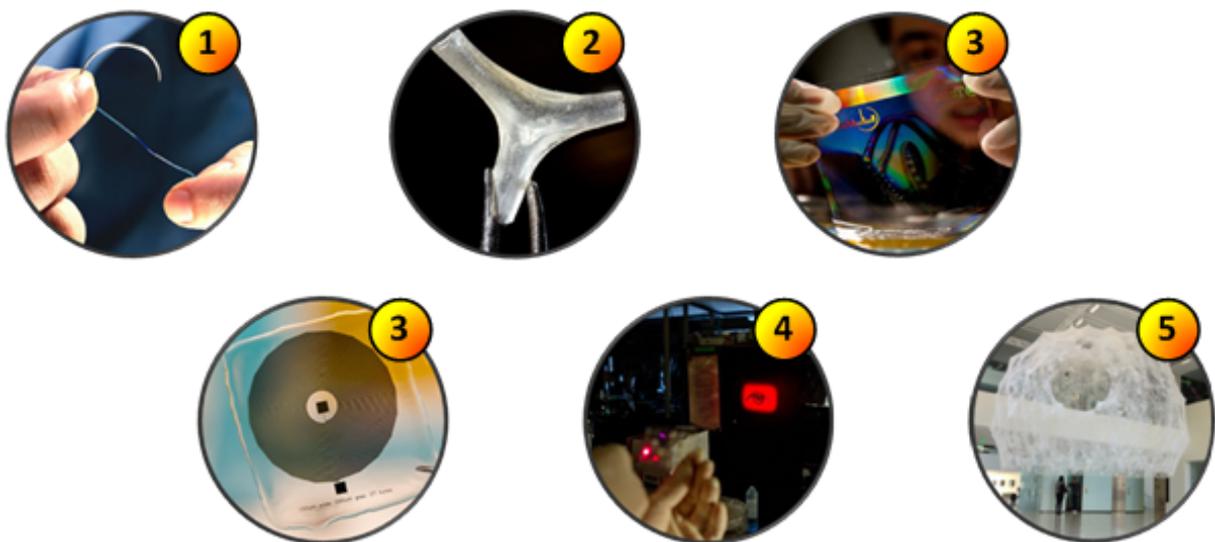
<sup>1</sup>Ainda existem controvérsias a respeito do período desta descoberta: de acordo com os textos de Confúcio, a descoberta da seda ocorreu por volta de 2700 a.C., mas registros arqueológicos apontam para o período Yangshao (5000 a.C 3000 a.C.) ([BARBER, 1992](#)).

Nos últimos anos, a seda tem deixado de ser apenas matéria prima para tecidos de luxo e vem encontrando aplicações em diversas áreas. Isso se deve, sobretudo, a duas características importantíssimas omitidas até o momento: sua **biocompatibilidade** e sua **alta resistência à tração**.

Dentre as novas aplicações, pode-se destacar (Figura 2):

1. **Suturas Cirúrgicas** - A resistência à tração da seda e sua composição orgânica a tornam ideal para a composição de fios de sutura não-absorvíveis pelo corpo humano;
2. **Aplicações Ortopédicas** - Por não disparar nenhuma reação do sistema imunológico, as fibras de seda podem ser utilizadas como estrutura de reparação dos músculos, ossos, cartilagens e tendões;
3. **Aplicações Ópticas, Eletrônicas e Fotônicas;**
4. **Composição de Metamateriais;**
5. **Pesquisa de Materiais Bioinspirados.**

Figura 2 – Exemplos de novas aplicações da seda



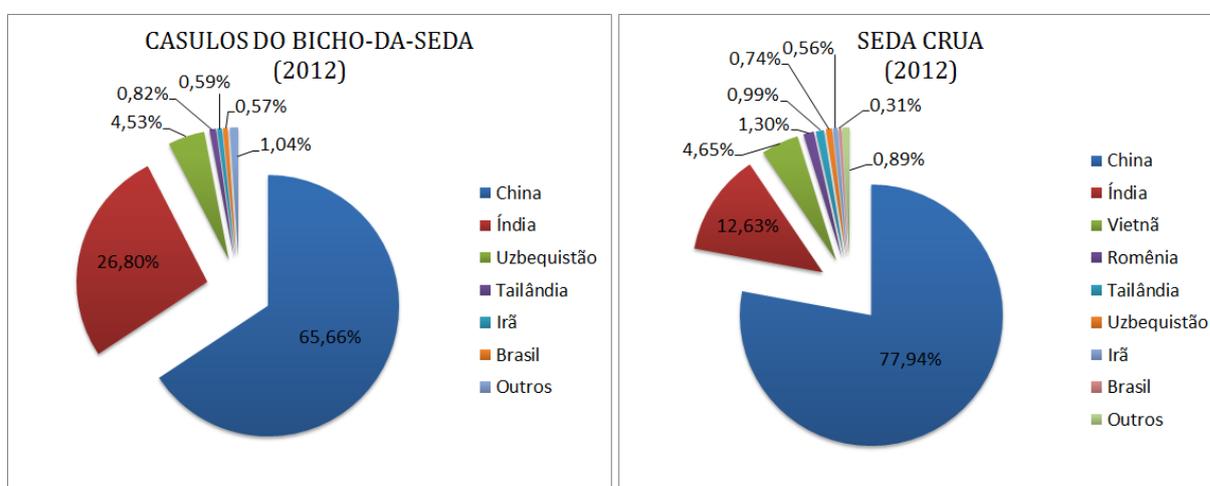
Fonte: Adaptado de (TIMES, 2011) e (BOADO, 2012)

### 2.2.1 Contexto Econômico Mundial

A seda é um *commodity* agrícola de alto valor, mas de baixa produção, responsável por apenas 0,2% da produção têxtil do mundo. Por ser uma atividade intensiva em mão-de-obra e de alta geração de renda, a produção da seda é considerada uma ferramenta importante para o desenvolvimento econômico de um país. Deste modo, muitos países em desenvolvimento a utilizam para gerar empregos (sobretudo no setor rural) e, ao mesmo tempo, obter moeda estrangeira (INDIA, 2013).

De acordo com os dados de 2012 da FAO (*Food and Agriculture Organization of the United Nations*), a produção da seda no mundo se divide da seguinte maneira (Figura 3):

Figura 3 – Principais produtores mundiais de seda



Fonte: Adaptado de (FAO, 2012)

É interessante observar que, geograficamente, a Ásia é a região que mais produz seda no mundo, sendo responsável por mais de 95% da produção mundial. Além disso, embora existam mais de 20 países no mapa da seda, o grosso da produção ocorre na China e na Índia (sendo complementada por países como Uzbequistão, Tailândia, Irã, Japão, Brasil e Coreia).

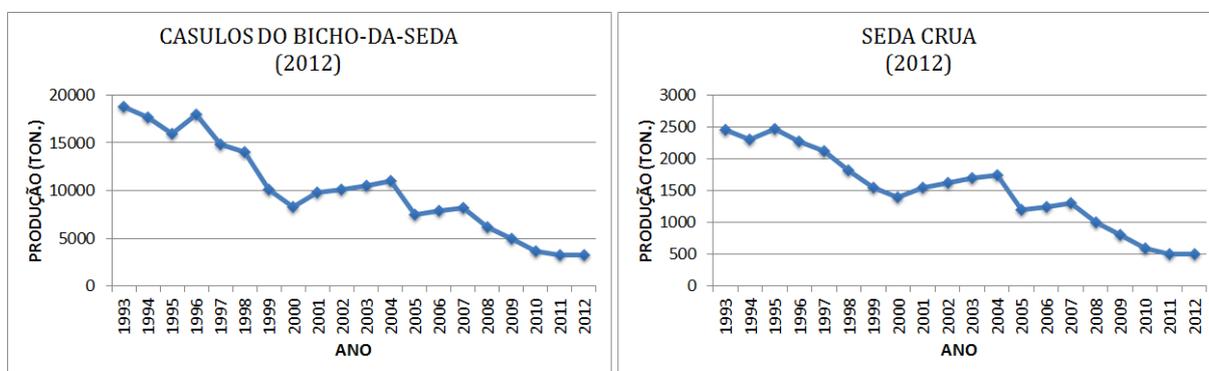
## 2.2.2 Contexto Econômico Nacional

No Brasil, a produção de casulos do bicho-da-seda é feita quase que exclusivamente no Paraná, envolvendo em torno de 4 mil famílias e sendo responsável por cerca de 92% da produção nacional. No Noroeste do estado localiza-se o Vale da Seda, maior polo de produção de casulos do Ocidente. Um clima propício à criação do bicho-da-seda, a dedicação dos criadores e o investimento em melhoramento genético das lagartas ao longo dos anos fez com que o fio de seda produzido no Vale da Seda fosse considerado o melhor do mundo ([Jornal de Londrina, 2013](#)).

Culturalmente, a produção nacional sempre foi voltada à exportação do fio de seda (cerca de 90% vai para o exterior, principalmente para o Japão). Entretanto, o paradigma atual é formar uma cadeia produtiva, capaz de transformar a seda crua em um produto final de maior valor agregado (1 kg de casulo de primeira qualidade transformado em produto final pode valer cerca de 30 vezes mais) ([Jornal de Londrina, 2013](#)).

Visando reverter o declínio da sericultura no país (Figura 4), esta mudança de paradigma tornou-se uma questão de sobrevivência. A concorrência da China e a migração dos produtores para outras atividades causou a diminuição da produção e do número de criadores ([Jornal de Londrina, 2013](#)).

Figura 4 – Evolução da produção de seda no Brasil ao longo dos anos



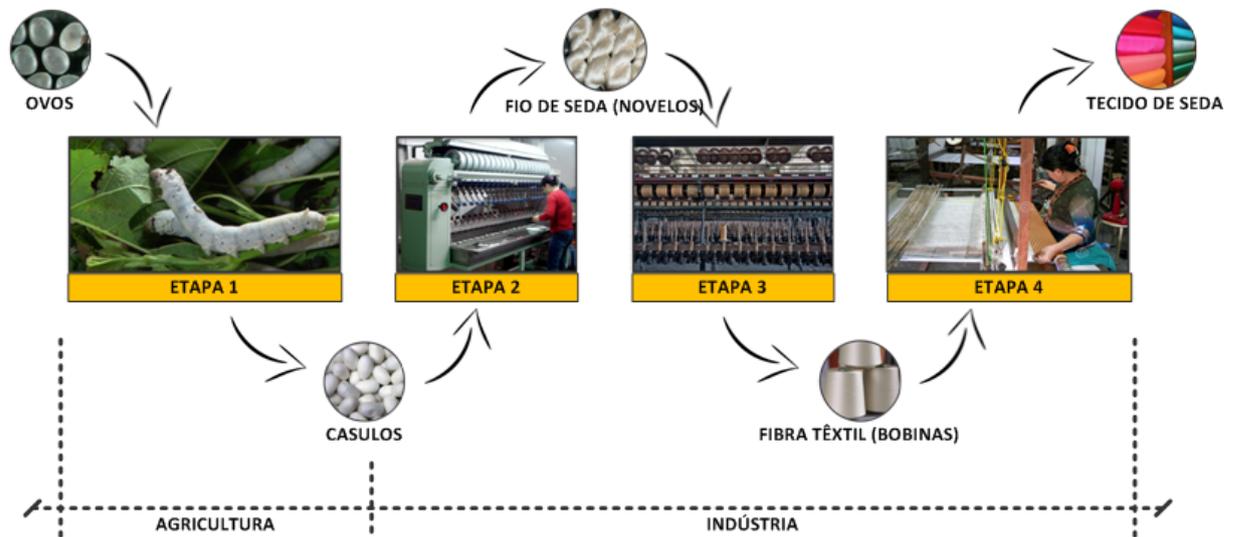
Fonte: Adaptado de ([FAO, 2012](#))

Das 17 fiações industriais existentes no país em 1993, restou apenas a Bratac, de Londrina. Com poucas exceções, hoje a cadeia produtiva do setor começa e termina nessa empresa. É ela quem fornece as larvas de bicho-da-seda e a assistência técnica aos produtores, e mais tarde compra e beneficia os casulos ([Jornal de Londrina, 2013](#)).

## 2.3 Processo de Produção da Seda

Como mostra a Figura 5 abaixo, o processo de produção da seda envolve 4 etapas:

Figura 5 – Processo de produção da seda e suas respectivas entradas e saídas



Fonte: Adaptado de (TESINOVA, 2011)

### Etapa 1 - Sericultura

Etapa de criação do bicho-da-seda.

### Etapa 2 - Fiação

Etapa de produção de novelos do fio de seda (i.e. seda crua).

### Etapa 3 - Manufatura da Fibra Têxtil

Etapa de trançagem do fio de seda para produção da fibra têxtil.

### Etapa 4 - Produção dos Tecidos

Etapa de produção de tecidos a partir da fibra têxtil da seda.

É interessante observar que, por serem atividades essencialmente distintas, normalmente cada uma dessas etapas é realizada por uma empresa diferente (a exceção geralmente ocorre com as etapas 2 e 3).

Cada etapa, por sua vez, é composta por uma série de passos, que serão apresentados sucintamente nos tópicos seguintes.

### 2.3.1 Etapa 1: Processo de Produção dos Casulos do Bicho-da-Seda

O processo de produção dos casulos do bicho-da-seda normalmente envolve 3 atividades ([Encyclopedia Britannica, 2014a](#)) e ([Jornal de Londrina, 2013](#)):

#### Criação do Bicho-da-Seda

Desde a eclosão do ovo até a formação completa do casulo.

#### Cultivo das Amoreiras

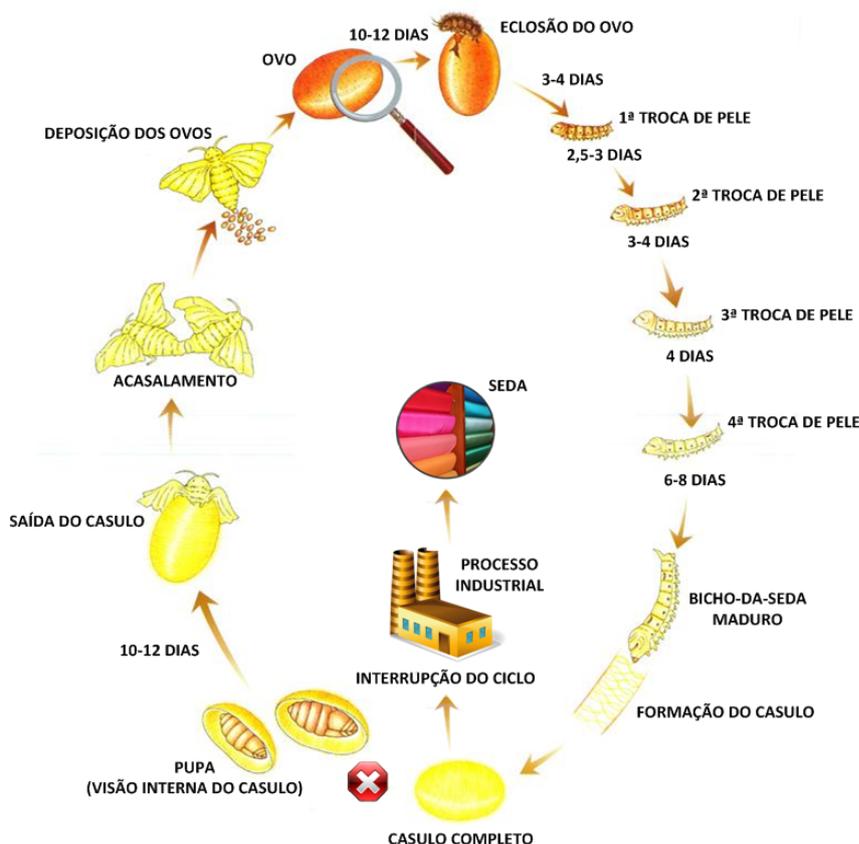
Que fornecem as folhas das quais as larvas do bicho-da-seda se alimentam.

#### Produção dos Ovos do Bicho-da-Seda

Considerando aspectos como a preservação e o melhoramento genético das raças.

No contexto deste trabalho, apenas a atividade de criação do bicho-da-seda é relevante. Deste modo, o ciclo de vida da espécie *Bombyx mori* será analisado logo abaixo (Figura 6)<sup>2</sup>:

Figura 6 – Ciclo de vida do Bicho-da-Seda (*Bombyx mori*)



Fonte: Adaptado de ([SRIDHAR, 2010](#))

<sup>2</sup>Um conjunto de fotos em alta definição de todas as etapas do ciclo de vida do bicho-da-seda está disponível em ([SRIDHAR, 2010](#)).

Para tornar a compreensão do ciclo de vida do bicho-da-seda mais intuitiva, é interessante começar a análise a partir dos ovos fecundados do bicho-da-seda.

Após a eclosão dos ovos, a larva passa por 4 trocas de pele (i.e. exoesqueleto quitinoso) até atingir a maturidade (5ª idade).

Ao final desta idade, a larva tece um casulo de seda, utilizando um único filamento contínuo de seda (que pode ter de 500 a 1500 metros de comprimento). Este filamento é constituído principalmente pelas proteínas **fibroína** (76 %) e **sericina** (22 %), que são produzidas por glândulas especiais, expelidas pela boca e se solidificam imediatamente ao contato com o ar.<sup>3</sup>

A conclusão do casulo corresponde a um marco de decisão, onde o ciclo de vida do bicho-da-seda pode prosseguir normalmente ou ser interrompido pelo homem para produção da seda.

Quando o ciclo é interrompido pelo homem, o processo segue as etapas 2 e 3, que serão detalhadas no tópico seguinte (tópico 2.3.2).

Caso não haja interrupção do ciclo, o ciclo natural prossegue normalmente: após a construção do casulo, o corpo da lagarta transforma-se em seu interior, formando a *pupa* ou *crisálida*. O processo de transformação continua até a formação do indivíduo adulto (i.e. a mariposa).

Finalmente, para conseguir sair, a mariposa libera um líquido alcalino que corrói uma das extremidades do casulo. Neste estágio, o inseto não se alimenta e se dedica apenas à reprodução da espécie. Depois do acasalamento a fêmea põe de 200 a 500 ovos, e, tal como o macho, morre (Centro Ciência Viva de Bragança, 2014).

---

<sup>3</sup>A **fibroína** corresponde ao fio de seda propriamente dito, enquanto a **sericina** é uma espécie de goma ou cola que reveste os fios de seda, mantendo-os unidos e transformando o casulo em uma estrutura uniforme e hermeticamente fechada.

### 2.3.2 Etapas 2 e 3: Processo de Produção da Fibra Têxtil da Seda

Conforme ilustrado na Figura 5, o processo de produção da fibra têxtil da seda a partir do casulo do bicho-da-seda envolve basicamente duas etapas: a **fiação** e a **manufatura da fibra têxtil**.

Por serem semelhantes, normalmente essas etapas são executadas em uma mesma indústria e, deste modo, serão descritas aqui num único tópico. Para facilitar o entendimento do processo, o diagrama abaixo (Figura 7) esquematiza seus principais passos:

Figura 7 – Processo de produção da fibra têxtil da seda



Fonte: Adaptado de (KUSNAMAN, 2004)

Uma descrição detalhada de cada passo do processo de produção da fibra têxtil da seda pode ser encontrada em (LEE, 1999). De maneira resumida e com base nesta referência, têm-se:

#### Passo 0 - Aquisição dos Casulos (*Cocoon Buying*)

O passo inicial do processo de produção da fibra têxtil da seda é a aquisição dos casulos. Como acontece com qualquer matéria prima, o preço pago pelas indústrias aos agricultores depende da qualidade dos casulos. Este controle de qualidade, por sua vez, normalmente considera parâmetros como *porcentagem de seda bruta*<sup>4</sup>, *comprimento do filamento*, *capacidade de enrolamento do filamento* e *porcentagem de casulos defeituosos*.

#### Passo 1 - Secagem dos Casulos (*Cocoon Drying*)

O principal objetivo deste passo é preservar a qualidade dos casulos, mantendo-os adequados para o processo de enrolamento e prevenindo danos causados por longos períodos de armazenamento.

O primeiro perigo é a metamorfose contínua da pupa, pois para sair do casulo a mariposa precisa furá-lo (tornando-o inútil para a conversão em seda crua).

Outro problema surge a partir do excesso de umidade na parte interna do casulo, que pode levá-lo à putrefação ou favorecer o surgimento de mofo.

Em outras palavras, a secagem mata a pupa e evapora a umidade, evitando danos aos casulos.

<sup>4</sup>A porcentagem de seda bruta é definida como sendo a relação entre a massa do casulo vazio e a massa do casulo contendo a pupa do bicho-da-seda.

**Passo 2 - Armazenamento dos Casulos (*Cocoon Storage*)**

Após o processo de secagem, os casulos devem ser armazenados em condições específicas de temperatura e humidade (de modo a evitar danos causados por fungos e pestes).

**Passo 3 - Classificação dos Casulos (*Cocoon Sorting*)**

Mesmo tendo passado por uma classificação preliminar, realizada antes da compra, alguns casulos defeituosos podem ter passado despercebidos e defeitos podem ter se desenvolvido em casulos que originalmente estavam bons.

Sendo assim, de modo a garantir uma matéria prima boa e uniforme, antes de serem conduzidos à linha de produção os casulos passam novamente por um controle de qualidade, que, de maneira semelhante ao primeiro, em geral envolve testes mecânicos e/ou de inspeção visual.

**Passo 4 - Cozimento dos Casulos (*Cocoon Cooking*)**

A sericina (uma das proteínas que compõem o casulo do bicho-da-seda) é uma espécie de goma ou cola que reveste os fios de seda, mantendo-os unidos e transformando o casulo em uma estrutura uniforme e hermeticamente fechada. Durante o processo de secagem dos casulos, este revestimento endurece e dificulta ainda mais a retirada dos fios de seda.

Deste modo, para desenrolar o fio de seda que forma o casulo (vide **Passo 5 - Enrolamento**), é necessário amolecer e/ou remover a sericina através de um processo conhecido como **cozimento**. No cozimento, os casulos são submetidos a processos baseados em calor, água e vapor durante um tempo pré-estabelecido (em indústrias mais simples, os casulos são apenas postos em água fervente).

**Passo 5 - Enrolamento ou Bobinamento (*Reeling*)**

Este processo consiste basicamente em desenrolar o filamento de seda que forma cada um dos casulos e enrolá-los juntos (normalmente de 4 a 8 filamentos), produzindo um único fio. A saída do processo normalmente ocorre na forma de novelos do fio de seda (i.e. seda crua).

**Passo 6 - Produção da Fibra Têxtil<sup>5</sup> (*Throwing*)**

Processo industrial onde o fio de seda que foi enrolado em novelos é *limpo, trançado e enrolado em bobinas*, formando a fibra têxtil. O objetivo deste processo é uniformizar e aumentar a resistência do material.

---

<sup>5</sup>Passo também conhecido como **Re-enrolamento** ou **Rebobinamento**.

### 2.3.3 Etapa 4: Processo de Produção do Tecido de Seda

É interessante observar que uma grande diversidade de tecidos diferentes pode ser confeccionada a partir da fibra têxtil da seda (e.g. seda pura, seda selvagem, *chiffon* de seda e cetim). Essa variedade de tecidos é obtida através da escolha do tipo de seda, processo produtivo e da mistura com outras fibras têxteis.

Ao contrário das demais etapas, o detalhamento do processo de produção dos tecidos feitos de seda foge ao escopo do projeto e, portanto, não será descrito neste trabalho.

## 2.4 Controle de Qualidade dos Casulos

Conforme apresentado no Tópico 2.3.2, ao longo da cadeia de produção da seda existem 3 situações onde é feito o controle de qualidade dos casulos (BRATAC, 2002). São elas:

### **Entrepósitos**

O controle de qualidade é efetuado antes da comercialização dos casulos, numa transação que envolve o agricultor e a indústria de fiação (produtor de casulos e cliente, respectivamente).

### **Preparação para a Produção**

Já na indústria de fiação, após a retirada do armazém e antes de serem conduzidos à linha de produção, os casulos passam novamente por um controle de qualidade.

### **Linha de Produção**

O controle de qualidade é efetuado em diversos pontos ao longo da linha de produção.

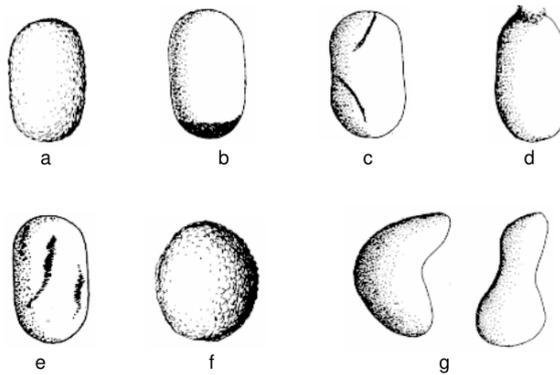
De acordo com Lee (1999), o mecanismo de classificação dos casulos varia de acordo com o grau de maturidade de cada país (do ponto de vista da sericultura). Países menos desenvolvidos realizam o controle de qualidade baseado apenas em inspeção visual e na experiência de seus funcionários, enquanto países modernos utilizam ainda uma série de testes sistemáticos.

Os testes normalmente utilizam análises estatísticas e, dentre os parâmetros de qualidade utilizados, talvez o mais importante seja a *porcentagem de casulos defeituosos*.

A título de ilustração, as Figuras 8 e 9 abaixo mostram alguns dos principais defeitos encontrados nos casulos do bicho-da-seda.

Figura 8 – Exemplos de defeitos comuns em casulos do bicho-da-seda.

(a) Casulo de Primeira (b) Manchado (c) Riscado (d) Furado (e) Sujo (f) Duplo (g) Irregular.



Fonte: Adaptado de (ZANETTI, 2013)

Figura 9 – Quadro de defeitos comuns em casulos do bicho-da-seda



Fonte: Fiação de Seda BRATAC S/A

# 3 Processamento de Imagens & Visão Computacional

## 3.1 Introdução

Visando nivelar o conhecimento do leitor e estabelecer uma linguagem comum com o mesmo, este capítulo aborda os conceitos básicos das áreas de Processamento de Imagens & Visão Computacional.

Inicialmente, analisa-se a relação existente entre as áreas de processamento de imagens, visão computacional e disciplinas correlatas. Em seguida, são apresentados alguns conceitos ligados às áreas e indispensáveis a uma melhor compreensão do trabalho. Por último, discute-se as principais características dos sistemas de visão computacional aplicados à indústria, incluindo informações referentes ao projeto.

## 3.2 Relação entre as Disciplinas

Existe uma série de disciplinas intimamente relacionadas ou até mesmo em parte coincidentes com a área de **processamento de imagens**, como a **visão computacional**, a **computação gráfica** ou ainda a **análise de imagens** (MAINTZ, 2005).

Apesar de não haver um consenso entre os especialistas a respeito das fronteiras entre essas disciplinas (GONZALEZ; WOODS, 2002), normalmente adota-se duas abordagens:

- Classificação de acordo com a entrada e a saída do processo (tópico 3.2.1);
- Classificação de acordo com o nível do processo (tópico 3.2.2).

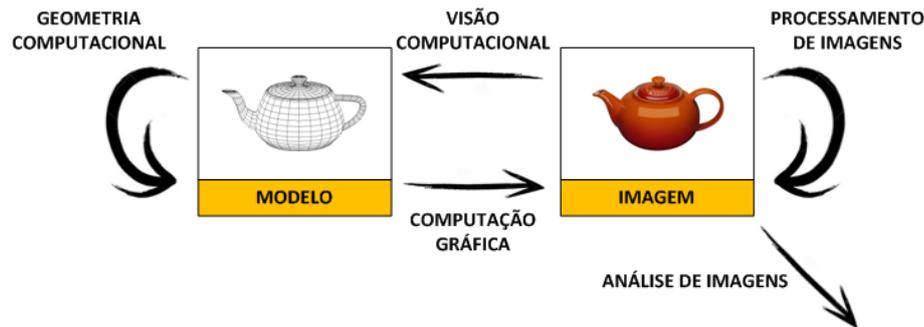
Em ambos os casos, considerando a sutileza das diferenças entre algumas das disciplinas, deduz-se que, na prática, a divisão utilizada nem sempre é clara, o que lhe fornece um caráter sobretudo didático. Numa aplicação do dia a dia, normalmente é irrelevante fazer esta classificação.

Sendo assim, com exceção do presente capítulo, ao longo deste trabalho serão utilizados de maneira intercambiável os termos **Processamento de Imagens & Visão Computacional**.

### 3.2.1 Classificação de acordo com a Entrada e a Saída do Processo

O diagrama da Figura 10 representa um modelo comumente adotado da relação existente entre a área de processamento de imagens e as demais disciplinas correlatas:

Figura 10 – Relação existente entre a área de processamento de imagens e disciplinas correlatas



Fonte: Adaptado de (MAINTZ, 2005)

No diagrama acima, as diferentes disciplinas são definidas de acordo com suas ações sobre as estruturas **imagem** e **modelo**.<sup>1</sup> Para entender essa relação, deve-se analisar o diagrama em termos de entrada e saída de cada processo.

De modo a esclarecer ainda mais a ideia, pode-se considerar um exemplo onde:

- A **imagem** é a fotografia de uma face;
- O **modelo** é um conjunto de pontos característicos da face (e.g. canto do olho, ponta do nariz);
- Os **parâmetros do modelo** são as distâncias entre esses pontos característicos.

Neste caso, aplicações típicas de cada disciplina seriam:

#### Processamento de Imagens

Gerar uma nova imagem a partir do aprimoramento da imagem original (e.g. aumento do contraste, remoção de olhos vermelhos).

#### Visão Computacional

Encontrar os pontos característicos da imagem e fazer a correspondência com o modelo (e.g. reconhecimento facial).

#### Computação Gráfica

Gerar a imagem de uma face a partir do modelo facial e de seus respectivos parâmetros.

<sup>1</sup>Um modelo matemático pode ser definido como *a representação de um sistema por meio dos conceitos e da linguagem da matemática*.

### 3.2.2 Classificação de acordo com o Nível do Processo

De acordo com o paradigma apresentado no tópico anterior, que considera como processamento de imagens todo processo cuja entrada é uma imagem e a saída também, tarefas extremamente triviais, tal como calcular a intensidade média de uma imagem (cuja resposta é apenas um número), não são consideradas operações de processamento de imagens.

Por este motivo, alguns autores consideram essa classificação limitante e artificial, adotando uma abordagem diferente para caracterizar os processos pertencentes ao espaço contínuo entre o processamento de imagens e a visão computacional.

De acordo com [Gonzalez e Woods \(2002\)](#) e [Davies \(2012\)](#), por exemplo, essa classificação deve ser feita de acordo com o nível do processo em análise (i.e. baixo, médio ou alto nível):

#### **Processo de Baixo Nível**

Processo cuja entrada é uma imagem e a saída também.

Envolve operações primitivas como redução de ruído, aumento de contraste ou aprimoramento de imagens.

#### **Processo de Médio Nível**

Processo cuja entrada é uma imagem e a saída um atributo extraído dessa imagem (e.g. bordas, contornos, objetos identificados). Em outras palavras, transforma-se um conjunto de dados “brutos” em uma informação mais relevante à resolução do problema.

Envolve tarefas como segmentação (dividir uma imagem em diferentes regiões ou objetos) e reconhecimento de objetos.

#### **Processo de Alto Nível**

Processo cuja entrada é um atributo extraído de uma imagem (i.e. informação relevante) e a saída uma informação final que o problema almeja encontrar.

Envolve obter informações dos objetos reconhecidos (exatamente como na análise de imagens) e realizar as demais funções cognitivas normalmente associadas com o processo de visão.

### 3.3 Processamento de Imagens

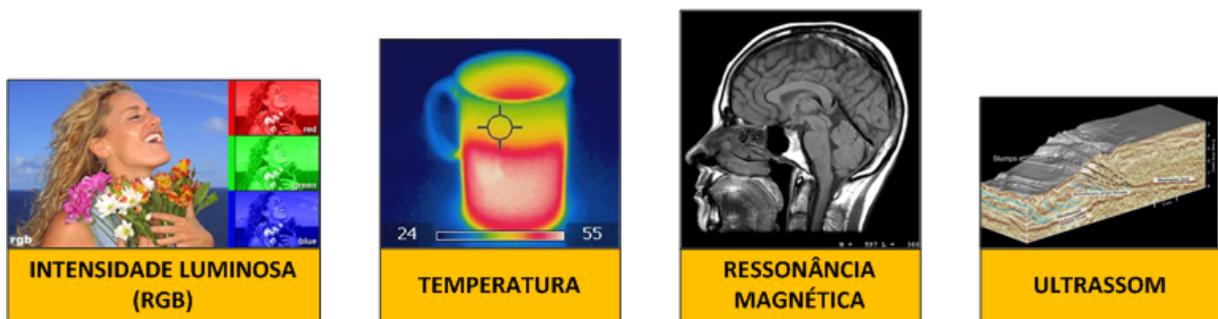
Conforme ilustrado na Figura 10, entende-se como processamento de imagens *todo processo cuja entrada é uma imagem e a saída uma versão alterada desta imagem* (normalmente modificada por meio de um conjunto de técnicas computacionais) (SHIH, 2010).

Uma imagem, por sua vez, é a *representação gráfica de alguma informação em termos espaciais* (FISHER et al., 2005).

A natureza da imagem depende do dispositivo físico utilizado (e.g. câmeras fotográficas e de vídeo, radares, equipamentos de raio-X, de ressonância magnética e de ultrassom), que deve ser selecionado de acordo com o tipo de informação (e.g. luminosidade, temperatura) e quantidade de dimensões (de 1D, 2D, 3D ou multidimensional) que se deseja (SHIH, 2010).

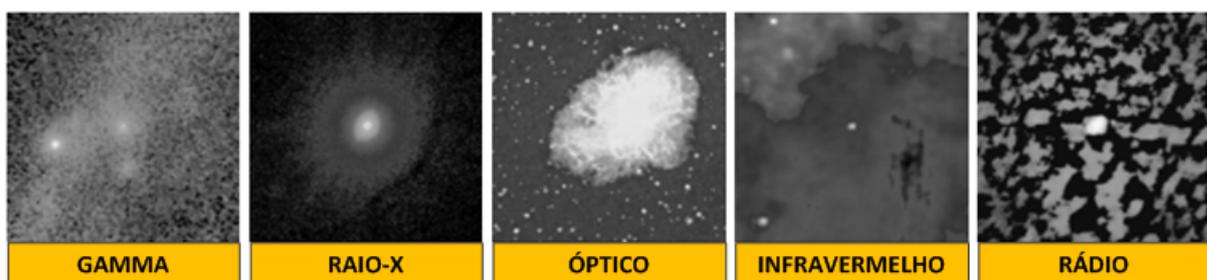
As Figuras 11 e 12 abaixo mostram exemplos de imagens obtidas a partir de diferentes tipos de sensores:

Figura 11 – Imagens obtidas a partir de diferentes tipos de sensores



Fonte: Autoria própria

Figura 12 – Imagens do Pulsar de Caranguejo obtidas em diferentes regiões do espectro eletromagnético



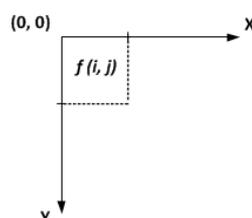
Fonte: Adaptado de (GONZALEZ; WOODS, 2002)

### 3.3.1 Processamento de Imagens Digitais

Ao falar de processamento de imagens, normalmente está implícito tratar-se de **imagens digitais**<sup>2</sup>.

Uma imagem digital nada mais é do que uma matriz de números, onde cada elemento  $(i, j)$  da matriz é conhecido como **pixel** (do inglês *picture element*). Por convenção, o ponto de referência espacial  $(0, 0)$  localiza-se no canto superior esquerdo da imagem, conforme indicado na Figura 13.

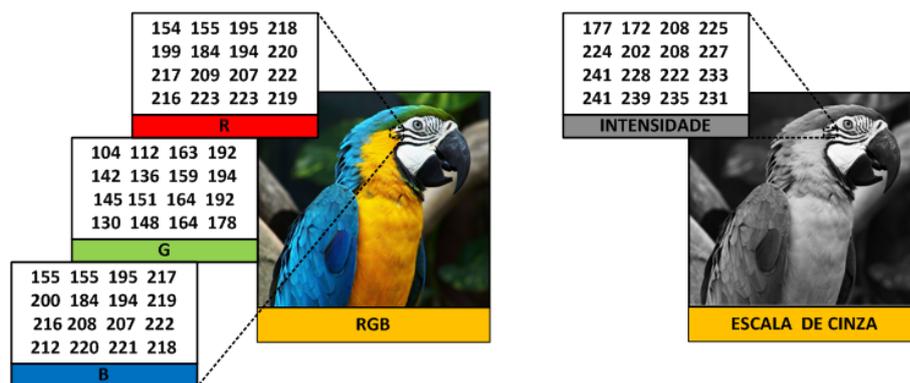
Figura 13 – Convenção utilizada na representação espacial de uma imagem



Fonte: Adaptado de (HORNBERG, 2006)

Cada pixel, por sua vez, é representado por uma  $n$ -upla de números inteiros (normalmente entre 0 e 255), que varia de acordo com o **espaço de cores** utilizado<sup>3</sup>. Na representação de uma imagem no espaço de cores RGB, por exemplo, cada pixel é representado por 3 números inteiros. Por outro lado, para representar esta mesma imagem no espaço de cores de escala de cinza, é necessário apenas 1 número inteiro por pixel (Figura 14).

Figura 14 – Imagem de uma arara-canindé representada em 2 espaços de cores diferentes



Fonte: Autoria própria

<sup>2</sup>Apesar de o processamento de imagens analógicas existir, na prática sua aplicação é bastante restrita, devido à complexidade do *hardware* envolvido, dos custos elevados e da falta de flexibilidade e modularidade dos equipamentos.

<sup>3</sup>Um **espaço de cores** é uma notação utilizada para especificar, de maneira padronizada e unívoca, as diferentes cores perceptíveis pelo sistema visual humano (TKALCIC; TASIC, 2003).

De acordo com [Tkalcic e Tasic \(2003\)](#), os espaços de cores podem ser classificados em 3 tipos:

### Espaço de Cores Baseados em HSV

Abrange o espaço de cores RGB, os espaços de cores baseados na teoria das cores opostas e os espaços de cores fenomenológicos.

### Espaço de Cores para Aplicações Específicas

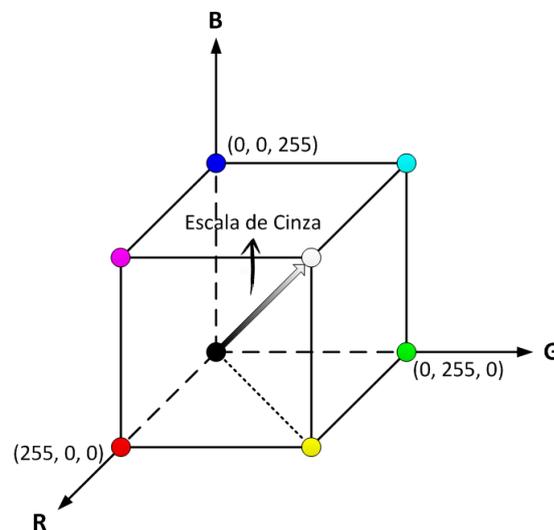
Abrange os espaços de cores adotados nos sistemas de televisão (YUV, YIQ), sistemas de fotografia (Kodak PhotoYCC) e sistemas de impressão (CMY(K)).

### Espaço de Cores CIE

Propostos pela CIE (*Commission Internationale de l'Eclairage*), possuem algumas propriedades de grande importância como a independência do dispositivo e a linearidade perceptual (CIE XYZ, Lab e Luv).

Uma representação gráfica do espaço de cores RGB (atualmente o mais utilizado em câmeras e monitores de vídeo) é apresentada na Figura 15:

Figura 15 – Representação gráfica do espaço de cores RGB



Fonte: Adaptado de ([FILHO; NETO, 1999](#))

Conforme pode ser visto, o espaço de cores RGB é baseado em um sistema de coordenadas cartesianas 3D, que pode ser visto como um cubo onde 3 de seus vértices representam as cores primárias, outros 3 as cores secundárias, o vértice junto à origem representa o preto e o mais afastado da origem representa a cor branca.

É interessante destacar ainda que, neste modelo, a escala de cinza é representada pela diagonal do cubo que se estende da origem (preto) até o vértice oposto (branco).

### 3.3.2 Operações Básicas

Ao representar uma imagem por uma matriz de pixels, pode-se efetuar uma miríade de operações matemáticas sobre esses dados para implementar a aplicação que se deseja. Do ponto de vista prático, as operações matemáticas utilizadas correspondem a derivadas, convoluções, operações estatísticas (médias simples ou ponderadas, medianas) transformadas (de Fourier, Wavelet, de Hough), operações booleanas, dentre outras.

Uma peculiaridade ao se trabalhar com imagens, que pode se tornar uma das principais dificuldades de implementação, está no fato de que uma imagem quase nunca é uma função “bem comportada”(i.e. contínua e derivável ao longo do intervalo de interesse) (MAINTZ, 2005). Deste modo, tornam-se necessárias algumas aproximações e adaptações nos cálculos (cujos detalhes não serão abordadas neste trabalho).

A lista a seguir explica, sucintamente, o conceito das principais operações básicas utilizadas no algoritmo desenvolvido (vide Capítulo 5):

#### Mudança de Espaço de Cores (*Color Space Conversion*)

Em alguns cenários, é interessante trabalhar com um espaço de cores diferente do padrão utilizado pela maioria das câmeras comerciais (i.e. RGB).

Essa necessidade costuma decorrer de 3 casos:

- Aplicações onde a utilização de um espaço de cores específico ressalte alguma característica desejável da imagem;
- Redução da quantidade de dados a ser processada (sem perda de informação relevante);
- Adaptação a algum padrão exigido (e.g. padrão para transmissão de TV).

O caso mais comum de mudança de espaço de cores é a conversão de RGB para Escala de Cinza.

De acordo com Cook (1999), o método normalmente utilizado para realizar essa conversão (i.e. mais adequado aos casos gerais e mais utilizado nos *softwares* de edição de imagens, como o GIMP - *GNU Image Manipulation Program*), é o **método da luminosidade** (Equação 3.1).

$$\text{Intensidade} = (0,299 \times R) + (0,587 \times G) + (0,114 \times B) \quad (3.1)$$

Tal método consiste no cálculo da média ponderada dos canais, cujos pesos foram calculados com base em estudos da percepção do sistema visual humano.

### Operações Morfológicas (*Morphological Operations*)

Conjunto de técnicas e ferramentas baseadas na teoria da **morfologia matemática**, cujos resultados dependem mais da disposição espacial dos dados da imagem de entrada do que de seus valores em si (FISHER et al., 2005).

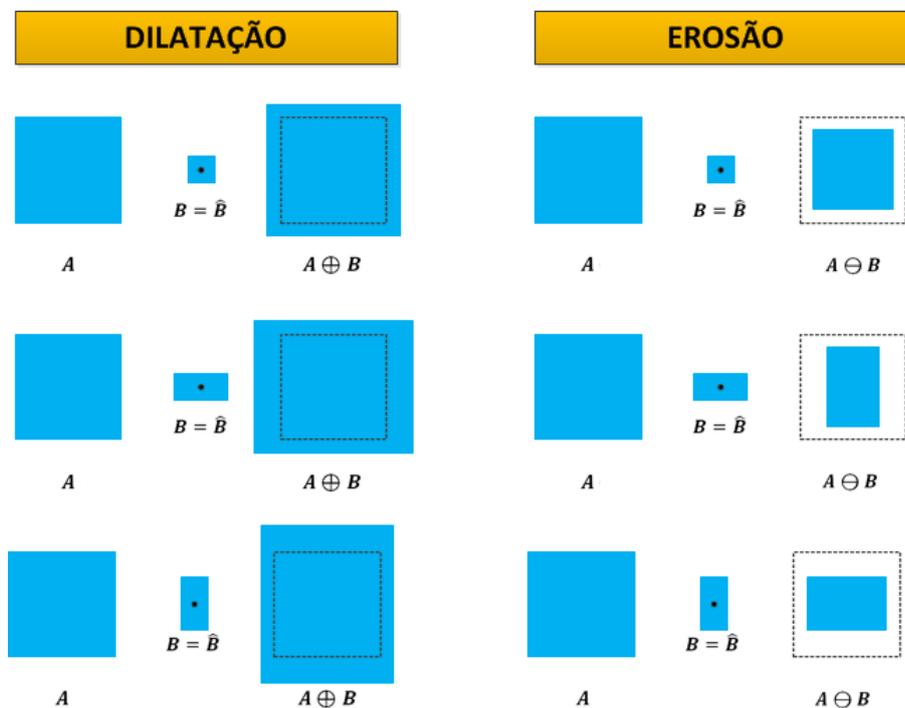
O principal objetivo da morfologia matemática é extrair informações relativas à geometria e à topologia de um conjunto desconhecido (neste contexto uma imagem), pela transformação através de outro conjunto completamente definido, chamado de **elemento estruturante**. Portanto, a base da morfologia matemática é a **teoria dos conjuntos** (FILHO; NETO, 1999).

De acordo com Davies (2012), existem duas operações morfológicas básicas que dão origem à grande parte das outras:

- **Dilatação (*Dilation*)** - Promove o CRESCIMENTO espacial de um objeto de maneira uniforme;
- **Erosão (*Erosion*)** - Promove a DIMINUIÇÃO espacial de um objeto de maneira uniforme (i.e. contraparte da operação de dilatação).

A Figura 16 traz um exemplo da aplicação de operações morfológicas de dilatação e erosão utilizando elementos estruturantes de formatos distintos:

Figura 16 – Exemplo da aplicação de operações morfológicas utilizando elementos estruturantes de formatos distintos



Fonte: Adaptado de (FILHO; NETO, 1999)

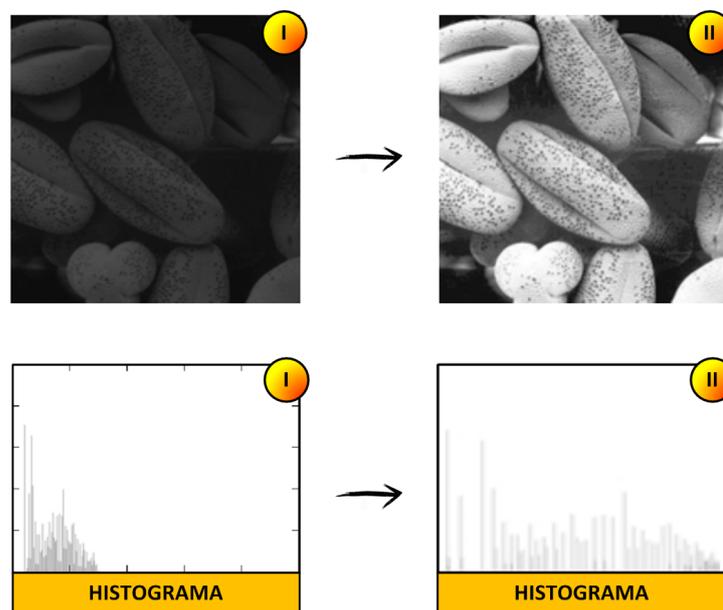
### Equalização do Histograma (*Histogram Equalization*)

O **histograma** de uma imagem é uma estrutura de dados que retrata a quantidade de pixels que é representada por cada valor possível do espaço de cores (e.g. no caso de uma imagem em escala de cinza, o histograma vai de 0 a 255).

A equalização do histograma visa aumentar o contraste de uma determinada imagem através da redistribuição de seu espectro de intensidades, aproveitando ao máximo a faixa de valores disponível (FISHER et al., 2005).

A Figura 17 traz um exemplo da aplicação de um processo de equalização de histograma:

Figura 17 – Exemplo da aplicação de um processo de equalização de histograma



Fonte: Adaptado de (GONZALEZ; WOODS, 2002)

### Correção de Iluminação (*Illumination Correction*)

Etapa de pré-processamento que visa corrigir os efeitos provocados por uma iluminação não uniforme.

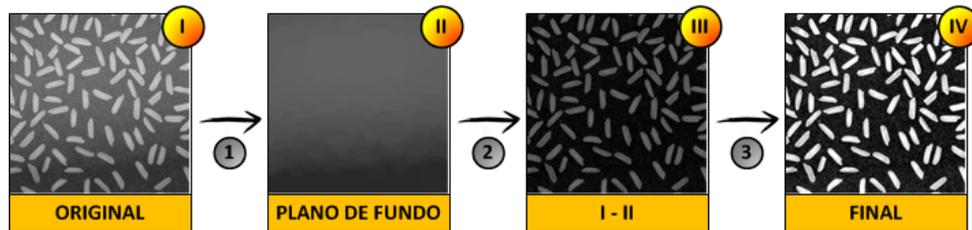
O processo consiste em 3 etapas<sup>4</sup>:

1. Aplica-se uma operação morfológica de abertura (i.e. *opening*) na imagem, utilizando um elemento estruturante em forma de disco (a imagem resultante será algo semelhante ao padrão de iluminação da imagem original);
2. Subtrai-se o resultado obtido da imagem original;
3. Realiza-se uma equalização do histograma para aumentar o contraste da imagem.

<sup>4</sup>As etapas 1 e 2 do processo são equivalentes a realizar uma filtragem da imagem com um filtro do tipo *top hat*.

A Figura 18 mostra um exemplo da aplicação de um algoritmo de correção de iluminação feito em MATLAB:

Figura 18 – Exemplo da aplicação de um algoritmo de correção de iluminação feito em MATLAB



Fonte: Adaptado de (MATHWORKS, 2014a)

### Detecção de Bordas (*Edge Detection*)

Processo muito utilizado, sobretudo em etapas de segmentação de imagem, que visa detectar a fronteira entre duas regiões de uma imagem.

As bordas dos objetos são zonas de descontinuidade dos níveis de intensidade luminosa de uma imagem, que podem ser encontradas por meio do cálculo do vetor gradiente (i.e. magnitude e orientação). Para tal, pode-se trabalhar definindo um limiar ou buscando os máximos locais.

De acordo com (GONZALEZ; WOODS, 2002), a detecção de bordas possui a grande vantagem de reduzir consideravelmente (normalmente por um fator de 100) a redundância dos dados de uma imagem, diminuindo significativamente a quantidade de memória e a velocidade de processamento exigida pelos algoritmos subsequentes.

A Figura 19 mostra um exemplo da aplicação de um algoritmo de detecção de bordas:

Figura 19 – Exemplo da aplicação de um algoritmo de detecção de bordas



Fonte: Adaptado de (FILHO; NETO, 1999)

### 3.3.3 Principais Processos

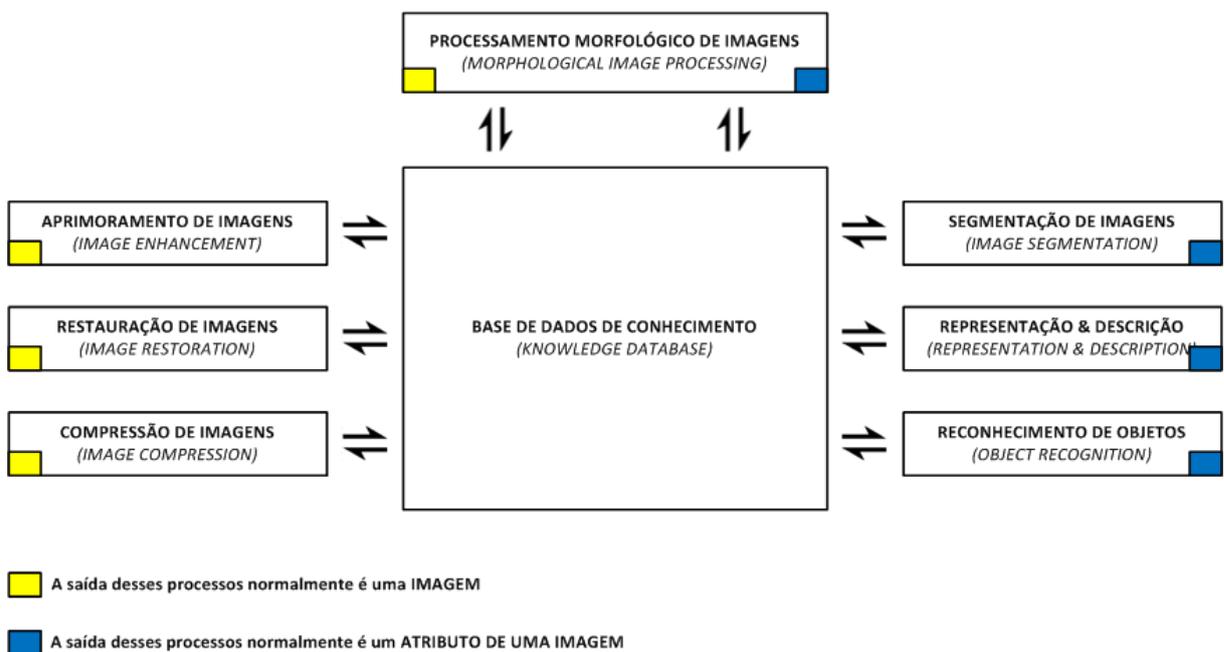
As técnicas de processamento de imagens possuem aplicações em inúmeras áreas, tais como industrial, militar, medicina e entretenimento. De acordo com [Gonzalez e Woods \(2002\)](#), o crescente interesse nessas técnicas advém de 2 desejos:

1. Melhorar a informação pictórica para a interpretação humana;
2. Processar os dados de uma imagem para permitir a representação, armazenamento e transmissão por parte de máquinas autônomas.

Dentro de um contexto mais amplo, que abrange tanto o processamento de imagens quanto a visão computacional, normalmente pode-se enquadrar os processos relacionados a imagens em uma ou mais categorias da lista não exaustiva fornecida abaixo (adaptada de [Gonzalez e Woods \(2002\)](#)).

As informações da lista, devidamente compiladas, são expostas na [Figura 20<sup>5</sup>](#):

Figura 20 – Diagrama dos principais processos aplicados a imagens



Fonte: Adaptado de ([GONZALEZ; WOODS, 2002](#))

<sup>5</sup>Apesar do diagrama omitir o processo de Exibição de Imagens (*Image Display*), este processo pode ser feito na saída de qualquer um dos estágios representados.

### **Aprimoramento de Imagens (*Image Enhancement*)**

Visa favorecer a visualização da imagem ou, simplesmente, evidenciar alguma característica de interesse desta.

Um exemplo clássico são as técnicas de aumento de contraste, que fazem a imagem “parecer melhor”. É importante frisar o grau de subjetividade associado a este processo, visto que a ideia de melhor depende do avaliador.

### **Restauração de Imagens (*Image Restoration*)**

Assim como o processo de aprimoramento de imagens, também trata da melhoria da aparência de uma imagem.

Os dois processos se diferenciam no sentido de que, na restauração de imagens, os elementos de avaliação do resultado são sempre objetivos (baseados em modelos matemáticos ou probabilísticos da degradação de imagens).

### **Compressão de Imagens (*Image Compression*)**

Como o nome sugere, este processo visa reduzir o armazenamento necessário para salvar uma imagem ou a largura de banda necessária para transmiti-la.

Devido à difusão do uso de arquivos de imagem que utilizam diversos padrões de compressão (tais como os de extensão *.jpg*, *.bmp*, *.tiff* e *.png*), o conceito deste processo se tornou familiar à grande maioria dos usuários de computador.

### **Processamento Morfológico de Imagens (*Morphological Image Processing*)**

Consiste em extrair as informações relativas à geometria e à topologia de uma imagem através do uso da **morfologia matemática**<sup>6</sup>.

As técnicas podem ser aplicadas em várias áreas de processamento e análise de imagens, com objetivos tão distintos como realce, filtragem, segmentação, detecção de bordas, esqueletização, afinamento, dentre outras.

### **Segmentação de Imagens (*Image Segmentation*)**

Particiona a imagem, decompondo-a em suas partes constituintes (e.g. objetos, regiões, plano de fundo).

A implementação de um procedimento robusto de segmentação é fundamental para o sucesso de uma solução que requer que objetos sejam identificados individualmente. Em geral, quanto mais preciso o processo de segmentação, maior é a eficácia do algoritmo de reconhecimento de objetos.

A saída deste processo normalmente é feita sob a forma de dados brutos (i.e. pixels), que representam as fronteiras entre as regiões ou os pontos internos a cada região.

---

<sup>6</sup>No contexto da biologia, a expressão **morfologia** se refere ao estudo da estrutura dos animais e plantas. A morfologia matemática, por sua vez, concentra seus esforços no estudo da estrutura geométrica das entidades presentes em uma imagem (FILHO; NETO, 1999).

### **Representação e Descrição (*Representation and Description*)**

Converte os dados de saída da etapa de segmentação a uma forma adequada às etapas de processamento subsequentes.

A 1ª decisão é relativa ao tipo de **representação** dos dados (i.e. fronteiras ou regiões). Em seguida, deve-se especificar um método para descrever os dados, de modo a destacar as características de interesse. A **descrição**, também conhecida como **seleção de características**, trata de extrair atributos que resultem em alguma informação quantitativa de interesse, ou que sejam fundamentais para diferenciar uma classe de objetos das outras.

### **Reconhecimento de Objetos (*Object Recognition*)**

Atribui um rótulo a um objeto com base em seus descritores (e.g. “veículo”, “fruta”, “rosto”).

No diagrama da Figura 20, o bloco denominado **Base de Dados de Conhecimento** representa todo o conhecimento prévio a respeito do problema em análise (que deve ser codificado dentro do sistema de processamento de imagens a ser desenvolvido).

Este conhecimento pode ser tão simples quanto a especificação das regiões da imagem onde as informações de interesse podem estar contidas, de forma a reduzir o espaço de busca do algoritmo.

Por outro lado, a base de dados de conhecimento pode ser deveras complexa, como uma lista inter-relacionada de todos os possíveis defeitos em um problema de inspeção de peças ou um banco de imagens que contém imagens em alta resolução de uma região (para utilização em aplicações capazes de detectar mudanças na região).

Finalmente, convém destacar que nem todos os processos ilustrados no diagrama da Figura 20 precisam estar presentes em uma aplicação da área de processamento de imagens e visão computacional. Entretanto, a medida que a complexidade do problema aumenta, o número de processos necessários para resolvê-lo também tende a aumentar.

## 3.4 Visão Computacional

Visão Computacional é o campo de estudos de sistemas computadorizados capazes de adquirir, processar e interpretar imagens (FILHO; NETO, 1999).

De acordo com Shapiro e Stockman (2001), o objetivo geral da visão computacional é, com o mínimo de interferência humana possível, tomar decisões relativas a objetos presentes em diferentes cenários do mundo real através de informações contidas em imagens. De maneira informal e simplista, pode-se dizer, ainda, que o objetivo é “ensinar robôs à enxergar” (FILHO; NETO, 1999).

Apesar de ser uma área relativamente nova, a visão computacional sofreu uma notável evolução nos últimos 25 anos (DAVIES, 2012). Tal progresso foi motivado, sobretudo, pelo desenvolvimento exponencial que vem ocorrendo em 2 campos tecnológicos que influenciam diretamente no desempenho das aplicações:

### Sistemas de Aquisição de Imagens

Surgimento de dispositivos de alta definição e baixo custo (e.g. câmeras fotográficas, câmeras de vídeo).

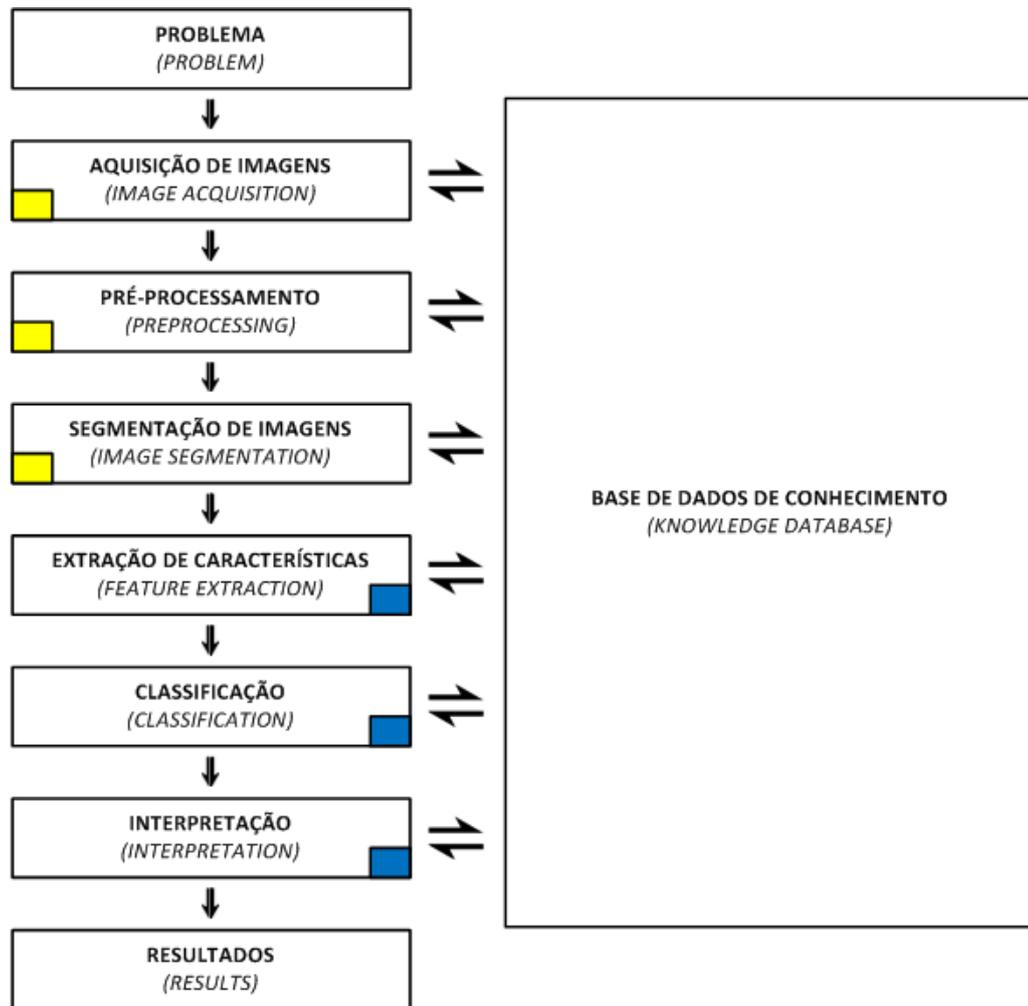
### Sistemas Computacionais

Evolução dos dispositivos através do aumento de seu poder de processamento e de sua quantidade de memória, aliada à redução de seus custos e de seu tamanho.

Regida pela famosa Lei de Moore, a evolução desses sistemas criou o cenário ideal para o desenvolvimento/aperfeiçoamento de aplicações em diversas áreas. Deste modo, o que era algo restrito a pesquisas acadêmicas e aplicações militares e espaciais, passou a ser utilizado em aplicações ligadas a imagens biomédicas, inspeção de qualidade nas indústrias, vigilância e monitoramento (DAVIES, 2012).

A Figura 21 mostra um diagrama de blocos das principais etapas realizadas por um sistema de visão computacional:

Figura 21 – Diagrama de blocos das principais etapas realizadas por um sistema de visão computacional



 A saída desses processos normalmente é uma IMAGEM

 A saída desses processos normalmente é um ATRIBUTO DE UMA IMAGEM

Fonte: Adaptado de Filho e Neto (1999)

### 3.4.1 Sistemas de Visão Computacional aplicados à Indústria<sup>7</sup>

Na indústria, os sistemas de visão computacional são utilizados para inspecionar linhas de produção de forma semi ou totalmente automatizada, capturando imagens de peças ou produtos e analisando estas imagens através de um *software* embarcado em um computador ([Excellentware Brasil, 2004](#)).

Sistemas dessa natureza costumam ser classificados de acordo com o tipo de tarefa a ser executada. Os autores [Demant, Streicher-Abel e Garnica \(2013\)](#) sugerem a seguinte classificação<sup>8</sup>:

- Sistemas de contagem;
- Sistemas de posicionamento (e.g. de rótulos, etiquetas, componentes);
- Sistemas de identificação de marcas;
- Sistemas de análise e categorização de produtos a partir de suas características (e.g. cor, formato, dimensões, textura);
- Sistemas de verificação do nível de líquidos;
- Sistemas de comparação entre imagens e objetos;

Essas tarefas, normalmente executadas *online* e em tempo real, permitem o controle do processo produtivo em níveis de **qualidade**, **precisão** e **velocidade** impossíveis de se conseguir de forma não automatizada.

Os resultados alcançados estão entre as metas mais almejadas pela administração de qualquer empresa competitiva, dentre os quais destaca-se ([Excellentware Brasil, 2004](#)):

- Aumento da produtividade;
- Melhoria da qualidade;
- Diminuição dos custos de produção;
- Geração de informações para um maior controle do processo produtivo.

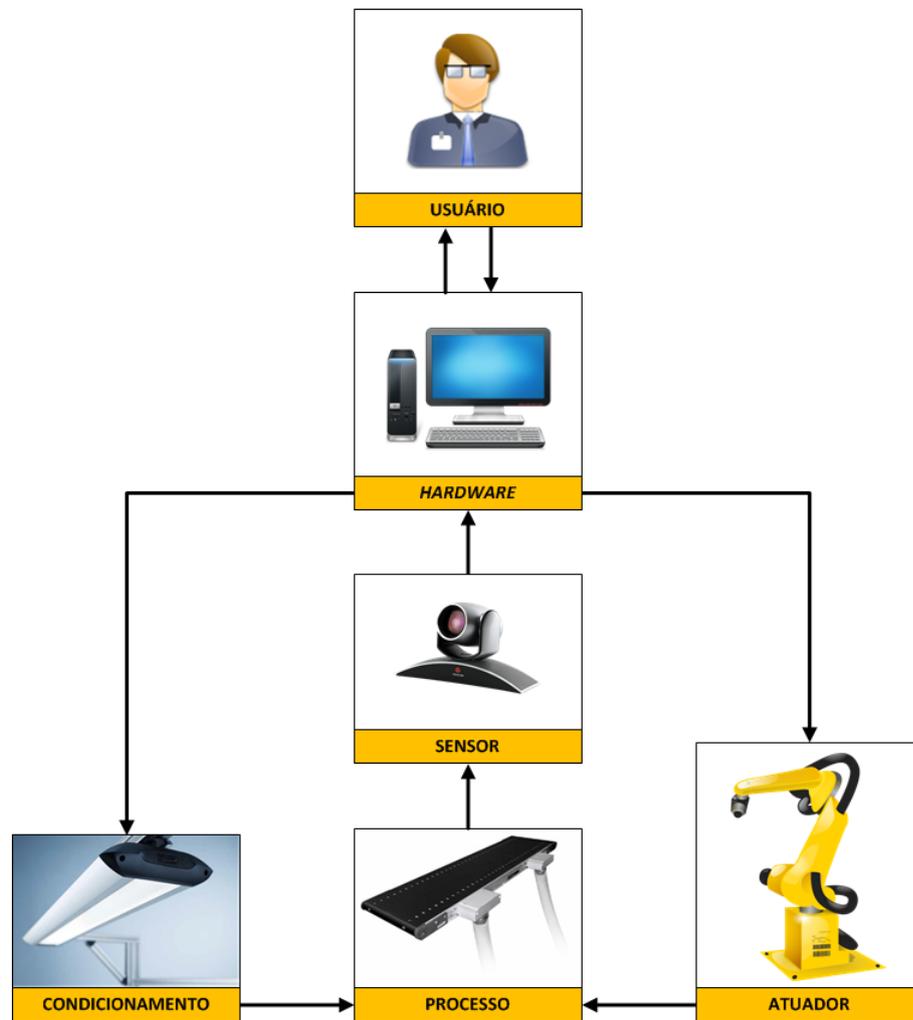
---

<sup>7</sup>De acordo com [Shapiro e Stockman \(2001\)](#), os sistemas de visão computacional aplicados à indústria normalmente são denominados *Machine Vision Systems* (**Sistemas de Visão de Máquina** ou **Sistema de Inspeção Automatizada**). Neste trabalho, independentemente do contexto, será utilizado apenas o termo genérico “Sistema de Visão Computacional”.

<sup>8</sup>Uma proposta de classificação diferente e igualmente interessante pode ser encontrada em ([Jähne; Haubeker, 2000](#)).

Como mostra a Figura 22, um sistema de visão computacional é composto por um conjunto de componentes, que atuam de maneira integrada para executar alguma tarefa relacionada ao processo:

Figura 22 – Diagrama de um sistema de visão computacional genérico voltado a aplicações industriais



Fonte: Autoria própria

De maneira resumida, os componentes de um sistema de visão computacional genérico voltado a aplicações industriais podem ser descritos da seguinte maneira:

### Sensor

O sensor (neste caso representando todo o sistema de aquisição de imagens) é responsável por obter imagens do processo e transmiti-las ao computador. Conforme exposto na Seção 3.3, a natureza da imagem depende do dispositivo físico utilizado (e.g. câmeras fotográficas e de vídeo, radares, equipamentos de raio-X, de ressonância magnética e de ultrassom).

**Hardware (i.e. PC,  $\mu$ C, DSP, FPGA, ASIC)**

Dispositivo que possui o *software* de visão computacional embarcado. Sua função é receber as imagens do sensor, processar as informações e enviar sinais de controle ao atuador e ao sistema de condicionamento.

O tipo de *hardware* utilizado depende da aplicação, podendo variar entre um Computador Pessoal (PC), um MicroControlador ( $\mu$ C), um Processador Digital de Sinais (DSP), um Arranjo de Portas Programável em Campo (FPGA), um Circuito Integrado de Aplicação Específica (ASIC), dentre outros<sup>9</sup>.

Conforme exposto por Davies (2012), que compara as vantagens e desvantagens de cada tipo de *hardware*, esta escolha constitui uma decisão de projeto extremamente importante.

**Atuador**

Recebe instruções do *hardware* e age sobre o processo, executando a tarefa desejada.

**Condicionamento**

Garante as condições ideais para o bom funcionamento do sistema (e.g. iluminação adequada, posicionamento correto). Num sistema mais sofisticado, pode ser ajustado automaticamente, com base em instruções fornecidas pelo *hardware*.

É importante destacar que os blocos do diagrama representam **funções** a serem desempenhadas dentro do sistema. Na prática, os sistemas podem não possuir algumas dessas funções (e.g. condicionamento, atuador) ou um mesmo módulo pode acumular diversas funções (e.g. uma câmera industrial com um processador embarcado).

---

<sup>9</sup>Apesar dos termos técnicos estarem em português, suas respectivas siglas quase nunca são utilizadas na literatura. Deste modo, optou-se por utilizar as siglas do inglês.

### 3.4.2 Projeto de Sistemas de Visão Computacional

Embora possuam os mesmos componentes básicos, os sistemas de visão computacional são construídos sob medida para cada situação, visto que a tarefa a ser efetuada em cada ambiente produtivo possui suas próprias exigências e peculiaridades (Excellentware Brasil, 2004).

Conforme implícito no Tópico 3.4.1, o projeto desses sistemas é multidisciplinar, envolvendo profissionais de diferentes formações (e.g. óptica, elétrica, mecânica, computação).

Além disso, as decisões de projeto de uma determinada etapa influenciam enormemente nas demais etapas, o que exige que todo o projeto seja feito de maneira integrada, evitando, assim, retrabalhos.

A visão computacional é uma disciplina da engenharia e, como tal, deve se apoiar na ciência e na compreensão dos processos fundamentais. No entanto, como em qualquer projeto de engenharia, sua elaboração deve ser **baseada em especificações**. Uma vez que as especificações do sistema estejam definidas, verifica-se como elas se relacionam com as restrições da natureza e da tecnologia. Isso traz algumas pistas de como o problema pode ser resolvido (DAVIES, 2012).

Sendo assim, o 1º passo do projeto e provavelmente o mais importante é definir as especificações do sistema. De modo a sistematizar o levantamento dessas informações junto ao cliente, é interessante utilizar um *checklist*, como o proposto pelo CTMV (*Consulting Team Machine Vision*) e disponibilizado no capítulo de anexos de Hornberg (2006).

Em seguida, costuma-se seguir as demais etapas transcritas na Tabela 1:

Tabela 1 – Etapas de projeto de um sistema de visão computacional

ETAPA	DESCRIÇÃO
01	Ouvir a respeito do problema.
02	Analizar a situação.
03	Observar os dados.
04	Testar os algoritmos mais óbvios.
05	Perceber as limitações.
06	Aprimorar os algoritmos.
07	Constatar que a tarefa é difícil e, até certo ponto, impossível.
08	Revisar a teoria para encontrar a fonte das limitações do sistema.
09	Realizar outros testes.
10	Encontrar uma abordagem mais adequada.
11	Reavaliar as especificações do projeto.
12	Decidir sobre continuar o projeto.
13	Finalizar a etapa de <i>software</i> do sistema.
14	Avaliar as limitações de velocidade.

*Continua na próxima página...*

Continuação da Tabela 1.

ETAPA	DESCRIÇÃO
15	Começar de novo (se necessário).
16	Melhorar a velocidade do <i>software</i> .
17	Chegar a um resultado aceitável.
18	Realizar testes sistemáticos com 1 000 imagens.
19	Projetar uma implementação em <i>hardware</i> .
20	Reformular o <i>software</i> (se necessário).
21	Realizar testes sistemáticos com outras 100 000 imagens.
22	Avaliar as dificuldades associadas a eventos raros.
23	Avaliar os problemas de temporização.
24	Validar o sistema final.

Fonte: Adaptado de (DAVIES, 2012)

Davies (2012) elaborou também uma lista das principais dificuldades de projeto de um sistema de visão computacional (Tabela 2). De maneira geral, sua avaliação indica que, apesar de todos os esforços, ainda existe um nível de subjetividade indesejável inerente ao projeto desse tipo de sistema.

Tabela 2 – Principais dificuldades inerentes ao projeto de um sistema de visão computacional

#	DESCRIÇÃO
01	Nem sempre é evidente avaliar se existe uma solução para o problema proposto (mais precisamente uma solução financeiramente viável).
02	Especificações nem sempre podem ser feitas de maneira objetiva (i.e. lógica binária ao invés de lógica <i>fuzzy</i> ).
03	Normalmente não existe uma metodologia científica de projeto para chegar à solução a partir das especificações (sobretudo de maneira ótima).
04	Os parâmetros de otimização não são óbvios e nem suas prioridades relativas são claras.
05	Pode ser extremamente difícil avaliar se uma solução é melhor do que outra (faltam critérios objetivos).
06	Alguns ambientes de inspeção dificultam avaliar se existe uma solução válida ou não.

Fonte: Adaptado de (DAVIES, 2012)

A Figura 23 abaixo representa as especificações de projeto de um sistema de visão computacional e alguns exemplos de disciplinas e ferramentas relacionadas:

Figura 23 – Especificações de projeto de um sistema de visão computacional



Fonte: Adaptado de (DAVIES, 2012)

O conjunto de especificações listado acima pode, em qualquer estágio de desenvolvimento tecnológico, ser inalcançável (sobretudo em termos de *hardware*). Caso alguma especificação exija mais do que o limite do sistema projetado inicialmente possa fornecer, tem-se basicamente duas alternativas:

- Trocar de algoritmo;
- Optimizar os parâmetro internos do algoritmo, reajustando-os de modo a garantir que todas as especificações sejam respeitadas (mesmo que isso implique em sacrificar o desempenho do algoritmo em relação à outras especificações).

Em geral, existirão algumas especificações inflexíveis (i.e. bastante restritivas e que devem ser absolutamente respeitadas) e outras que possuem uma certa flexibilidade (DAVIES, 2012). Deste modo, o grande desafio do projetista é dominar toda essa miríade de ferramentas e conceitos relacionados ao projeto e encontrar uma relação de compromisso entre as especificações do sistema, de modo a satisfazer as necessidades do cliente.

## 4 Segmentação de Imagens

### 4.1 Introdução

De modo a identificar os métodos de segmentação de imagens mais adequados à resolução do problema proposto, este capítulo faz uma análise comparativa dos diferentes métodos com base em suas principais características.

A 1ª seção traz o conceito de segmentação de imagens, descreve resumidamente os principais métodos existentes na literatura e apresenta os 3 métodos utilizados no trabalho. Finalmente, as seções posteriores são dedicadas ao detalhamento dos métodos escolhidos.

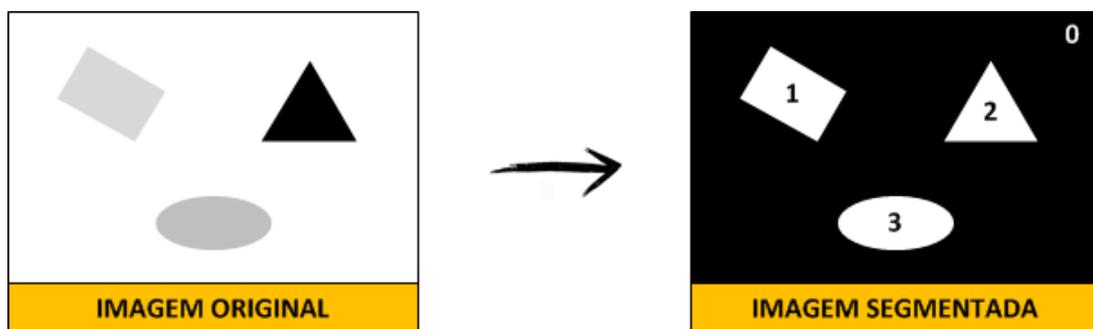
### 4.2 Processo de Segmentação de Imagens

#### 4.2.1 Conceito

De acordo com [Gonzalez e Woods \(2002\)](#), **segmentação de imagens** é o processo que visa particionar uma imagem, decompondo-a em suas partes constituintes (e.g. objetos, regiões, plano de fundo).

A Figura 24 apresenta o resultado de um processo de segmentação realizado em uma imagem composta por diversas formas geométricas (de tamanhos e cores diferentes) dispostas em um fundo branco:

Figura 24 – Resultado de um processo de segmentação realizado em uma imagem simples



Fonte: Adaptado de ([HO, 2011](#))

A partir da imagem original, representada em escala de cinza e constituída por uma massiva quantidade de dados, o algoritmo de segmentação é capaz de separar os diferentes objetos do plano de fundo e identificá-los adequadamente.

Por ser uma tarefa intuitiva para o ser humano, facilmente executada até por um recém nascido, este processo pode parecer simples. Entretanto, a segmentação de imagens não-triviais é uma das tarefas mais difíceis no campo do processamento de imagens (GONZALEZ; WOODS, 2002).

Esta complexidade está associada ao fato que, para um computador, uma imagem é apenas uma matriz de pixels. O grande desafio, então, está em achar uma relação entre essa infinidade de dados, identificando as diferentes partes da imagem (em outras palavras, a dificuldade advém da subjetividade implícita à tarefa).

#### 4.2.2 Definição Matemática

Do ponto de vista matemático, o processo de segmentação de imagens se baseia na **teoria dos conjuntos**.

Seja  $I$  o conjunto de todos os pixels de uma imagem. Aplicando um processo de segmentação a essa imagem, obtém-se um conjunto de regiões  $\{R_1, R_2, R_3, \dots, R_n\}$  mutuamente exclusivas e que, quando combinadas, formam o conjunto  $I$ .

Formalmente, tem-se (KLETTE, 2014), (GONZALEZ; WOODS, 2002):

$$\begin{aligned}
 \text{(a)} \quad & I = \bigcup_{i=1}^n R_i \\
 \text{(b)} \quad & R_i \cap R_j = \emptyset, \forall i \neq j \\
 \text{(c)} \quad & R_i \neq \emptyset, \forall i \in \{1, 2, \dots, n\} \\
 \text{(d)} \quad & R_i \text{ é uma região conectada, } \forall i = 1, 2, \dots, n \\
 \text{(e)} \quad & P(R_i) = \text{TRUE}, \forall i = 1, 2, \dots, n \\
 \text{(f)} \quad & P(R_i \cup R_j) = \text{FALSE}, \forall i \neq j
 \end{aligned} \tag{4.1}$$

Onde  $P(R_i)$  é um predicado em  $R_i$  (i.e. uma função booleana).

### 4.2.3 Principais Métodos

Os algoritmos de segmentação de imagens normalmente utilizam 2 tipos de abordagem (GONZALEZ; WOODS, 2002):

#### **Verificação das Descontinuidades**

Consiste em particionar uma imagem com base em mudanças abruptas do valor de intensidade (como aquelas causadas por bordas de objetos).

TODO → PARTES

#### **Verificação das Similaridades**

Consiste em particionar uma imagem em regiões similares, de acordo com algum critério pré-definido.

PARTES → TODO

Atualmente, existe uma enorme variedade de métodos de segmentação de imagens disponíveis na literatura, cada qual com suas respectivas vantagens e desvantagens. Apesar de não existir um consenso em relação à taxonomia desses métodos, é interessante categorizá-los de alguma forma.

De acordo com Maintz (2005), é aparentemente impossível definir uma categorização disjunta, pois até mesmo dois métodos muito diferentes compartilham propriedades que desafiam esta categorização singular.

Sendo assim, a categorização abaixo, proposta por Maintz (2005), considera muito mais a ênfase da abordagem do que uma divisão estrita:

#### **Segmentação baseada em Limiar (*Threshold based Segmentation*)**

Utiliza técnicas de limiarização do histograma e/ou técnicas de corte para segmentar a imagem. Tais técnicas podem ser aplicadas diretamente à imagem ou podem ser combinadas com operações de pré- e pós-processamento.

#### **Segmentação baseada em Bordas (*Edge based Segmentation*)**

Através dessa técnica, considera-se que as bordas detectadas em uma imagem representam fronteiras entre os objetos, podendo ser utilizadas para identificá-los.

#### **Segmentação baseada em Regiões (*Region based Segmentation*)**

Enquanto técnicas baseadas em bordas buscam determinar as fronteiras de um objeto e depois encontrar o objeto em si (preenchendo a fronteira encontrada), as técnicas baseadas em regiões utilizam a abordagem oposta (i.e. partem de um ponto interno ao objeto e “crescem para fora”, até atingir sua fronteira).

### **Segmentação baseada em Técnicas de Clusterização (*Clustering Techniques*)**

Embora o termo clusterização às vezes seja utilizado como um sinônimo para técnicas de segmentação aglomerativas, aqui ele se refere às técnicas utilizadas principalmente na análise exploratória de dados de padrões de medição de múltiplas dimensões.

Nesse contexto, métodos de clusterização visam agrupar padrões semelhantes em algum sentido. Como esse objetivo é análogo ao que se deseja ao segmentar uma imagem, as técnicas de clusterização podem ser prontamente aplicadas.

### **Segmentação por Comparação (*Matching*)**

Quando o objeto a ser localizado na imagem é conhecido *a priori*, pode-se usar essa informação para localizá-lo através de uma técnica conhecida como comparação de imagens.

É interessante frisar que, além das classes de métodos supracitadas, existe na literatura alguns métodos híbridos, que misturam técnicas diferentes de modo a aproveitar as vantagens de ambos os métodos.

#### **4.2.4 Métodos Utilizados neste Trabalho**

Conforme exposto no tópico anterior (Tópico 4.2.3), a categorização dos métodos de segmentação de imagens deixa evidente que cada método possui características e peculiaridades distintas, o que determina seu grau de adequação à uma determinada aplicação.

Ciente desse fato e visando comparar o desempenho dos métodos, neste trabalho optou-se por implementar 3 métodos diferentes<sup>1</sup>:

- Método do *Threshold*;
- Método do *Watershed*;
- Método da Transformada de Hough;

Os princípios de funcionamento dos métodos escolhidos serão apresentados a seguir, nas Seções 4.3 a 4.5.

---

<sup>1</sup>A justificativa de escolha dos métodos utilizados pode ser encontrada no Capítulo 5.

## 4.3 Segmentação pelo Método do *Threshold*

### 4.3.1 Conceito

Conforme visto no Capítulo 3, uma imagem digital nada mais é do que uma matriz de pixels  $I(i, j)$ .

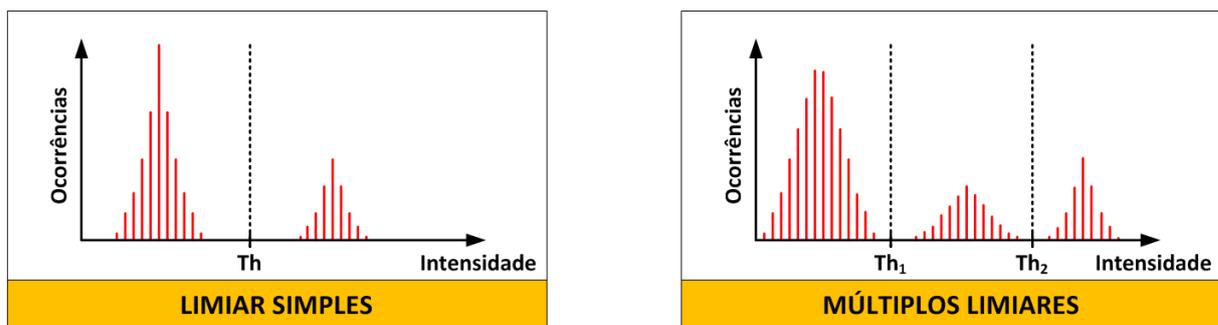
O histograma desta imagem, por sua vez, é uma estrutura de dados que retrata a quantidade de pixels que é representada por cada valor possível do espaço de cores. No caso de uma imagem em escala de cinza, por exemplo, cada pixel é descrito por um número inteiro que varia de 0 a 255 (do preto ao branco, respectivamente), denominado **intensidade**.

Em uma imagem composta por objetos claros em um fundo escuro, por exemplo, o histograma provavelmente será bimodal (i.e. possuirá 2 picos): o primeiro, maior e próximo ao preto, correspondente ao plano de fundo e o segundo, menor e próximo ao branco, representando os objetos.

Como o próprio nome indica, a proposta da segmentação pelo método do *threshold* é *definir um limiar, ao qual o valor de cada pixel será comparado*. Os pixels abaixo desse limiar serão classificados como plano de fundo e os pixels acima desse limiar como objetos.

A Figura 25 apresenta o histograma de duas imagens em escala de cinza compostas por objetos claros em um plano de fundo escuro.

Figura 25 – Histograma de duas imagens em escala de cinza compostas por objetos claros em um plano de fundo escuro



Fonte: Adaptado de (GONZALEZ; WOODS, 2002)

Matematicamente, a imagem limiarizada  $I_{Th}(i, j)$  pode ser definida por:

$$I_{Th}(i, j) = \begin{cases} 0 \text{ (Plano de Fundo),} & \text{se } I(i, j) < Th \\ 1 \text{ (Objeto),} & \text{se } I(i, j) > Th \end{cases} \quad (4.2)$$

### 4.3.2 Escolha do Limiar

O grande desafio da segmentação pelo método do *threshold* é justamente a escolha do limiar, que deve ser feita de acordo com as características do problema.

De acordo com [Gonzalez e Woods \(2002\)](#), o limiar pode ser representado matematicamente, de maneira genérica, por:

$$Th = Th[i, j, I(i, j), p(i, j)] \quad (4.3)$$

Onde,

$(i, j)$ : Coordenadas do ponto.

$I(i, j)$ : Nível de cinza do ponto  $(i, j)$ .

$p(i, j)$ : Propriedade local do ponto  $(i, j)$ .

Apesar do princípio fundamental da técnica ser sempre o mesmo (i.e. definir um limiar ao qual o valor de cada pixel será comparado), uma série de variações do método foram desenvolvidas visando aperfeiçoar seu funcionamento.

Pode-se classificar essas variações quanto a ([GONZALEZ; WOODS, 2002](#)):

#### Variação do Limiar

- **Global (*Global Thresholding*)**: O limiar utilizado é o mesmo para toda a imagem. Matematicamente:

$$Th = Th[I(i, j)]$$

- **Local (*Local Thresholding*)**: O limiar utilizado varia de acordo com alguma propriedade local do pixel em análise. Matematicamente:

$$Th = Th[I(i, j), p(i, j)]$$

- **Dinâmico/Adaptativo (*Dynamic/Adaptive Thresholding*)**: O limiar utilizado varia de acordo com alguma propriedade local e com a posição espacial do pixel em análise. Matematicamente:

$$Th = Th[i, j, I(i, j), p(i, j)]$$

#### Quantidade de Variáveis

Apesar de ser mais fácil compreender a aplicação do método em imagens em escala de cinza (onde cada pixel é caracterizado apenas pelo seu valor de intensidade), o mesmo é igualmente aplicável a outros tipos de imagens, como a RGB (onde cada pixel é caracterizado pelos valores de suas 3 componentes de cor primária aditiva).

Neste caso, a técnica é conhecida como Limiarização Multiespectral (*Multispectral Thresholding*) ou Limiarização Multivariável (*Multivariable Thresholding*).

### Quantidade de Limiares

Conforme ilustrado no histograma da Figura 25, uma imagem que contenha 2 objetos distintos e de cor clara dispostos em um fundo escuro pode precisar de mais de um limiar para ser corretamente segmentada.

Neste caso, a técnica é conhecida como Limiarização Multinível (*Multilevel Thresholding*).

### Método de Escolha do Limiar

O limiar pode ser escolhido de diversas formas, mas quase todos os métodos de escolha se baseiam no histograma.

Os principais métodos disponíveis na literatura são (MAINTZ, 2005) (DAVIES, 2012):

- **Manual;**
- **Baseado nos Picos do Histograma:** e.g. valor médio entre os picos, valor mínimo entre os picos;
- **Baseado na Variância:** e.g. Método de Otsu (i.e. minimização da variância de cada segmento e maximização da variância entre os segmentos);
- **Baseado na Entropia;**
- **Limiarização Ótima/Máxima Verossimilhança;**
- **Estimação do Histograma;**

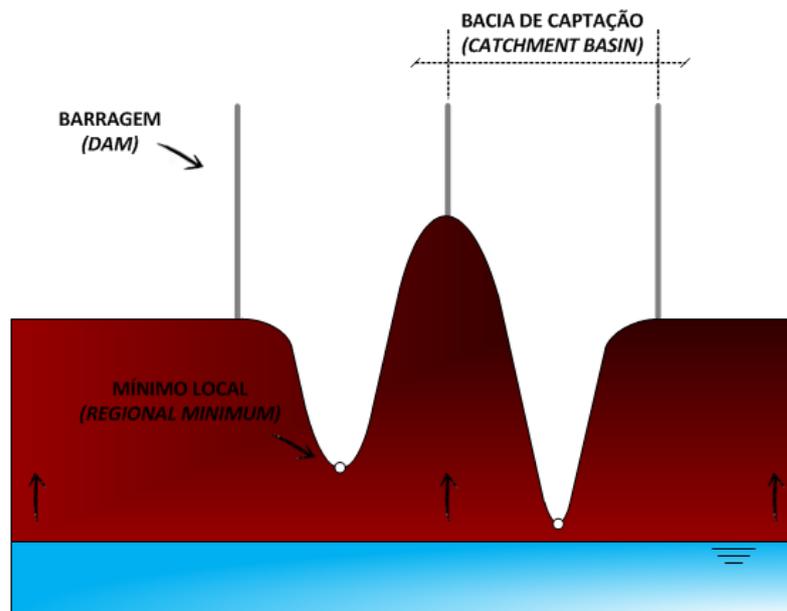
O método de Otsu, por ser relativamente simples, bem definido do ponto de vista matemático e fornecer bons resultados, é uma alternativa bastante utilizada na literatura.

## 4.4 Segmentação pelo Método do *Watershed*

### 4.4.1 Conceito

A técnica de segmentação por *watershed* consiste em uma transformação morfológica, cujo nome advém da analogia feita com os padrões de drenagem de água de uma paisagem (Figura 26).

Figura 26 – Elementos de uma paisagem utilizados na construção da analogia feita com o método de segmentação por *watershed* (Bacia Hidrográfica).



Fonte: Autoria própria

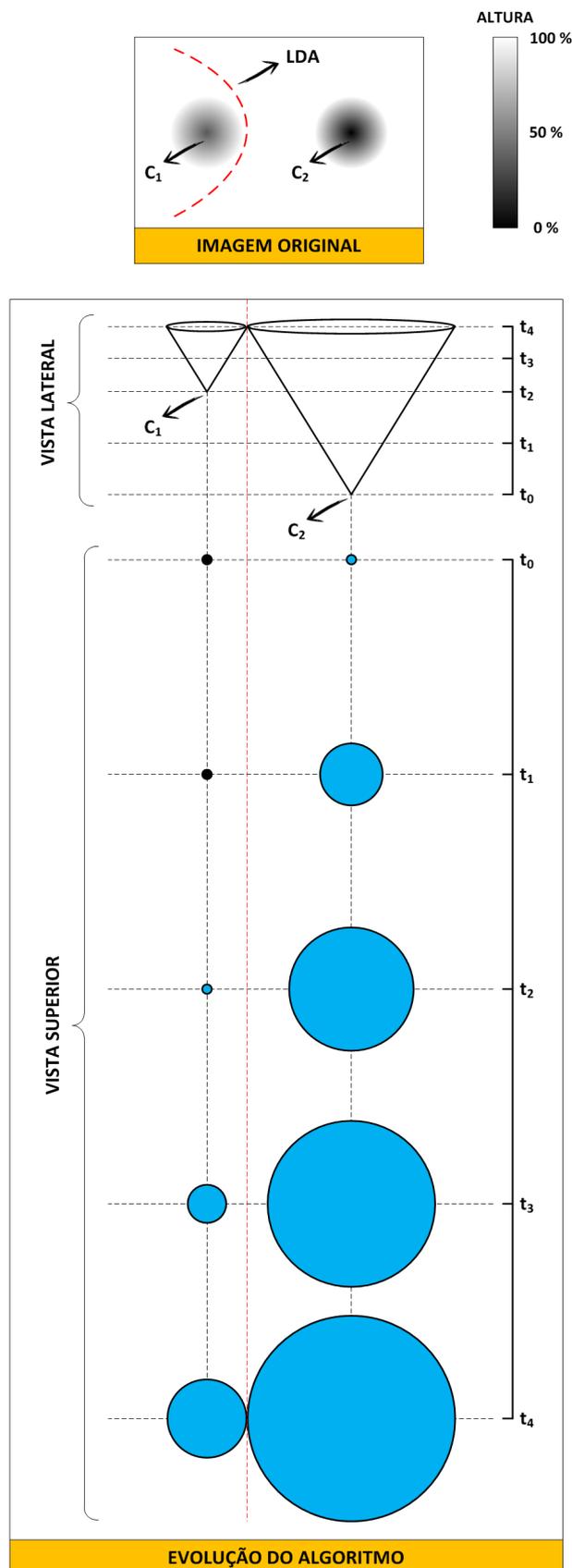
Seu princípio de funcionamento se baseia em interpretar uma imagem em escala de cinza como sendo um mapa topográfico, onde a intensidade de cada pixel é associada à uma altura (i.e. quanto mais escuro o pixel, menor a altura do ponto).

Em seguida, de acordo com [Gonzalez e Woods \(2002\)](#), pode-se enxergar a progressão do algoritmo da seguinte forma (vide Figura 27): imaginemos que um buraco seja feito em cada mínimo local da imagem, permitindo que a topografia seja inundada de baixo para cima a um fluxo constante.

Na medida em que o nível da água vai subindo, há um momento em que a água de duas bacias de captação diferentes fica na iminência de se encontrar. Nesse instante, é construída uma barragem para evitar que a água das duas fontes se misturem.

O nível da água continua a subir até o momento em que apenas as barragens não estejam submersas. Deste modo, as fronteiras definidas pelas barragens representam as linhas divisoras de água (LDAs), que correspondem às bordas contínuas extraídas pelo algoritmo de segmentação.

Figura 27 – Visualização da progressão do algoritmo de segmentação por *watershed* aplicado a uma imagem artificial.



Fonte: Autoria própria

É importante frisar que, na prática, ao invés de aplicar este algoritmo à imagem em si, é recomendável aplicá-lo ao gradiente da imagem (GONZALEZ; WOODS, 2002). A razão para tal pode ser visualizada na Figura 24.

Como existe uma diferença muito grande entre o nível de cinza dos objetos, o fato de utilizar o algoritmo diretamente na imagem original acaba “privilegiando” a segmentação dos objetos mais escuros. Ao trabalhar com o gradiente da imagem, todas as regiões caracterizadas por pequenas variações dos níveis de cinza terão um valor de gradiente pequeno (GONZALEZ; WOODS, 2002).

#### 4.4.2 Utilização de Marcadores

De acordo com Gonzalez e Woods (2002), a aplicação direta da técnica de segmentação por *watershed* geralmente conduz à segmentação excessiva da imagem (devido à presença de ruídos e outras irregularidades do gradiente), podendo arruinar completamente o resultado obtido.

Uma solução prática para este problema é limitar o número de regiões possíveis através da utilização de **marcadores**. Um marcador é um componente conectado pertencente à uma imagem, podendo ser **interno** (i.e. associado a objetos) ou **externo** (i.e. associado ao plano de fundo).

Assim como no método do *threshold* o maior problema é escolher o limiar, no método do *watershed* a grande dificuldade é escolher adequadamente os marcadores.

Caso a tarefa seja feita de forma semi-automatizada, essa escolha cabe ao operador e pode ser feita de maneira intuitiva.

Caso contrário, normalmente o procedimento consiste em 2 etapas:

- **Pré-Processamento:** e.g utilização de um filtro Gaussiano;
- **Classificação:** A escolha dos marcadores é feita com base em critérios pré-estabelecidos, podendo variar de procedimentos simples (baseados nos níveis de cinza e na conectividade) a procedimentos muito complexos (envolvendo tamanho, forma, localização, distâncias relativas)<sup>2</sup>.

---

<sup>2</sup>O que implica em utilizar um conhecimento *a priori* na construção do algoritmo. É importante lembrar que a possibilidade da inclusão deste tipo de conhecimento no algoritmo é uma vantagem significativa deste método.

## 4.5 Segmentação pelo Método da Transformada de Hough

### 4.5.1 Conceito

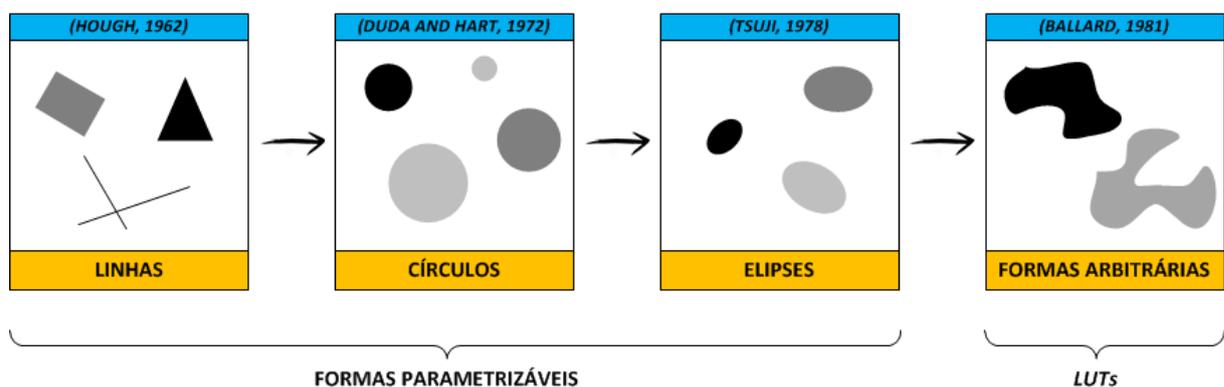
A transformada de Hough é uma operação matemática utilizada para alternar do espaço cartesiano  $(x, y)$  para um **espaço de parâmetros**<sup>3</sup>.

O objetivo dessa mudança de representação é *permitir a detecção de uma determinada classe de formas geométricas em uma imagem de maneira altamente robusta*, mitigando os efeitos indesejáveis provocados por fenômenos como oclusão, ruído e outros artefatos (DAVIES, 2012).

A robustez desse método está associada à utilização de um **processo de votação**, realizado no espaço de parâmetros, que elege os candidatos mais prováveis a objetos (i.e. que possuem a forma geométrica procurada).

A transformada de Hough foi originalmente concebida visando única e exclusivamente a detecção de linhas. Em seguida, como mostra a evolução exposta na Figura 28, a transformada foi adaptada à detecção de círculos, elipses, formas geométricas parametrizáveis quaisquer e, finalmente, formas geométricas arbitrárias, culminando na célebre *Generalized Hough Transform (GHT)*.

Figura 28 – Evolução das aplicações possíveis da transformada de Hough



Fonte: Autoria própria

Os tópicos seguintes (4.5.2 a 4.5.5) abordarão em maiores detalhes as características de cada uma destas etapas da evolução da transformada de Hough. Finalmente, o Tópico 4.5.6 apresenta alguns conceitos e técnicas que podem ser implementados para melhoria do desempenho global do algoritmo.

<sup>3</sup>Como será visto posteriormente, a dimensão do espaço de parâmetros e as variáveis utilizadas em sua definição dependem do problema em análise.

### 4.5.2 Detecção de Linhas

O conceito chave envolvido na detecção de linhas a partir da transformada de Hough é a **dualidade ponto-linha** (DAVIES, 2012).

Essa dualidade pode ser observada a partir da equação clássica definida por<sup>4</sup>:

$$y = ax + b \quad (4.4)$$

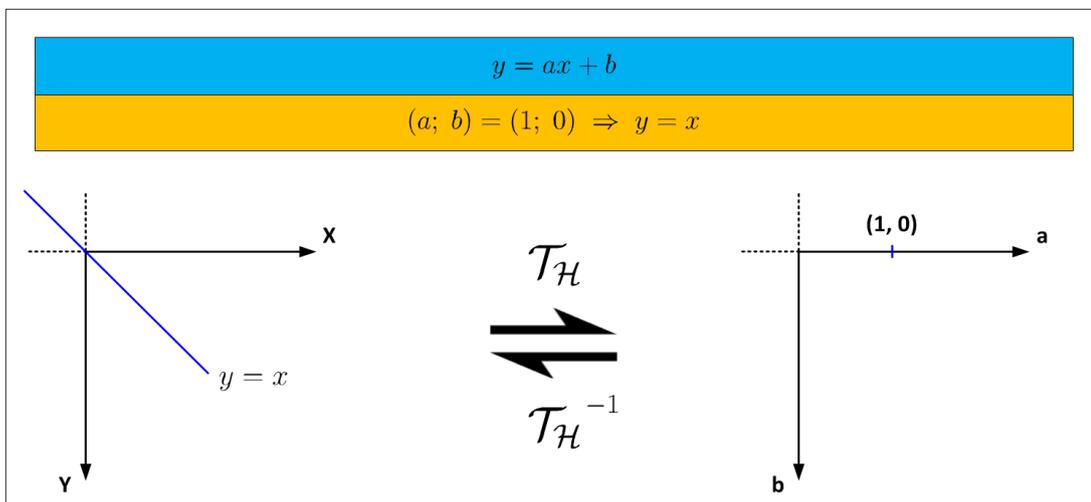
Habitualmente, costuma-se relacionar esta equação à representação de uma reta no espaço cartesiano  $(x, y)$ . Entretanto, o que a transformada de Hough propõe é representar essa expressão num espaço conhecido como **espaço de parâmetros**  $(a, b)$ . Em outras palavras, *uma reta em  $(x, y)$  pode ser representada por um ponto em  $(a, b)$  e vice-versa.*

Matematicamente, tem-se:

$$(x, y) \xrightarrow{\mathcal{T}_{\mathcal{H}}} (a, b) \quad (4.5)$$

As Figuras 29 e 30 mostram dois exemplos da relação entre o espaço cartesiano e o espaço de parâmetros  $(a, b)$ , evidenciando a dualidade existente:

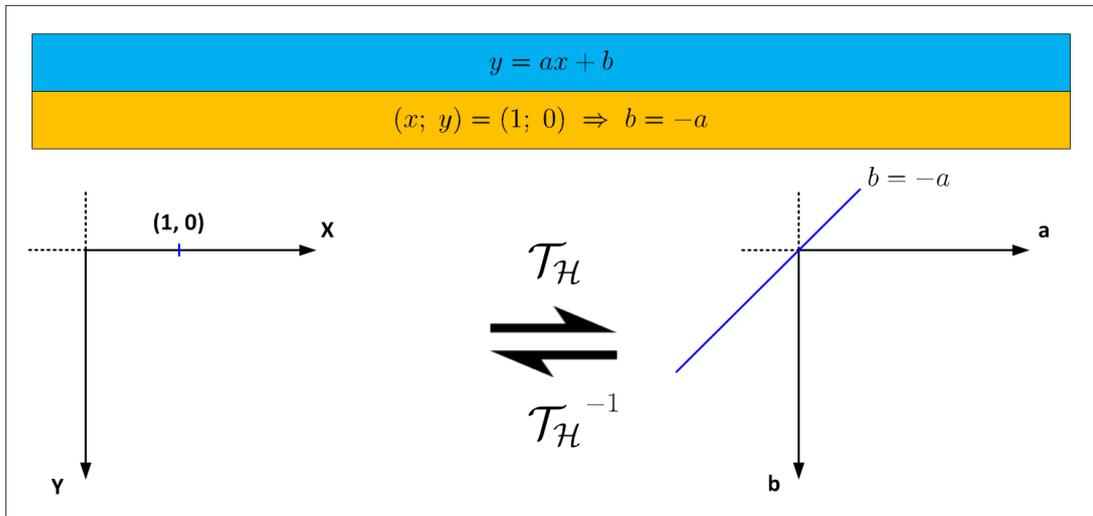
Figura 29 – Aplicação da transformada de Hough a uma reta no espaço cartesiano (resultando em um ponto no espaço de parâmetros  $(a, b)$ ).



Fonte: Autoria própria

<sup>4</sup>Essa representação costuma ser denominada de **inclinação-interseção** (do inglês, *slope-intersect*).

Figura 30 – Aplicação da transformada de Hough a um ponto no espaço cartesiano (resultando em uma reta no espaço de parâmetros  $(a, b)$ ).



Fonte: Autoria própria

Por outro lado, representar uma linha em coordenadas cartesianas como sugerido na Equação 4.4 possui uma limitação grave: *como descrever uma reta vertical no espaço de parâmetros?*

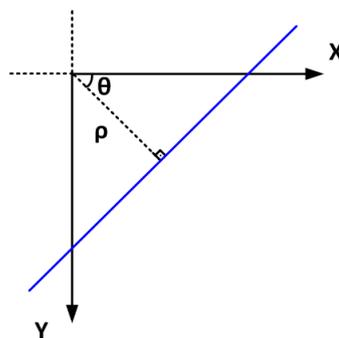
À medida que a inclinação da reta aumenta, o valor dos parâmetros  $(a, b)$  também aumenta e, no caso extremo de uma reta vertical, os parâmetros assumem valores que tendem a infinito (DAVIES, 2012).

Para resolver este problema, costuma-se adotar a notação em coordenadas polares:

$$(x, y) \xrightarrow{\mathcal{T}_H} (\rho, \theta) \tag{4.6}$$

Onde o significado dos parâmetros  $\rho$  e  $\theta$  pode ser visualizado na Figura 31 abaixo:

Figura 31 – Parâmetros  $\rho$  e  $\theta$  para representação de uma reta em coordenadas polares.



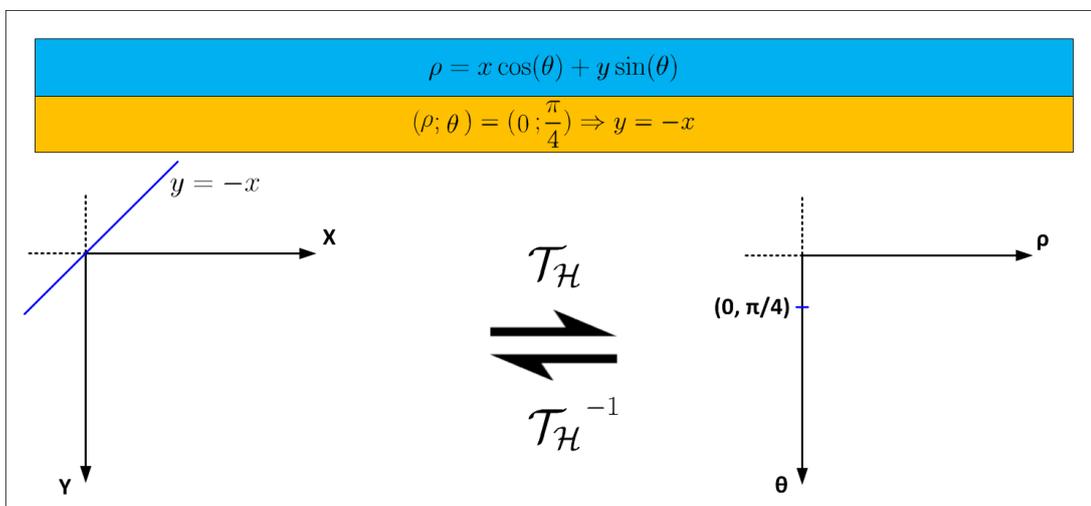
Fonte: Autoria própria

A partir da equação 4.4 e dos parâmetros definidos na Figura 31, pode-se deduzir a equação de uma linha em coordenadas polares:

$$\rho = x \cos(\theta) + y \sin(\theta) \tag{4.7}$$

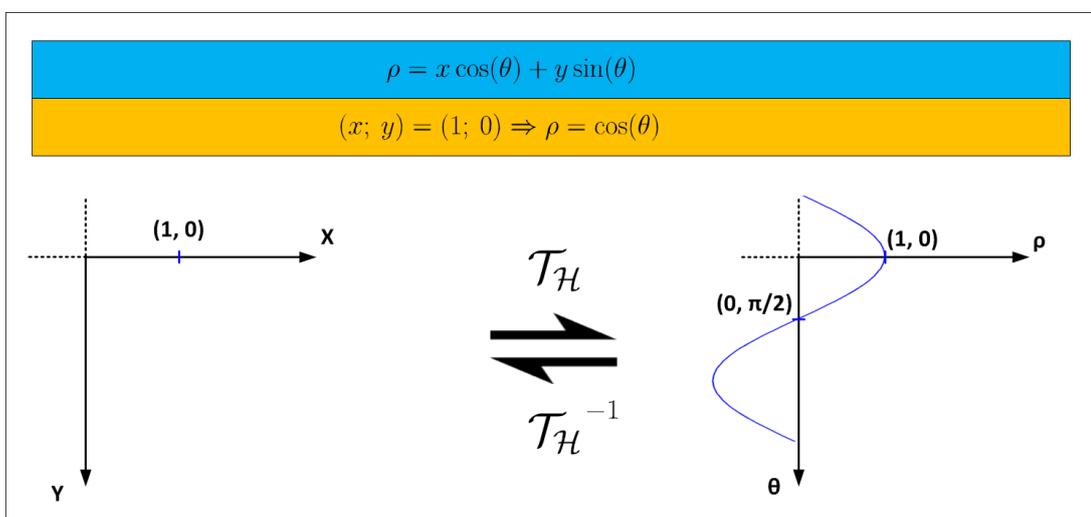
As Figuras 32 e 33 mostram dois exemplos da relação entre o espaço cartesiano e o espaço de parâmetros  $(\rho, \theta)$ . Apesar da dualidade ponto-linha não se aplicar mais a esta representação, o *modus operandi* do método é basicamente o mesmo.

Figura 32 – Aplicação da transformada de Hough a uma reta no espaço cartesiano (resultando em um ponto no espaço de parâmetros  $(\rho, \theta)$ ).



Fonte: Autoria própria

Figura 33 – Aplicação da transformada de Hough a um ponto no espaço cartesiano (resultando em uma senóide no espaço de parâmetros  $(\rho, \theta)$ ).



Fonte: Autoria própria

Na prática, a implementação da transformada de Hough exige a realização de algumas etapas preliminares.

Primeiramente, aplica-se um algoritmo de detecção de bordas à imagem original, seguida de uma etapa de limiarização (responsável por binarizar a imagem).

A partir daí, a imagem é analisada pixel a pixel. Conforme discutido anteriormente (vide Figura 33), cada pixel correspondente a uma borda irá gerar uma senóide no espaço de parâmetros, que será armazenada numa matriz conhecida como **acumulador**.

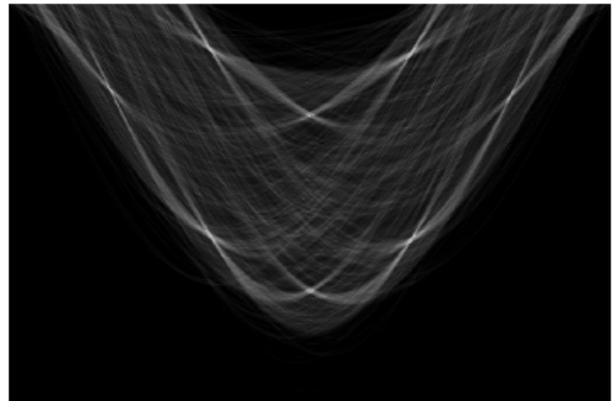
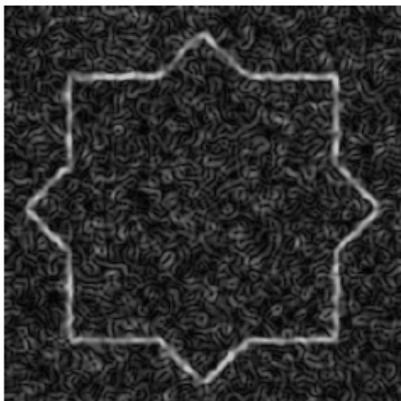
A medida que os pixels da imagem vão sendo analisados, as diferentes senóides irão se cruzar em pontos específicos, aumentando a evidência de que aquele ponto no espaço de parâmetros representa uma reta na imagem original.

Esse processo pode ser entendido como uma espécie de **votação**, onde o algoritmo elegerá os pontos do espaço de parâmetros com maior probabilidade de representar uma reta na imagem original (i.e. pontos de máximo).

A Figura 34 mostra um exemplo de aplicação da transformada de Hough:

Figura 34 – Exemplo de aplicação da transformada de Hough.

[ESQ] Imagem Original [DIR] Resultado da aplicação da transformada de Hough.



Fonte: Adaptado de (MAINTZ, 2005)

À esquerda tem-se uma imagem de uma figura geométrica qualquer, formada por linhas retas. À direita está o resultado da aplicação da transformada de Hough: cada pixel da imagem original correspondente a uma borda gera uma senóide no espaço de parâmetros.

Analisando os resultados, é possível observar a existência de 8 pontos de máximo no espaço de parâmetros, o que indica que todos os segmentos de linha da imagem original estão contidos em 8 linhas retas.

### 4.5.3 Detecção de Círculos

A detecção de formas circulares pode ser utilizada em diversas aplicações industriais, como em sistemas automáticos de inspeção ou de montagem (DAVIES, 2012).

Na indústria de alimentos, por exemplo, inúmeros produtos como biscoitos, bolos, pizzas e frutas possuem formato arredondado. Na indústria automotiva, por sua vez, os círculos podem ser encontrados em componentes diversos, como rodas, arruelas, pistões e cabeças de parafuso, bem como furos redondos de vários tamanhos e funções.

Do ponto de vista conceitual, a ideia de utilizar a transformada de Hough para detecção de círculos é semelhante à ideia apresentada para detecção de linhas. A diferença primordial advém da equação que descreve o círculo no espaço cartesiano  $(x, y)$ , dada por:

$$(x - x_0)^2 + (y - y_0)^2 = R^2 \quad (4.8)$$

Onde,

$(x_0, y_0)$ : Coordenadas do centro do círculo.

$R$ : Raio do círculo.

Nesse caso, o fato da equação possuir 3 parâmetros dará origem a um **espaço de parâmetros 3D**, ou seja:

$$(x, y) \xrightarrow{\mathcal{T}_H} (x_0, y_0, R) \quad (4.9)$$

Na prática, a utilização de uma abordagem baseada em um espaço de parâmetros com mais de duas dimensões se revela extremamente custosa do ponto de vista computacional, demandando uma quantidade de memória e poder de processamento proibitivos para a maior parte das aplicações.

Desse modo, para contornar esse tipo de problema, uma série de artifícios e modificações costumam ser feitas. Um exemplo de modificação simples e eficaz é implementada quando o raio do círculo a ser detectado é conhecido, permitindo a eliminação de uma das dimensões do espaço de parâmetros (DAVIES, 2012).

#### 4.5.4 Detecção de Elipses

Devido à uma diferença de alinhamento entre a câmera e o plano do objeto, muitas vezes os objetos circulares são vistos obliquamente, aparentando ser elipses. Em outros casos, os objetos a serem detectados realmente são elípticos (e.g. mesas, janelas, piscinas). Sendo assim, a utilização de algoritmos para detecção de elipses também é um caso relativamente comum na indústria (DAVIES, 2012).

A detecção de elipses pode parecer ligeiramente mais complexa que a detecção de círculos, considerando apenas a adição de mais um parâmetro. Contudo, a excentricidade destrói a simetria do círculo, de modo que a direção do eixo principal também tem de ser definida (o que resulta numa equação com 5 parâmetros).

A equação de uma elipse no espaço cartesiano  $(x, y)$  é semelhante à de um círculo, sendo dada por:

$$\frac{(x')^2}{a^2} + \frac{(y')^2}{b^2} = 1 \quad \therefore \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \cdot \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} \quad (4.10)$$

Onde,

$(x_0, y_0)$ : Coordenadas do centro da elipse.

$a$ : Semieixo maior.

$b$ : Semieixo menor.

$\theta$ : Ângulo de rotação do eixo maior.

Nesse caso, o fato da equação possuir 5 parâmetros dará origem a um **espaço de parâmetros 5D**, ou seja:

$$(x, y) \xrightarrow{\mathcal{T}_H} (x_0, y_0, a, b, \theta) \quad (4.11)$$

Se, no caso do círculo, trabalhar em um espaço de parâmetros de 3 dimensões já era algo custoso do ponto de vista computacional, pensar em utilizar um espaço com 5 dimensões se torna altamente inviável. Felizmente, assim como no caso anterior existe uma série de artifícios disponíveis para aumentar a velocidade de execução do algoritmo e diminuir os requisitos de processamento e memória, como os sugeridos em (DAVIES, 2012) e (XIE; JI, 2002).

#### 4.5.5 Detecção de Formas Arbitrárias

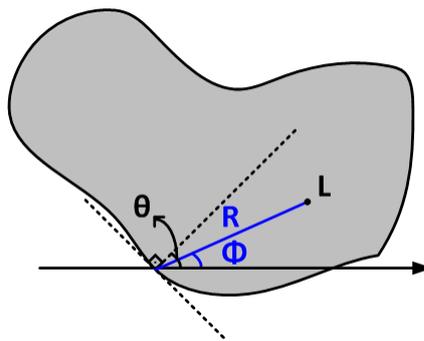
Baseado no exposto a respeito da transformada de Hough, é de se esperar que a técnica possa ser aplicada à qualquer **curva parametrizável**, bastando que sejam feitas algumas adaptações<sup>5</sup>.

Entretanto, aplicações reais podem possuir formas geométricas das mais diversas, incluindo curvas **curvas não-parametrizáveis** (i.e. formas arbitrárias). Visando resolver essa necessidade, foi desenvolvida a **Transformada de Hough Generalizada** (em inglês, GHT).

O princípio de funcionamento da GHT é o seguinte: primeiramente, a partir do *template* da forma arbitrária que se deseja detectar, escolhe-se um ponto de localização  $L$  (e.g. o centróide da forma).

A Figura 35 fornece um exemplo de uma forma geométrica arbitrária para definição dos parâmetros da GHT:

Figura 35 – Representação de uma forma geométrica arbitrária para definição dos parâmetros da GHT



Fonte: Adaptado de (DAVIES, 2012)

Em seguida, para cada pixel de borda da imagem, marca-se um ponto na matriz de acumulação à uma distância  $R(\theta)$ , na direção  $\phi(\theta)$  (onde  $\theta$  é a direção da normal à borda em análise). Esse procedimento irá gerar, na matriz de acumulação, um ponto de máximo localizado em  $L$ .

As funções  $R(\theta)$  e  $\phi(\theta)$  podem ser programadas no algoritmo ou, no caso de formas completamente arbitrárias, armazenadas em *look-up tables* (LUTs).

<sup>5</sup>Além disso, pode-se afirmar que a quantidade de dimensões da matriz de acumulação é igual à quantidade de parâmetros da curva que se deseja procurar em uma imagem.

#### 4.5.6 Variações da Transformada de Hough

Apesar de ser bastante robusto, o método da transformada de Hough é extremamente custoso em termos de quantidade de memória e poder de processamento, o que acaba tornando-o proibitivo em muitas aplicações.

Para contornar essas limitações, uma série de variações de implementação da transformada foram desenvolvidas, visando sobretudo o aumento da velocidade de execução do algoritmo e a redução da memória necessária (tornando-o mais adequado à aplicações em tempo-real).

Dentre as diversas variações da técnica, pode-se destacar as seguintes:

##### **Aleatorização/Randomização (*Randomization*)**

A **aleatorização** se baseia na ideia de que um pequeno subconjunto dos pontos de bordas, selecionados aleatoriamente, podem ser suficientes como entrada para o algoritmo.

De acordo com [Kiryati, Eldar e Bruckstein \(1991\)](#), essa modificação é capaz de reduzir significativamente o tempo de execução, ao custo de uma redução mínima em termos de precisão.

##### **Ponderação com base no Gradiente (*Gradient Weighting*)**

Tradicionalmente, ao construir a matriz de acumulação, o “voto” de cada pixel de borda possui o mesmo peso, independentemente do contraste da borda na imagem original (Ponderação Uniforme ou *Uniform Weighting*).

Uma maneira de melhorar a precisão do método é adicionar essa informação do contraste ao algoritmo, construindo um sistema de ponderação baseado na intensidade do gradiente ([DAVIES, 2012](#)).

##### **Redução da Dimensão da Matriz de Acumulação**

A construção de uma matriz de acumulação gera 2 problemas de ordem prática:

1. A necessidade de uma zona de memória muito grande;
2. Um custo computacional significativo (oriundo do processo de busca dos pontos de máximo dessa matriz).

Uma alternativa proposta por [Davies \(2012\)](#) é **dividir uma matriz de acumulação  $nD$  em  $n$  matrizes  $1D$** .

No caso da detecção de linhas, por exemplo, primeiramente se constrói a matriz de acumulação referente à um dos parâmetros (i.e.  $\rho$  ou  $\theta$ ) e realiza-se uma busca pelos pontos de máximo dessa matriz. Em seguida, com base nos resultados encontrados, constrói-se a 2ª matriz de acumulação e realiza-se outra busca pelos pontos de máximo.

### Utilização de Restrições Geométricas

Conforme descrito no Tópico 4.5.4, o elevadíssimo custo computacional que surge com a aplicação do método da transformada de Hough à uma curva descrita por vários parâmetros torna a utilização direta do método inviável.

Em alguns casos é possível reduzir a complexidade do problema e, conseqüentemente, seu custo computacional, através de restrições geométricas definidas com base na forma procurada.

A detecção de elipses, por exemplo, normalmente se basearia na determinação de 5 parâmetros diferentes. Entretanto, para contornar esse problema, Xie e Ji (2002) sugere a utilização de uma técnica baseada apenas na detecção do eixo maior, reduzindo o espaço de parâmetros de  $5D$  para  $1D$ .

Como será visto no Capítulo 5, algumas das modificações listadas acima foram implementadas no algoritmo desenvolvido, enquanto outras foram deixadas como sugestões de melhorias para os trabalhos posteriores.

# 5 Desenvolvimento da Pesquisa

## 5.1 Introdução

O objetivo deste capítulo é expor as principais etapas do desenvolvimento do trabalho, abordando desde a escolha dos métodos de segmentação de imagens até a análise dos resultados obtidos.

A 1ª seção introduz as características do problema, de modo a subsidiar a escolha dos métodos de segmentação de imagens adotados. A 2ª seção descreve os diferentes cenários de simulação utilizados. As seções 3, 4 e 5 dizem respeito ao detalhamento de um dos algoritmos utilizados, incluindo uma breve discussão de seus resultados. Finalmente, a última seção é dedicada à comparação dos resultados obtidos.

## 5.2 Escolha dos Métodos de Segmentação de Imagens

Infelizmente, ainda não existe panacéia para a segmentação de imagens (i.e. um método que resolva qualquer problema). Sendo assim, em todo projeto de sistema de visão computacional é necessário avaliar quais os métodos de segmentação mais adequados à aplicação.

O 1º passo desse processo é analisar as características do problema, que servirão de guia para a escolha dos métodos de segmentação a serem testados (essas etapas serão descritas a seguir, nos Tópicos 5.2.1 e 5.2.2).

Finalmente, após a etapa de implementação dos métodos escolhidos, realiza-se os respectivos testes e avalia-se os resultados obtidos.

### 5.2.1 Características do Problema

Conforme exposto no Capítulo 1, o objetivo a longo prazo do projeto é *automatizar o processo de detecção, classificação e retirada dos casulos que não atendam aos critérios de qualidade (e.g. tamanho, forma, cor e textura dentro de um padrão pré-estabelecido)*.

Apesar do recorte deste trabalho se limitar à etapa de segmentação de imagens, é essencial que o mesmo considere as características do problema como um todo.

Nesse contexto, a Figura 36 mostra uma imagem obtida da linha de produção e outra feita na bancada do laboratório e utilizada no desenvolvimento dos algoritmos.

Figura 36 – Exemplos de imagens dos casulos do bicho-da-seda



Fonte: Autoria própria

Com base nas imagens obtidas e nas demais informações do processo, foram identificadas as seguintes características do problema:

### Objetos

- **Tipo:** Casulos do Bicho-da-Seda;
- **Cor:** Brancos (Casulos Bons);  
Brancos com Tons de Cinza / Tons de Marrom (Casulos Defeituosos);
- **Forma:** Elipsóide;
- **Dimensões (Eixo Maior x Eixo Menor):** ~ 40 x 25 mm
- **Textura:** Peculiar;
- **Posição Relativa:** Separados / Se tocando / Sobrepostos;

### Plano de Fundo

- **Tipo:** Mesa / Esteira da Linha de Produção;
- **Cor:** Escura;
- **Textura:** Discreta / Inexistente;

### Tipo de Inspeção

- **Online x Offline:** Depende do local onde será feita a inspeção (i.e. numa mesa de trabalho ou na esteira da linha de produção);
- **100 % dos Casulos x Amostras Aleatórias:** A definir junto ao cliente;

## 5.2.2 Métodos Escolhidos

À luz das características do problema e dos diversos métodos de segmentação de imagens disponíveis na literatura, optou-se por implementar 3 métodos:

### **Segmentação pelo Método do *Threshold***

Por ser um método simples (tanto conceitualmente quanto em termos de implementação) e de baixo custo computacional, sua implementação torna-se quase que obrigatória durante a etapa inicial de avaliação dos métodos de segmentação de imagens mais adequados à uma determinada aplicação.

Segundo [Maintz \(2005\)](#), provavelmente é a técnica de segmentação de imagens mais utilizada atualmente. É a escolha óbvia no início de um projeto e costuma ser adotada como solução final, caso seu desempenho seja satisfatório.

No contexto deste trabalho, o forte contraste entre a cor branca dos casulos e a cor escura do plano de fundo (vide Figura 36) sugere que o histograma da imagem seja do tipo **bimodal** (i.e. possuirá 2 picos), o que aumenta a viabilidade da segmentação por *threshold*.

### **Segmentação pelo Método do *Watershed***

Apesar de ser uma técnica mais complexa e custosa (do ponto de vista computacional) do que a do *threshold*, também é bastante utilizada, pois possui a grande vantagem de conseguir separar objetos que se tocam (um dos cenários que acontece na prática e, conseqüentemente, será analisado neste trabalho).

### **Segmentação pelo Método da Transformada de Hough**

Técnica bastante robusta, porém complexa e de custo computacional elevado. A robustez dessa técnica é capaz de mitigar efeitos indesejáveis provocados por fenômenos como oclusão, ruído e outros artefatos.

A transformada de Hough costuma ser utilizada em aplicações onde os objetos possuem formas específicas conhecidas e, preferencialmente, simples (i.e. que podem ser representadas com poucos parâmetros).

Sendo assim, a forma elipsóide e bem definida dos casulos pode funcionar como uma informação importante para a etapa de segmentação.

A ordem em que os métodos foram apresentados acima indica um aumento da **robustez** do algoritmo, ao preço de uma maior **complexidade** e de um maior **custo computacional** do mesmo.

Em suma, a metodologia adotada foi a de implementar 3 técnicas de características distintas e comparar os resultados obtidos (onde a escolha das técnicas também levou em consideração as aplicações semelhantes encontradas na literatura).

### 5.3 Metodologia & Cenários de Teste

O objetivo principal dessa seção é detalhar alguns aspectos de ordem prática da realização deste trabalho, de acordo com a metodologia definida no Tópico 1.8 - Aspectos Metodológicos.

O 1º passo do projeto consistiu em entender melhor o problema e definir o escopo do trabalho. Esta fase inicial contou com o apoio da empresa **Fiação de Seda BRATAC S/A**, que disponibilizou o seguinte material:

- Um vídeo da sua linha de produção em operação;
- Dois conjuntos de casulos selecionados manualmente, sendo um conjunto de casulos bons (98 unidades) e outro de casulos defeituosos (107 unidades);
- Uma imagem do quadro de tipos de defeitos comuns em casulos (utilizada como referência no controle de qualidade interno feito pela empresa).

Em seguida, foi necessário obter um conjunto de imagens dos casulos do bicho-da-seda, que fosse representativo, para utilização nos algoritmos.

Inicialmente cogitou-se a ideia de trabalhar diretamente com os *frames* do vídeo recebido, mas esta hipótese foi abandonada devido à variação constante de características da imagem ao longo da gravação (e.g. iluminação, distância de trabalho, enquadramento), o que aumentaria desnecessariamente a complexidade do problema.

De modo a contornar esse problema, optou-se por utilizar os conjuntos de casulos recebidos para recriar, na bancada do laboratório, condições semelhantes às da linha de produção (vide Figura 36).

Essa abordagem, de utilizar imagens obtidas em um ambiente controlado para desenvolver algoritmos que funcionarão em um ambiente real, pode ser considerada natural, visto que, numa aplicação industrial, o projeto do sistema deve incluir mecanismos que favoreçam o bom funcionamento do algoritmo e aumente a robustez da aplicação.

Para testar o desempenho dos métodos escolhidos, foram criados 3 cenários distintos em que cada um dos algoritmos desenvolvidos foi aplicado:

### **Cenário 01 - Casulos Separados**

Os casulos se encontram distantes uns dos outros, de modo a não se tocarem em hipótese alguma.

Como será visto posteriormente, esta configuração corresponde à condição ideal para o funcionamento dos algoritmos.

### **Cenário 02 - Casulos se Tocando**

Neste cenário, alguns casulos estão encostados uns nos outros (o que dificulta o processo de segmentação).

*A priori* pode parecer uma modificação desprezível em relação ao cenário anterior, mas esse detalhe pode arruinar o funcionamento de alguns métodos.

### **Cenário 03 - Casulos Sobrepostos**

Os casulos podem se tocar e até mesmo ficarem uns em cima dos outros.

No pior dos casos, quando os casulos estão sobrepostos, a detecção se torna praticamente inviável.

A Figura 37 mostra uma comparação entre os 3 cenários construídos:

Figura 37 – Exemplo de imagens de cada um dos cenários de teste escolhidos



Fonte: Autoria própria

Cada um dos conjuntos de casulos enviados pela empresa foi dividido em 4 subconjuntos (totalizando 4 subconjuntos de casulos bons e 4 subconjuntos de casulos defeituosos).

Em seguida, obteve-se uma imagem de cada subconjunto para cada um dos 3 cenários definidos acima, como mostram as Figuras 38 e 39.

Figura 38 – Imagens dos casulos bons utilizadas nos algoritmos

CASULOS BONS			
	CENÁRIO 01	CENÁRIO 02	CENÁRIO 03
SUBCONJUNTO 01 (23 CASULOS)	 <i>IMAGEM B01.01</i>	 <i>IMAGEM B01.02</i>	 <i>IMAGEM B01.03</i>
SUBCONJUNTO 02 (30 CASULOS)	 <i>IMAGEM B02.01</i>	 <i>IMAGEM B02.02</i>	 <i>IMAGEM B02.03</i>
SUBCONJUNTO 03 (20 CASULOS)	 <i>IMAGEM B03.01</i>	 <i>IMAGEM B03.02</i>	 <i>IMAGEM B03.03</i>
SUBCONJUNTO 04 (25 CASULOS)	 <i>IMAGEM B04.01</i>	 <i>IMAGEM B04.02</i>	 <i>IMAGEM B04.03</i>

Fonte: Autoria própria

Figura 39 – Imagens dos casulos defeituosos utilizadas nos algoritmos

CASULOS DEFEITUOSOS			
	CENÁRIO 01	CENÁRIO 02	CENÁRIO 03
SUBCONJUNTO 01 (23 CASULOS)	 <p>IMAGEM D01.01</p>	 <p>IMAGEM D01.02</p>	 <p>IMAGEM D01.03</p>
SUBCONJUNTO 02 (26 CASULOS)	 <p>IMAGEM D02.01</p>	 <p>IMAGEM D02.02</p>	 <p>IMAGEM D02.03</p>
SUBCONJUNTO 03 (31 CASULOS)	 <p>IMAGEM D03.01</p>	 <p>IMAGEM D03.02</p>	 <p>IMAGEM D03.03</p>
SUBCONJUNTO 04 (27 CASULOS)	 <p>IMAGEM D04.01</p>	 <p>IMAGEM D04.02</p>	 <p>IMAGEM D04.03</p>

Fonte: Autoria própria

Após aplicar os algoritmos desenvolvidos em cada um dos cenários escolhidos, os resultados encontrados são expostos e analisados. Por último, é feita uma comparação entre as técnicas, com foco em 2 parâmetros de desempenho:

- A taxa de acertos;
- O custo computacional (i.e. tempo de execução).

O critério adotado para calcular a taxa de acertos considerou como “acerto” todo casulo que foi segmentado individualmente (o que caracteriza uma abordagem conservadora).

O custo computacional, por sua vez, foi medido através do comando *Run and Time* do MATLAB, que fornece o tempo de execução das diferentes funções utilizadas no algoritmo.

Para garantir a isonomia nas medições do tempo de execução, é imprescindível trabalhar sempre sob as mesmas condições. Dentro deste contexto, todas as simulações foram feitas na seguinte plataforma:

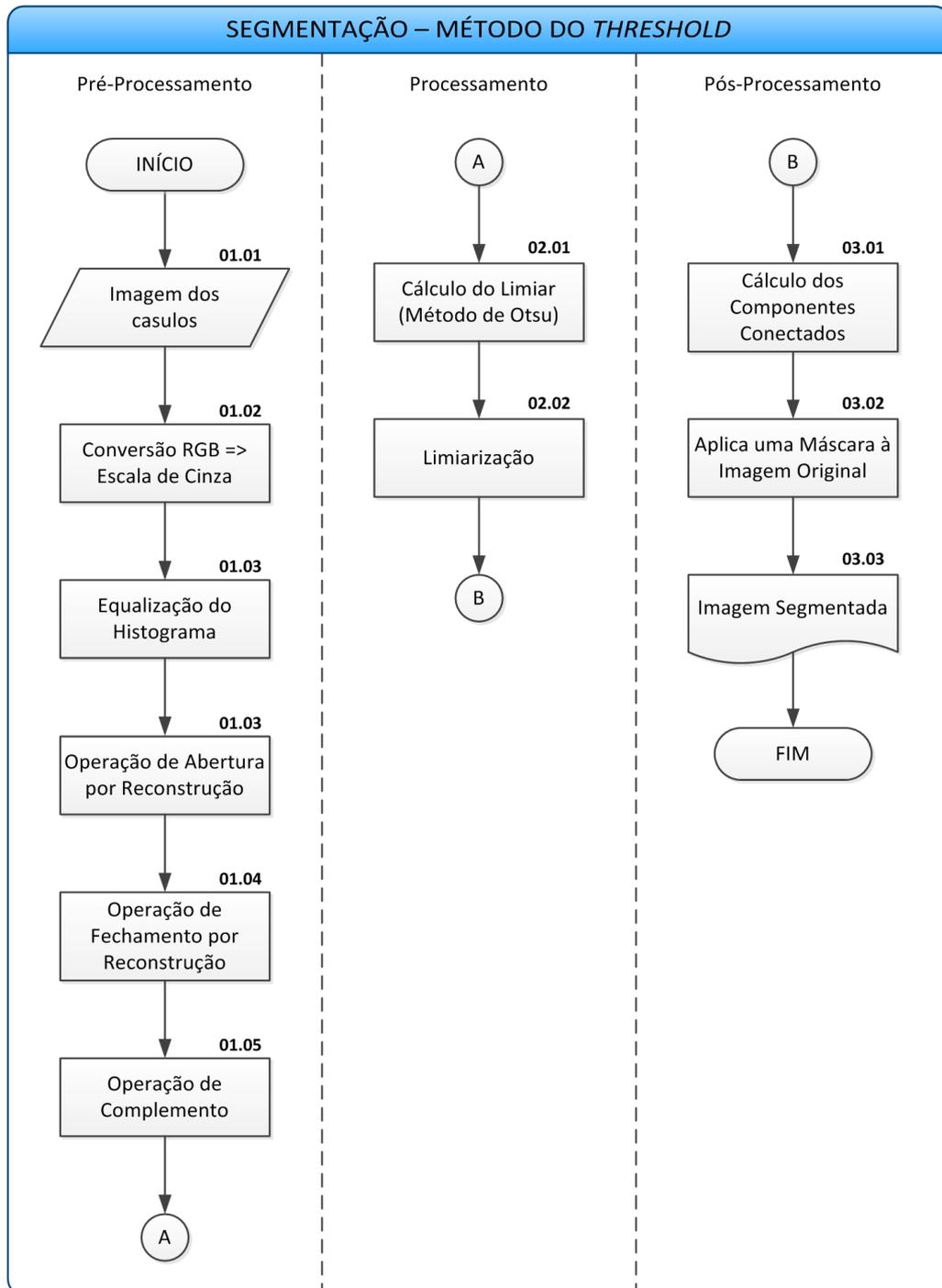
- Notebook DELL Latitude E6430
- Processador i7-3540M CPU @ 3 GHz x 2
- 8 GB de Memória RAM DDR3
- Sistema Operacional: Linux Mint 16 Cinnamon 64-bit
- MATLAB R2013a 64-bit
- Mesmos programas sendo executados em segundo plano

## 5.4 Segmentação pelo Método do *Threshold*

### 5.4.1 Algoritmo

A Figura 40 mostra o fluxograma do algoritmo implementado para realizar a segmentação pelo método do *threshold*:

Figura 40 – Fluxograma do algoritmo de segmentação - Método do *Threshold*



Fonte: Autoria própria

**Etapa 01.01**

Especifica o diretório e o nome do arquivo da imagem a ser segmentada e, em seguida, carrega a imagem na memória.

**Etapa 01.02 \***

Converte o espaço de cores de RGB para Escala de Cinza.

Como *a priori* a informação de cor não será utilizada na segmentação, é interessante realizar essa conversão para reduzir a quantidade de dados a ser processada.

**Etapa 01.03 \***

Aumenta o contraste da imagem através de uma equalização do histograma.

**Etapa 01.04 \***

Realiza uma operação de abertura-por-reconstrução (*Opening-by-Reconstruction*) utilizando um **elemento estruturante** em forma de disco.

Enquanto a operação de abertura clássica é implementada através de uma erosão seguida de uma dilatação, a abertura-por-reconstrução substitui a etapa de dilatação por uma reconstrução morfológica.

De uma maneira geral, as operações baseadas em reconstrução são mais eficientes em remover pequenas imperfeições do objeto a ser segmentado (sem afetar a forma do objeto).

**Etapa 01.05 \***

Realiza uma operação de fechamento-por-reconstrução (*Closing-by-Reconstruction*) utilizando um **elemento estruturante** em forma de disco.

**Etapa 01.06 \***

Substitui a intensidade de cada pixel por seu valor complementar:

$$I_{NEW} = |255 - I_{OLD}|$$

Esta etapa de correção é feita apenas para ajustar o resultado obtido à convenção utilizada pelo MATLAB (i.e. objetos brancos em um fundo preto).

**Etapa 02.01 †**

Calcula o valor do limiar através do método de Otsu.

**Etapa 02.02 †**

Compara o valor de cada pixel com o do limiar calculado anteriormente, de modo a transformar a imagem em binária.

Os pixels abaixo do limiar serão classificados como plano de fundo (cor preta) e os pixels acima desse limiar como objetos (cor branca).

---

\*Vide detalhes da operação no tópico [3.3.2](#).

†Vide detalhes da operação no tópico [5.4](#).

**Etapa 03.01**

Esta etapa calcula os diferentes **componentes conectados** presentes na imagem e suas respectivas características (e.g. coordenadas do centróide, circularidade).

**Etapa 03.02**

Aplica um nível de transparência à resposta encontrada (i.e. imagem segmentada) e, em seguida, a sobrepõe à imagem de entrada.

Esta etapa visa facilitar a visualização dos resultados através de uma comparação com a imagem original.

**Etapa 03.03**

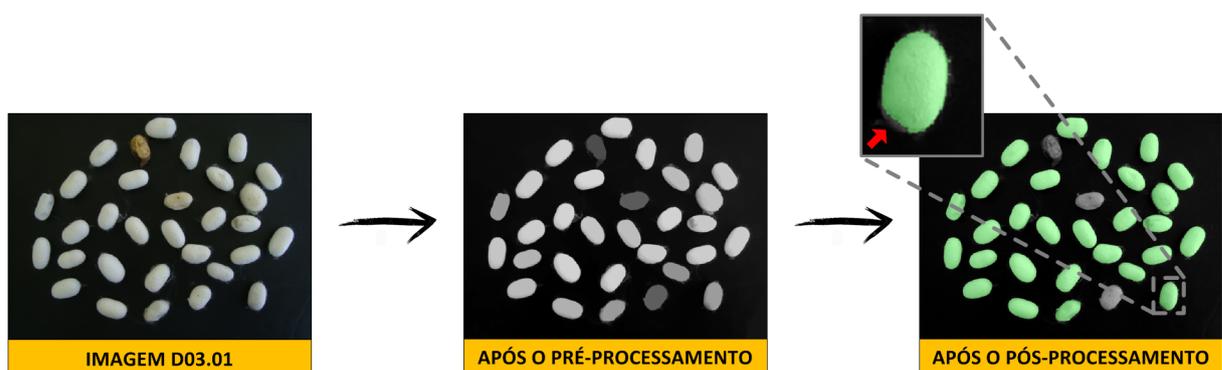
Exibe a imagem final após o processo de segmentação.

**5.4.2 Resultados & Discussão**

Apesar de não estar explícito no fluxograma da Figura 40, após a execução de cada etapa é exibida uma imagem de saída, o que permite acompanhar a evolução do processo de segmentação.

Um exemplo dessa evolução pode ser visto na Figura 41, que compara a imagem original com a mesma imagem após o pré-processamento e após o pós-processamento:

Figura 41 – Exemplo da evolução da imagem ao longo do processo de segmentação - Método do *Threshold*

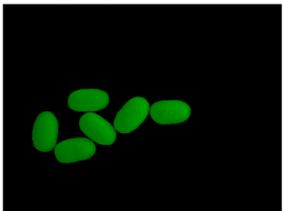
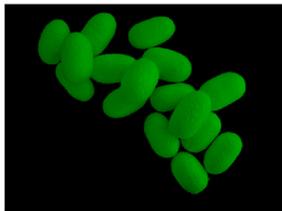
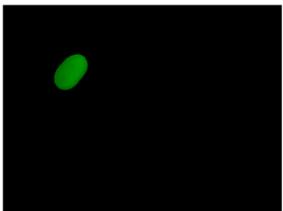
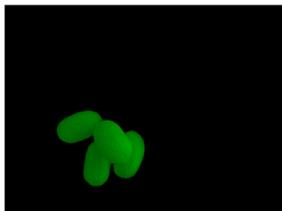
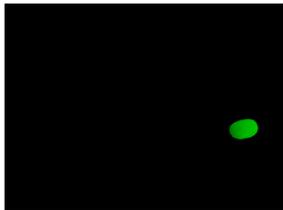
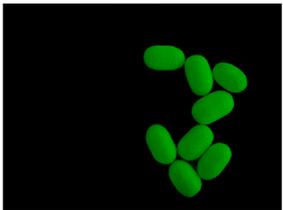
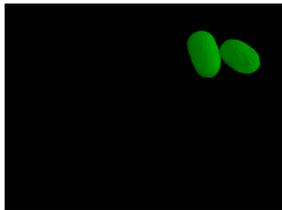


Fonte: Autoria própria

De modo a ilustrar o resultado final do algoritmo, que exibe uma imagem para cada objeto detectado, foram geradas as Figuras 42 e 43.

Como as figuras deixam claro, os resultados se referem à aplicação do algoritmo ao Subconjunto 01 de cada um dos tipos de casulos (i.e. bons e defeituosos) e para todos os 3 cenários definidos.

Figura 42 – Resultado da segmentação do Subconjunto 01 do conjunto de casulos bons - Método do *Threshold*

		CASULOS BONS		
		CENÁRIO 01	CENÁRIO 02	CENÁRIO 03
SUBCONJUNTO 01 (23 CASULOS)		 IMAGEM B01.01	 IMAGEM B01.02	 IMAGEM B01.03
OBJETO 01		 IMAGEM B01.01.01	 IMAGEM B01.02.01	 IMAGEM B01.03.01
OBJETO 02		 IMAGEM B01.01.02	 IMAGEM B01.02.02	 IMAGEM B01.03.02
•••		•••	•••	•••
OBJETO N		 IMAGEM B01.01.23	 IMAGEM B01.02.07	 IMAGEM B01.03.04

Fonte: Autoria própria

Figura 43 – Resultado da segmentação do Subconjunto 01 do conjunto de casulos defeituosos - Método do *Threshold*

		CASULOS DEFEITUOSOS		
		CENÁRIO 01	CENÁRIO 02	CENÁRIO 03
SUBCONJUNTO 01 (23 CASULOS)		 IMAGEM D01.01	 IMAGEM D01.02	 IMAGEM D01.03
OBJETO 01		 IMAGEM D01.01.01	 IMAGEM D01.02.01	 IMAGEM D01.03.01
OBJETO 02		 IMAGEM D01.01.02	 IMAGEM D01.02.02	 IMAGEM D01.03.02
•••		•••	•••	•••
OBJETO N		 IMAGEM D01.01.23	 IMAGEM D01.02.15	 IMAGEM D01.03.09

Fonte: Autoria própria

Os resultados encontrados evidenciam que o algoritmo desenvolvido está sujeito à basicamente 3 tipos de problemas:

#### **Falsos Negativos (vide Figura 41)**

Este problema ocorre quando o limiar escolhido não é suficiente para detectar todos os casulos da imagem.

Uma análise visual da imagem após o pré-processamento revela que é possível distinguir facilmente entre os casulos não-detectados e o plano de fundo, o que indica que o problema provavelmente pode ser resolvido através da mudança do método de escolha do limiar.

#### **Segmentação Incompleta (vide detalhe da Figura 41)**

A presença de uma região um pouco mais escura em um casulo, causada por uma sombra ou algum defeito de cor, pode fazer com que o objeto seja segmentado de maneira incompleta.

Para resolver o problema da sombra, pode-se realizar uma correção de iluminação (via *software* ou via *hardware*).

Quando o problema é gerado por defeitos de cor do próprio casulo, pode-se pensar em trabalhar com outro espaço de cores, que maximize a semelhança entre pixels do casulo e minimize a semelhança com os pixels do plano de fundo.

#### **Múltiplos Casulos Detectados como um Só (vide Figura 42)**

Apesar de aparentemente ser algo trivial, quando um ou mais casulos da imagem estão se tocando o processo de segmentação tem dificuldades em distinguir um objeto do outro.

Neste caso, o que acaba ocorrendo é que o grupo de casulos é considerado como se fosse um único objeto.

Além disso, nessas situações até mesmo alguns espaços entre casulos acabam sendo considerados como parte do objeto detectado.

A resolução desse problema é não-trivial, devendo envolver a implementação de algum método que evite que os objetos segmentados se toquem. No caso dos casulos sobrepostos, a correta segmentação pode se tornar impraticável.

A Tabela 3 traz uma compilação dos resultados encontrados para todas as imagens analisadas (conforme convenção de nomenclatura definida nas Figuras 38 e 39):

Tabela 3 – Resumo dos resultados da segmentação - Método do *Threshold*.

SEGMENTAÇÃO PELO MÉTODO DO <i>THRESHOLD</i>				
#	IMAGEM	TAXA DE ACERTOS		TEMPO DE EXECUÇÃO
		ABSOLUTA	RELATIVA	
01	B01.01	23/23	100,00 %	7,14 s
02	B01.02	3/23	13,04 %	3,17 s
03	B01.03	1/23	4,35 %	2,09 s
04	B02.01	30/30	100,00 %	8,49 s
05	B02.02	12/30	40,00 %	4,43 s
06	B02.03	5/30	16,67 %	2,62 s
07	B03.01	20/20	100,00 %	5,85 s
08	B03.02	8/20	40,00 %	1,79 s
09	B03.03	0/20	0 %	1,60 s
10	B04.01	25/25	100,00 %	6,81 s
11	B04.02	11/25	44,00 %	4,42 s
12	B04.03	4/25	16,00 %	2,55 s
13	D01.01	23/23	100,00 %	6,23 s
14	D01.02	10/23	43,48 %	4,44 s
15	D01.03	6/23	26,09 %	3,13 s
16	D02.01	26/26	100,00 %	6,50 s
17	D02.02	8/26	30,77 %	3,59 s
18	D02.03	6/26	23,08 %	3,50 s
19	D03.01	28/31	90,32 %	6,85 s
20	D03.02	3/31	9,68 %	2,35 s
21	D03.03	3/31	9,68 %	2,11 s
22	D04.01	27/27	100,00 %	6,75 s
23	D04.02	6/27	22,22 %	3,39 s
24	D04.03	6/27	22,22 %	2,81 s
MÉDIA POR CENÁRIO				
CENÁRIO 01		202/205	98,54 %	6,83 s
CENÁRIO 02		61/205	29,76 %	3,45 s
CENÁRIO 03		31/205	15,12 %	2,55 s

Fonte: Autoria própria

Como pode-se observar, para cada uma das imagens foram analisados dois parâmetros de desempenho:

- A taxa de acertos;
- O custo computacional (i.e. tempo de execução).

Ao analisar os resultados por cenário, fica evidente que o algoritmo funciona extremamente bem quando os casulos estão distantes uns dos outros, atingindo uma taxa de acertos próxima à 100 %.

No caso dos outros cenários a taxa de acertos decai bastante, principalmente em consequência da definição de acerto que foi adotada. Apesar de conseguir segmentar corretamente **um grupo** de casulos que estão em contato, o algoritmo não é capaz de, em seguida, segmentá-los individualmente, o que caracteriza um erro.

Em outras palavras, **qualquer casulo que não esteja isolado não será detectado corretamente.**

Em relação ao tempo de execução do algoritmo, os valores encontrados são relativamente pequenos, da ordem de poucos segundos. Obviamente que para uma aplicação em tempo real, como por exemplo uma inspeção feita na linha de produção, seria necessário reduzir esse tempo (mas esse não foi o foco do trabalho desenvolvido).

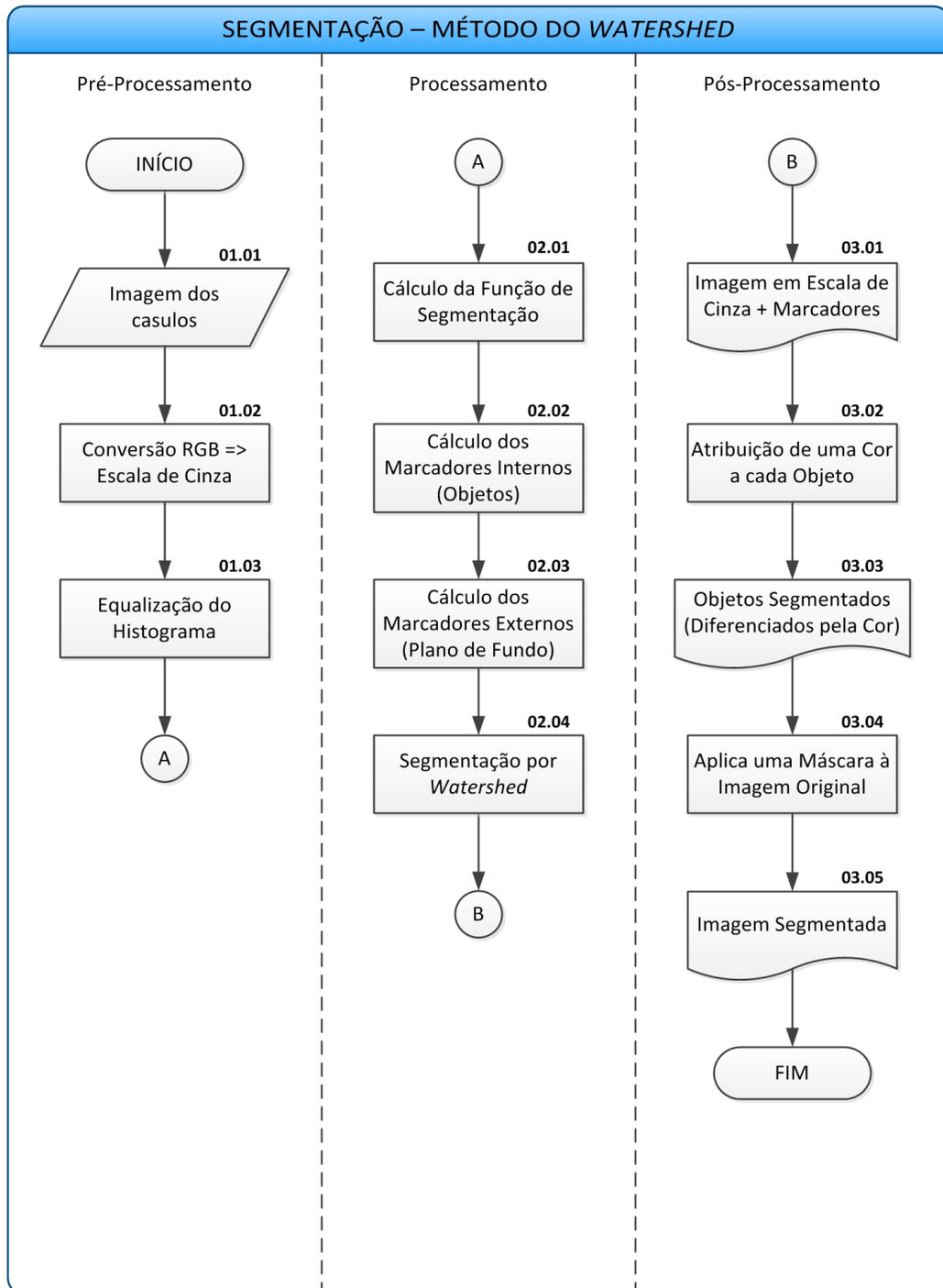
Finalmente, é interessante frisar que, de maneira geral, mais da metade do tempo de execução foi gasto com funções de exibição dos resultados, o que indica que há bastante espaço para otimização deste parâmetro de desempenho.

## 5.5 Segmentação pelo Método do *Watershed*

### 5.5.1 Algoritmo

A Figura 44 mostra o fluxograma do algoritmo implementado para realizar a segmentação pelo método do *watershed*:

Figura 44 – Fluxograma do algoritmo de segmentação - Método do *Watershed*



Fonte: Autoria própria

O algoritmo, desenvolvido com base na aplicação apresentada em (MATHWORKS, 2014b), segue as seguintes etapas:

#### **Etapas 01.01, 01.02 e 01.03 \***

Essas etapas são idênticas às etapas homônimas descritas para o caso do algoritmo de segmentação pelo método do *threshold*.

#### **Etapa 02.01 \***

Calcula a **função de segmentação** (que será utilizada no algoritmo juntamente com os marcadores).

A função de segmentação é uma versão modificada da imagem original, que visa garantir que o resultado do algoritmo do *watershed* corresponda às fronteiras entre os objetos.

Neste caso, a função de segmentação escolhida foi a **magnitude do gradiente** da imagem original.

#### **Etapa 02.02 \***

Calcula os marcadores internos (i.e. marcadores dos objetos) através de uma sucessão de sub-etapas baseadas em operações morfológicas:

1. Abertura-por-Reconstrução;
2. Fechamento-por-Reconstrução;
3. Localização dos Máximos Locais;
4. Eliminação dos Componentes Conectados abaixo de um determinado tamanho.

Como pode-se inferir a partir das sub-etapas acima, cada marcador obtido será um componente conectado interno à um único objeto.

#### **Etapa 02.03 \***

Calcula os marcadores externos (i.e. marcadores do plano de fundo) através de uma sucessão de sub-etapas:

1. Limiarização da Imagem de saída da Etapa 02.02.02;
2. Cálculo do Esqueleto por Zonas de Influência (*Skeleton by Influence Zones*).

Como pode-se inferir a partir das sub-etapas acima, os marcadores obtidos correspondem à um esqueleto do plano de fundo (i.e. linhas de 1 pixel de espessura, localizadas no centro de cada região do plano de fundo).

---

\*Vide detalhes da operação no tópico 3.3.2.

**Etapa 02.04** †

Realiza a segmentação da imagem através do método do *watershed*.

Para tal, utiliza-se a função de segmentação, os marcadores internos e os marcadores externos calculados nas etapas anteriores.

**Etapa 03.01**

Exibe a imagem original sobreposta pelos marcadores calculados.

**Etapa 03.02**

Utiliza uma escala de cores conhecida para atribuir uma cor a cada objeto identificado (incluindo o plano de fundo).

**Etapa 03.03**

Exibe a imagem segmentada, onde os objetos e o plano de fundo são diferenciados por meio de cores.

**Etapas 03.04 e 03.05**

Essas etapas são idênticas às etapas homônimas descritas para o caso do algoritmo de segmentação pelo método do *threshold*.

---

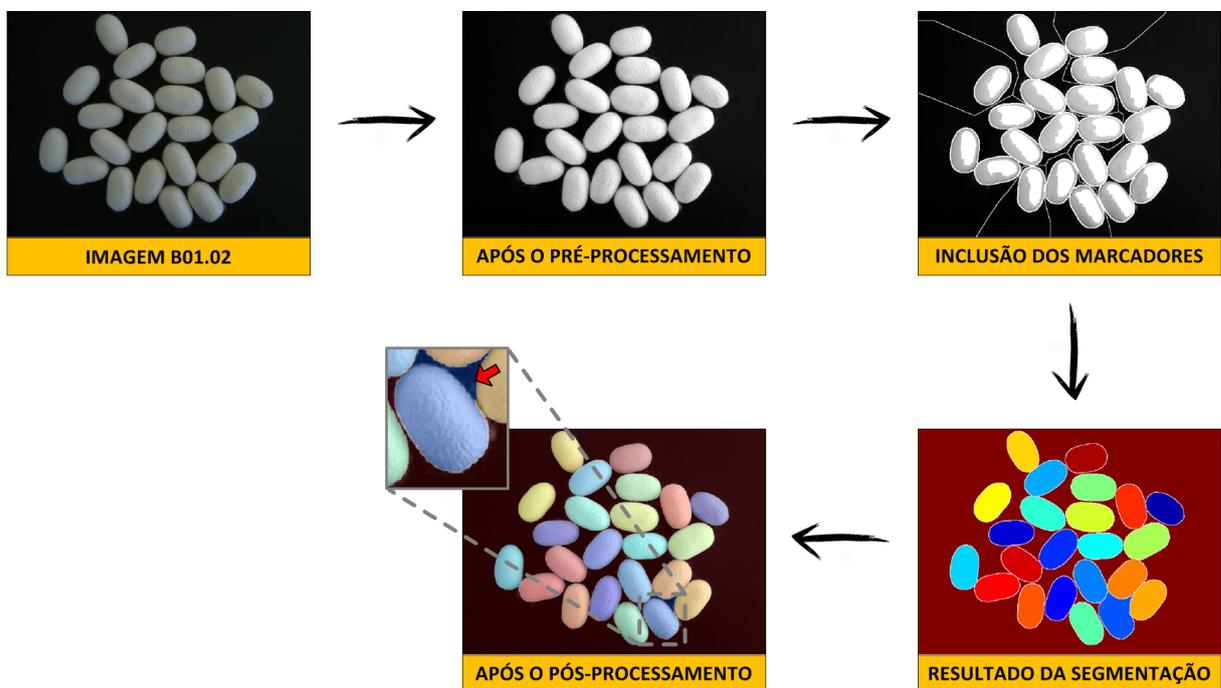
†Vide detalhes da operação no tópico [5.5](#).

### 5.5.2 Resultados & Discussão

Apesar de não estar explícito no fluxograma da Figura 44, após a execução de cada etapa é exibida uma imagem de saída, o que permite acompanhar a evolução do processo de segmentação.

Um exemplo dessa evolução pode ser visto na Figura 45, que compara a imagem original com a mesma imagem após o pré-processamento, após a inclusão dos marcadores, após a segmentação completa e após o pós-processamento:

Figura 45 – Exemplo da evolução da imagem ao longo do processo de segmentação - Método do *Watershed*

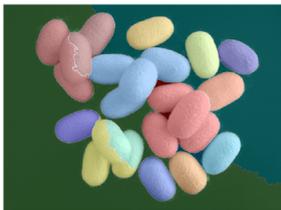
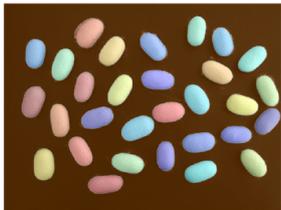
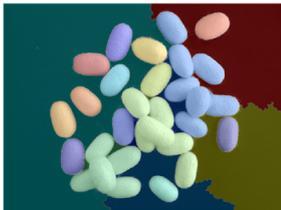


Fonte: Autoria própria

O resultado final do algoritmo é exposto nas Figuras 46 e 47.

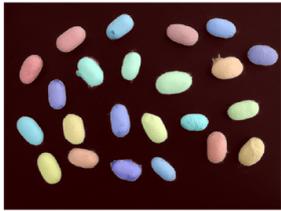
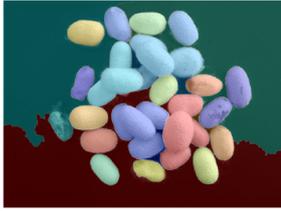
Como as figuras deixam claro, os resultados se referem à aplicação do algoritmo à todas as imagens de teste (conjunto de casulos bons e defeituosos, para todos os 3 cenários definidos).

Figura 46 – Resultado da segmentação do conjunto de casulos bons - Método do *Watershed*

CASULOS BONS			
	CENÁRIO 01	CENÁRIO 02	CENÁRIO 03
SUBCONJUNTO 01 (23 CASULOS)	 IMAGEM B01.01	 IMAGEM B01.02	 IMAGEM B01.03
SUBCONJUNTO 02 (30 CASULOS)	 IMAGEM B02.01	 IMAGEM B02.02	 IMAGEM B02.03
SUBCONJUNTO 03 (20 CASULOS)	 IMAGEM B03.01	 IMAGEM B03.02	 IMAGEM B03.03
SUBCONJUNTO 04 (25 CASULOS)	 IMAGEM B04.01	 IMAGEM B04.02	 IMAGEM B04.03

Fonte: Autoria própria

Figura 47 – Resultado da segmentação do conjunto de casulos defeituosos - Método do *Watershed*

CASULOS DEFEITUOSOS			
	CENÁRIO 01	CENÁRIO 02	CENÁRIO 03
SUBCONJUNTO 01 (23 CASULOS)	 IMAGEM D01.01	 IMAGEM D01.02	 IMAGEM D01.03
SUBCONJUNTO 02 (26 CASULOS)	 IMAGEM D02.01	 IMAGEM D02.02	 IMAGEM D02.03
SUBCONJUNTO 03 (31 CASULOS)	 IMAGEM D03.01	 IMAGEM D03.02	 IMAGEM D03.03
SUBCONJUNTO 04 (27 CASULOS)	 IMAGEM D04.01	 IMAGEM D04.02	 IMAGEM D04.03

Fonte: Autoria própria

Os resultados encontrados evidenciam que o algoritmo desenvolvido está sujeito à basicamente 5 tipos de problemas:

#### **Falsos Negativos (vide Figura 47)**

Uma análise visual das imagens ao longo da evolução do algoritmo revela que a ocorrência de falsos negativos está associada a 2 fatores:

1. Falha do algoritmo em gerar um marcador para algum dos objetos da imagem;
2. Escolha inadequada do limiar utilizado na Etapa 02.03.01 (neste caso, apesar dos marcadores existirem, eles acabam se fundindo com o marcador do plano de fundo).

Na 1ª situação, é necessário examinar, caso a caso, em qual etapa que antecede a geração dos marcadores está o problema.

Na 2ª situação, o problema provavelmente pode ser resolvido através da mudança do método de escolha do limiar.

#### **Segmentação Incompleta (vide Figura 47)**

A presença de uma região um pouco mais escura em um casulo, causada por uma sombra ou algum defeito de cor, pode fazer com que o objeto seja segmentado de maneira incompleta (e.g. resultado da imagem D02.01).

Em ambos os casos, a resolução do problema segue o mesmo raciocínio utilizado no caso da segmentação por *threshold*

#### **Múltiplos Casulos Detectados como um Só (vide Figura 46)**

Apesar de aparentemente ser algo trivial, quando um ou mais casulos da imagem estão se tocando o processo de segmentação tem dificuldades em distinguir um objeto do outro.

Neste caso, o que acaba ocorrendo é que o grupo de casulos é considerado como se fosse um único objeto.

Conforme exposto na Seção 5.5, devido a seu *modus operandi* o método do *watershed* é bastante eficiente em reduzir a ocorrência desse tipo de problema.

Para melhorar ainda mais o desempenho do algoritmo, é necessário aprimorar a etapa de geração de marcadores, evitando que haja falha na geração de um marcador ou fusão de dois marcadores ao longo das etapas.

No pior caso, quando os casulos estão sobrepostos, a solução do problema continua sendo semelhante, porém mais complexa de se obter.

**Plano de Fundo classificado como parte do Objeto (vide detalhe da Figura 45)**

Em algumas situações, o espaço entre casulos acaba sendo considerado como parte de um dos objetos detectados.

Esse problema é ocasionado quando há a ocorrência de 2 erros sucessivos:

1. Preenchimento do espaço entre casulos devido à Etapa 02.02.01;
2. Escolha inadequada do limiar utilizado na Etapa 02.03.01.

Na 1ª situação, é necessário examinar a influência do tamanho/forma do elemento estruturante utilizado, de modo a tentar minimizar a ocorrência desse tipo de problema.

Na 2ª situação, o problema provavelmente pode ser resolvido através da mudança do método de escolha do limiar.

**Sobre-Segmentação do Plano de Fundo (vide Figura 46)**

A sobre-segmentação do plano de fundo ocorre quando o algoritmo gera mais de um componente conectado associado ao plano de fundo da imagem.

As causas desse fenômeno não foram identificadas, exigindo uma análise mais aprofundada do algoritmo.

Entretanto, uma solução prática para o problema pode ser feita através da fusão dos componentes conectados acima de um determinado tamanho (visto que o tamanho dos componentes conectados associados ao plano de fundo são muito maiores que os associados aos objetos).

A Tabela 4 traz uma compilação dos resultados encontrados para todas as imagens analisadas (conforme convenção de nomenclatura definida nas Figuras 38 e 39):

Tabela 4 – Resumo dos resultados da segmentação - Método do *Watershed*.

SEGMENTAÇÃO PELO MÉTODO DO <i>WATERSHED</i>				
#	IMAGEM	TAXA DE ACERTOS		TEMPO DE EXECUÇÃO
		ABSOLUTA	RELATIVA	
01	B01.01	23/23	100,00 %	3,94 s
02	B01.02	22/23	95,65 %	4,08 s
03	B01.03	7/23	30,43 %	3,76 s
04	B02.01	30/30	100,00 %	3,63 s
05	B02.02	26/30	86,67 %	3,93 s
06	B02.03	11/30	36,67 %	3,48 s
07	B03.01	20/20	100,00 %	3,76 s
08	B03.02	20/20	100,00 %	3,60 s
09	B03.03	12/20	60,00 %	3,67 s
10	B04.01	25/25	100,00 %	3,76 s
11	B04.02	25/25	100,00 %	3,72 s
12	B04.03	9/25	36,00 %	3,49 s
13	D01.01	23/23	100,00 %	4,25 s
14	D01.02	21/23	91,30 %	4,01 s
15	D01.03	11/23	47,83 %	4,29 s
16	D02.01	26/26	100,00 %	4,65 s
17	D02.02	21/26	80,77 %	4,75 s
18	D02.03	15/26	57,69 %	3,62 s
19	D03.01	28/31	90,32 %	3,67 s
20	D03.02	19/31	61,29 %	4,71 s
21	D03.03	11/31	35,48 %	3,98 s
22	D04.01	27/27	100,00 %	4,22 s
23	D04.02	23/27	85,19 %	4,24 s
24	D04.03	12/27	44,44 %	4,18 s
MÉDIA POR CENÁRIO				
CENÁRIO 01		202/205	98,54 %	3,99 s
CENÁRIO 02		177/205	86,34 %	4,13 s
CENÁRIO 03		88/205	42,93 %	3,81 s

Fonte: Autoria própria

Como pode-se observar, para cada uma das imagens foram analisados dois parâmetros de desempenho:

- A taxa de acertos;
- O custo computacional (i.e. tempo de execução).

Ao analisar os resultados por cenário, fica evidente que o algoritmo funciona extremamente bem quando os casulos estão distantes uns dos outros, atingindo uma taxa de acertos próxima à 100 %.

Os resultados obtidos para o caso em que os casulos se tocam também são bastante satisfatórios, visto que a taxa de acertos foi acima de 85 %.

Finalmente, para o cenário em que os casulos estão sobrepostos, a taxa de acertos atingida foi razoável, girando em torno de 40 %.

Em ambos os casos, é importante destacar que ainda existe espaço para otimização do algoritmo: uma melhoria nos resultados pode ser alcançada através do estudo e implementação das sugestões expostas para resolução dos problemas encontrados.

Em relação ao tempo de execução do algoritmo, os valores encontrados são relativamente pequenos, da ordem de poucos segundos. Deste modo, para uma aplicação em tempo real, como por exemplo uma inspeção feita na linha de produção, seria necessário reduzir esse tempo (mas esse não foi o foco do trabalho desenvolvido).

Finalmente, é interessante frisar que, de maneira geral, mais da metade do tempo de execução foi gasto com funções de exibição dos resultados, o que indica que há bastante espaço para otimização deste parâmetro de desempenho.

## 5.6 Segmentação pelo Método da Transformada de Hough

### 5.6.1 Modelagem do Problema

No Tópico 5.2.1, que analisa as características do problema, uma das informações que mais chama a atenção nas imagens é a forma geométrica peculiar dos casulos. Apesar disso, nenhum dos métodos de segmentação apresentados até o momento explora essa propriedade.

Visando aproveitar esta característica para auxiliar no processo de segmentação, foi desenvolvido um algoritmo baseado na transformada de Hough.

Conforme apresentado na Seção 4.5, a segmentação pelo método da transformada de Hough se baseia na detecção de formas geométricas definidas *a priori* (que podem variar desde simples retas até formas geométricas arbitrárias).

Numa análise visual das imagens, a primeira impressão é que os casulos podem ser representados por **elipses**. Entretanto, após a execução de alguns testes empíricos, pôde-se concluir que a forma geométrica que mais se aproxima dos casulos é, na verdade, um **retângulo de bordas arredondadas**.

De acordo com (WEISSTEIN, 2014), um retângulo de bordas arredondadas pode ser descrito através de uma curva conhecida como **superelipse**, cuja equação no espaço cartesiano  $(x, y)$  é dada por:

$$\left| \frac{x'}{a} \right|^n + \left| \frac{y'}{b} \right|^n = 1 \quad \therefore \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \cdot \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} \quad (5.1)$$

Onde,

$(x_0, y_0)$ : Coordenadas do centro da superelipse.

$a$ : Semieixo maior.

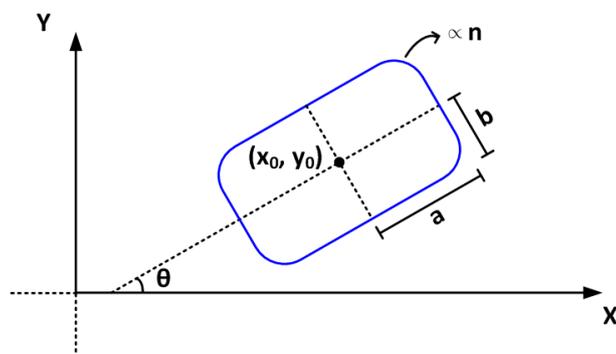
$b$ : Semieixo menor.

$\theta$ : Ângulo de rotação do eixo maior.

$n$ : Parâmetro relacionado ao grau de convexidade da curva.

A Figura 48 traz um exemplo de uma curva da família das superelipses, juntamente com a indicação de seus respectivos parâmetros:

Figura 48 – Exemplo de curva da família das superelipses (para  $n > 2$ )

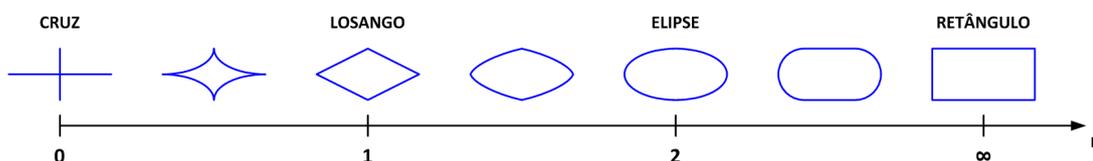


Fonte: Autoria própria

É interessante notar que a equação da superelipse nada mais é que uma generalização da equação clássica da elipse, onde o parâmetro  $n$  pode assumir qualquer valor positivo.

Mantendo os demais parâmetros constantes e aumentando progressivamente o valor de  $n$ , a curva da superelipse evolui conforme exposto na Figura 49:

Figura 49 – Influência do parâmetro  $n$  na curva da superelipse



Fonte: Autoria própria

A análise matemática das superelipses revela um problema em modelar os casulos a partir de retângulos de bordas arredondadas: o fato da equação possuir 6 parâmetros dá origem a um **espaço de parâmetros 6D**, ou seja:

$$(x, y) \xrightarrow{\mathcal{T}_{\mathcal{H}}} (x_0, y_0, a, b, \theta, n) \tag{5.2}$$

De modo a reduzir o tamanho do espaço de parâmetros, diminuindo o custo computacional, decidiu-se fixar o parâmetro  $n$  (trabalhando sempre com  $n = 3$ ).

A determinação do valor de  $n$  mais adequado à modelagem foi feita de maneira empírica. Para um dado casulo de referência, cujos demais parâmetros foram medidos manualmente, foi feita uma variação progressiva do valor de  $n$ . Em seguida, o valor mais adequado foi encontrado por meio de interpolações sucessivas.

### 5.6.2 Algoritmo

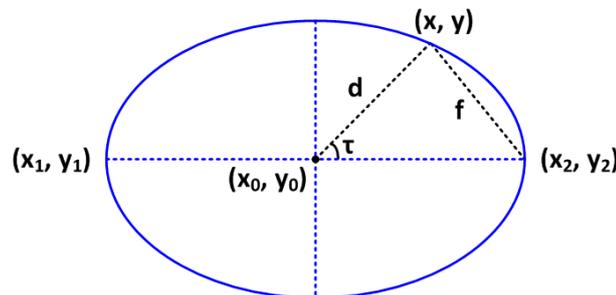
Ao fixar o valor de  $n$ , o espaço de parâmetros reduz-se de 6D para 5D. Apesar disso, utilizar uma transformada de Hough em um problema de 5 dimensões ainda é extremamente custoso do ponto de vista computacional (vide Tópico 4.5.4).

Para aumentar a velocidade de execução do algoritmo e diminuir os requisitos de processamento e memória, Xie e Ji (2002) desenvolveram um método de detecção de elipses baseado na transformada de Hough de apenas uma das dimensões.

A ideia geral do método é utilizar propriedades geométricas bem definidas da elipse para reduzir a quantidade de cálculo necessária à detecção. Em outras palavras, através dos vértices do eixo maior da elipse, pode-se calcular diretamente 4 entre os 5 parâmetros necessários. O parâmetro restante, então, pode ser determinado através de um processo de votação.

Primeiramente, realiza-se uma detecção de bordas na imagem original. Em seguida, considera-se que cada par de pixels  $(x_1, y_1)$ ,  $(x_2, y_2)$  corresponde aos 2 vértices do eixo maior de uma possível elipse (Figura 50).

Figura 50 – Definição dos pontos e parâmetros da elipse utilizados no algoritmo<sup>1</sup>



Fonte: Adaptado de (XIE; JI, 2002)

Conhecendo os vértices do eixo maior de uma possível elipse, pode-se calcular 4 dos 5 parâmetros necessários:

$$x_0 = (x_1 + x_2)/2$$

$$y_0 = (y_1 + y_2)/2$$

$$a = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}/2$$

$$\theta = \arctan[(y_2 - y_1)/(x_2 - x_1)] \tag{5.3}$$

<sup>1</sup>Para definição dos parâmetros da elipse, consultar Figura 48.

Enfim, para calcular o parâmetro restante, utiliza-se um terceiro ponto  $(x, y)$ . Deste modo, o semieixo menor pode ser estimado através da Equação 5.4:

$$b^2 = (a^2 d^2 \sin^2 \tau) / (a^2 - d^2 \cos^2 \tau) \quad \therefore \quad \cos \tau = (a^2 + d^2 - f^2) / (2ad) \quad (5.4)$$

Apesar do algoritmo de Xie e Ji (2002) ser dedicado à detecção de **elipses**, é de se esperar que o mesmo possa ser adaptado e estendido à detecção de **superelipses**.

Uma análise criteriosa das expressões matemáticas revela que:

- Para a Equação 5.3: As relações não sofrem alterações;
- Para a Equação 5.4: É necessário deduzir a nova expressão a partir da Equação 5.1:

$$\begin{aligned} \left| \frac{x}{a} \right|^3 + \left| \frac{y}{b} \right|^3 = 1 &\Rightarrow \left| \frac{b}{a} \right|^3 |x|^3 + |y|^3 = |b|^3 \\ &\Rightarrow \left| \frac{b}{a} \right|^3 |x|^3 - |b|^3 = -|y|^3 \\ &\Rightarrow |b|^3 \left( \left| \frac{x}{a} \right|^3 - 1 \right) = -|y|^3 \\ &\Rightarrow |b|^3 \left( \frac{|x|^3 - |a|^3}{|a|^3} \right) = -|y|^3 \\ &\Rightarrow |b|^3 = -\frac{|a|^3 |y|^3}{|x|^3 - |a|^3} \\ &\Rightarrow |b|^3 = \frac{|a|^3 |y|^3}{|a|^3 - |x|^3} \\ &\Rightarrow |b|^3 = \frac{|a \cdot y|^3}{|a|^3 - |x|^3} \\ &\Rightarrow \boxed{|b|^3 = \frac{|a \cdot d \cdot \sin \tau|^3}{|a|^3 - |d \cdot \cos \tau|^3}} \end{aligned} \quad (5.5)$$

Onde foram utilizadas as relações (vide Figura 50):

$$\begin{cases} x = \cos \tau \\ y = \sin \tau \end{cases}$$

O algoritmo desenvolvido neste trabalho utilizou como ponto de partida os algoritmos desenvolvidos em (SIMONOVSKY, 2013) (para detectar a elipse) e (LONG, 1998) (para desenhar a elipse detectada) e disponibilizados pelos respectivos autores na plataforma MATLAB Central.

Além das alterações necessárias para adaptar o algoritmo à detecção de superelipses, foram implementadas algumas melhorias no algoritmo original, dentre as quais as principais foram:

### **Visualização da Matriz de Acumulação**

É dada ao usuário a possibilidade de visualizar a matriz de acumulação.

### **Eliminação de Resultados Similares**

Para evitar que o algoritmo detecte mais de uma elipse referente a um único objeto, foi implementado um parâmetro que mede o grau de similaridade entre as elipses detectadas.

Deste modo, a versão atual do algoritmo calcula uma simples distância euclidiana entre os parâmetros da elipse em análise e as demais elipses, e, em seguida, compara essa distância à um **limiar de similaridade**, eliminando os resultados que são muito próximos.

### **Modos de Saída do Algoritmo**

No algoritmo original, o usuário precisa definir *a priori* quantas elipses deseja detectar na imagem. O algoritmo, então, fornece as  $n$  melhores elipses que forem encontradas, com base em um critério de pontuação que mede o quão próximo de uma elipse o resultado encontrado está.

O algoritmo proposto fornece ao usuário a possibilidade de optar por uma saída com base em uma pontuação mínima (*score*) definida pelo usuário. Nesse caso, não é necessário definir quantos objetos estão sendo procurados na imagem, o que caracteriza uma vantagem em muitas aplicações industriais.

A Figura 51 mostra o fluxograma do algoritmo implementado para realizar a segmentação pelo método da Transformada de Hough, cujas etapas são descritas abaixo.

### **Etapa 00**

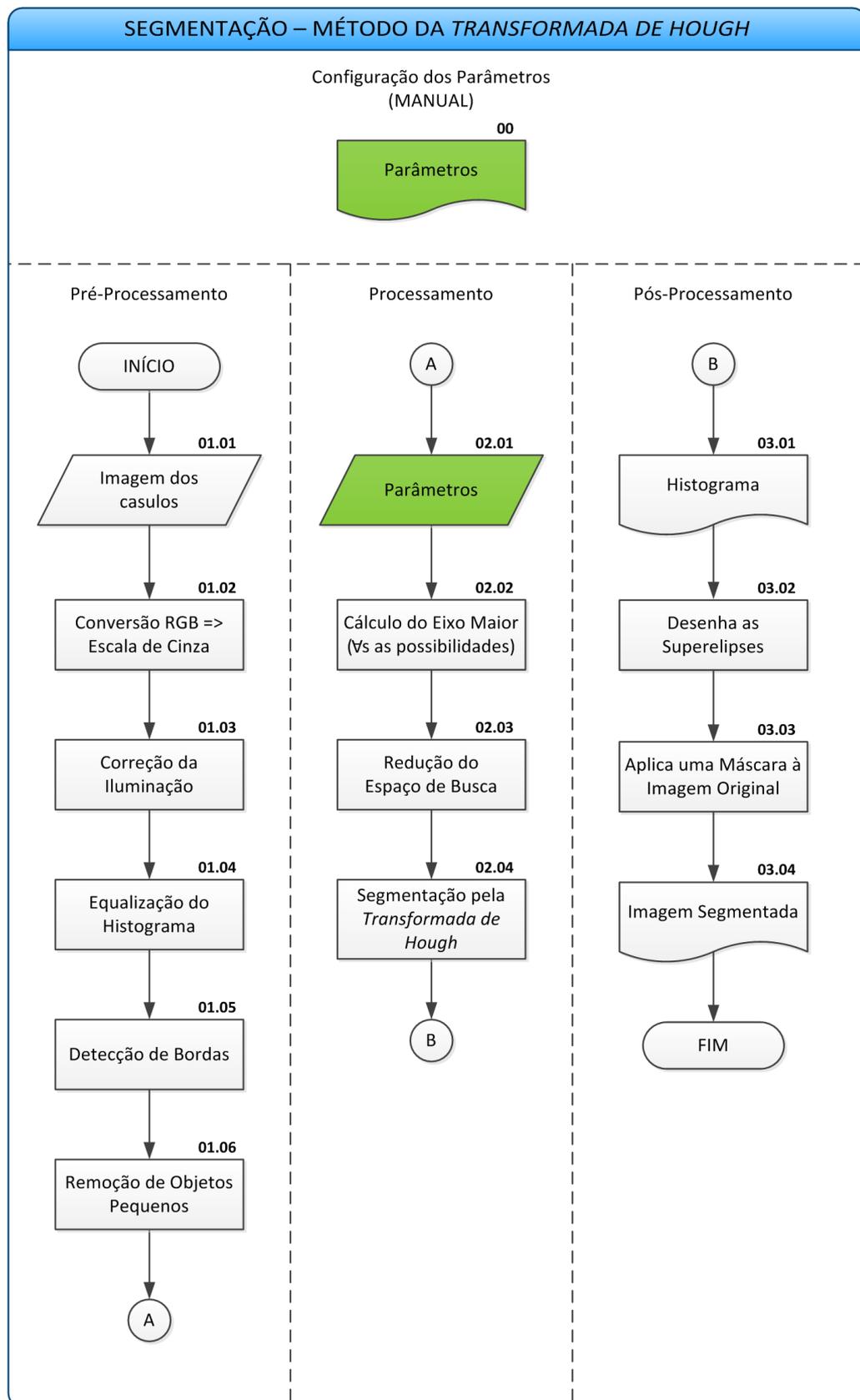
Para facilitar a modificação dos parâmetros de entrada do algoritmo de segmentação, foi desenvolvida uma rotina que salva os parâmetros configurados em um arquivo.

Toda vez que o algoritmo de segmentação é executado, faz-se uma busca por um arquivo com o mesmo nome, que contém os parâmetros necessários. Caso o arquivo não exista, é utilizada uma configuração padrão.

### **Etapas 01.01, 01.02 e 01.04 \***

Essas etapas são idênticas às etapas homônimas descritas para o caso do algoritmo de segmentação pelo método do *threshold*.

Figura 51 – Fluxograma do algoritmo de segmentação - Método da Transformada de Hough



Fonte: Autoria própria

**Etapa 01.03** \*

Corrige os efeitos provocados por uma iluminação não uniforme (através de um filtro do tipo *top hat*).

**Etapas 01.05 e 01.06** \*

Conforme exposto em (XIE; JI, 2002), para reduzir a quantidade de dados de entrada no algoritmo da transformada de Hough, focando apenas nas informações relevantes, o algoritmo deve trabalhar com as bordas da imagem original.

Este processo é realizado em 2 sub-etapas:

1. Detecção de Bordas através de um Operador de Sobel;
2. Eliminação dos Componentes Conectados abaixo de um determinado tamanho.

**Etapa 02.01**

Verifica se todos os parâmetros foram configurados pelo usuário e, em caso negativo, adota a configuração padrão.

**Etapa 02.02**

Considera que cada par de pixels  $(x_1, y_1)$ ,  $(x_2, y_2)$  da imagem corresponde aos 2 vértices do eixo maior de uma possível elipse.

Em seguida, calcula uma matriz com as distâncias entre esses pares de vértices (i.e. calcula o tamanho do eixo maior da elipse para cada situação).

**Etapa 02.03**

Reduz o espaço de buscas através da aplicação de algumas regras pré-definidas.

O processo de redução segue a seguinte ordem:

1. O **tamanho do eixo maior** deve estar dentro de uma faixa específica (i.e. entre os valores mínimos e máximos estabelecidos);
2. O **ângulo de rotação** deve estar dentro de uma faixa específica (i.e. entre os valores mínimos e máximos estabelecidos);
3. É feita uma **seleção aleatória** dos elementos da matriz de distância (apenas se a opção **randomização** estiver ativa).

**Etapa 02.04** †

Realiza a segmentação da imagem através do método da *transformada de Hough*.

**Etapa 03.01**

Exibe o histograma das pontuações atingidas por cada possível superelipse, juntamente com o limiar escolhido.

---

\*Vide detalhes da operação no tópico 3.3.2.

†Vide detalhes da operação no tópico 5.6.

**Etapa 03.02**

Traça a curva das superelipses encontradas.

**Etapas 03.03 e 03.04**

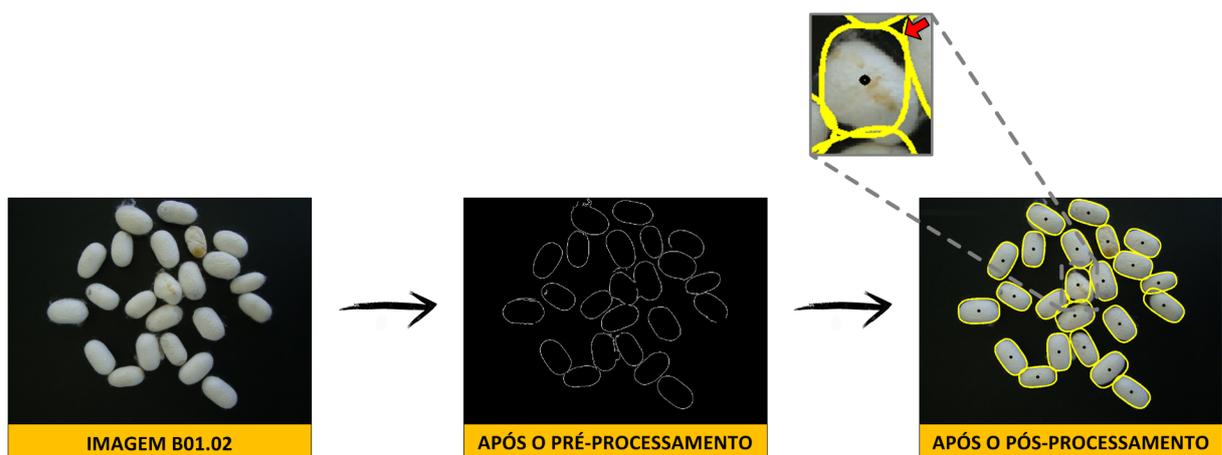
Essas etapas são idênticas às etapas homônimas descritas para o caso do algoritmo de segmentação pelo método do *threshold*.

**5.6.3 Resultados & Discussão**

Apesar de não estar explícito no fluxograma da Figura 51, após a execução de cada etapa é exibida uma imagem de saída, o que permite acompanhar a evolução do processo de segmentação.

Um exemplo dessa evolução pode ser visto na Figura 52, que compara a imagem original com a mesma imagem após o pré-processamento e após o pós-processamento:

Figura 52 – Exemplo da evolução da imagem ao longo do processo de segmentação - Método da Transformada de Hough

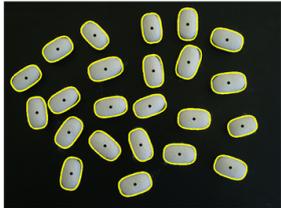
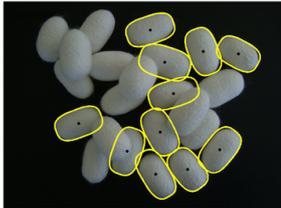
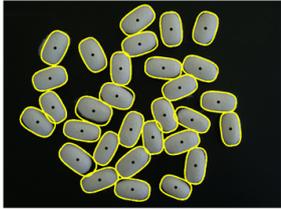
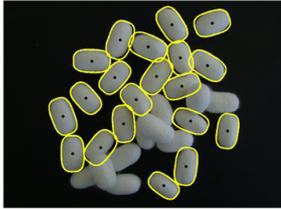
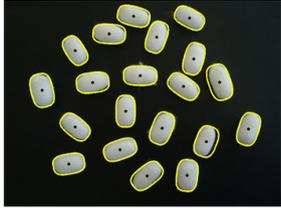
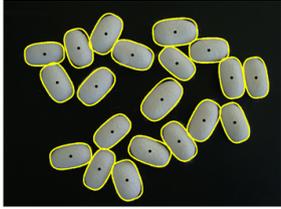
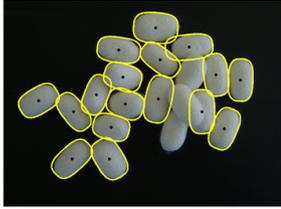


Fonte: Autoria própria

O resultado final do algoritmo é exposto nas Figuras 53 e 54.

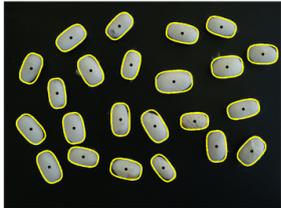
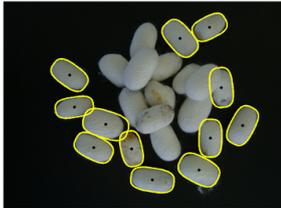
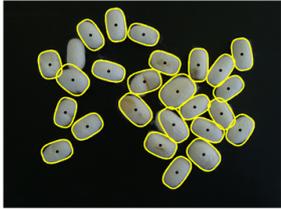
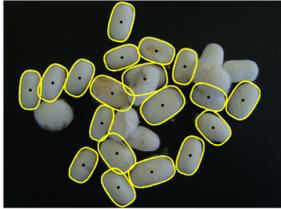
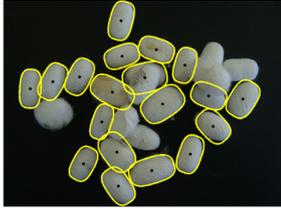
Como as figuras deixam claro, os resultados se referem à aplicação do algoritmo à todas as imagens de teste (conjunto de casulos bons e defeituosos, para todos os 3 cenários definidos).

Figura 53 – Resultado da segmentação do conjunto de casulos bons - Método da Transformada de Hough

CASULOS BONS			
	CENÁRIO 01	CENÁRIO 02	CENÁRIO 03
SUBCONJUNTO 01 (23 CASULOS)	 <p>IMAGEM B01.01</p>	 <p>IMAGEM B01.02</p>	 <p>IMAGEM B01.03</p>
SUBCONJUNTO 02 (30 CASULOS)	 <p>IMAGEM B02.01</p>	 <p>IMAGEM B02.02</p>	 <p>IMAGEM B02.03</p>
SUBCONJUNTO 03 (20 CASULOS)	 <p>IMAGEM B03.01</p>	 <p>IMAGEM B03.02</p>	 <p>IMAGEM B03.03</p>
SUBCONJUNTO 04 (25 CASULOS)	 <p>IMAGEM B04.01</p>	 <p>IMAGEM B04.02</p>	 <p>IMAGEM B04.03</p>

Fonte: Autoria própria

Figura 54 – Resultado da segmentação do conjunto de casulos defeituosos - Método da Transformada de Hough

CASULOS DEFEITUOSOS			
	CENÁRIO 01	CENÁRIO 02	CENÁRIO 03
SUBCONJUNTO 01 (23 CASULOS)	 <p>IMAGEM D01.01</p>	 <p>IMAGEM D01.02</p>	 <p>IMAGEM D01.03</p>
SUBCONJUNTO 02 (26 CASULOS)	 <p>IMAGEM D02.01</p>	 <p>IMAGEM D02.02</p>	 <p>IMAGEM D02.03</p>
SUBCONJUNTO 03 (31 CASULOS)	 <p>IMAGEM D03.01</p>	 <p>IMAGEM D03.02</p>	 <p>IMAGEM D03.03</p>
SUBCONJUNTO 04 (27 CASULOS)	 <p>IMAGEM D04.01</p>	 <p>IMAGEM D04.02</p>	 <p>IMAGEM D04.03</p>

Fonte: Autoria própria

Os resultados encontrados evidenciam que o algoritmo desenvolvido está sujeito à basicamente 4 tipos de problemas:

### **Falsos Negativos (vide Figura 54)**

Uma análise visual das imagens ao longo da evolução do algoritmo revela que a ocorrência de falsos negativos está associada a 3 fatores:

1. Falha do algoritmo de detecção de bordas;
2. Escolha inadequada do limiar de similaridade entre duas superelipses encontradas, utilizado na Etapa 02.04.
3. Escolha inadequada do limiar de pontuação, utilizado na Etapa 03.01;

Na 1ª situação, o algoritmo de segmentação não é capaz de encontrar uma superelipse pelo simples fato de que suas bordas não foram corretamente detectadas. Esse problema é bem nítido quando os casulos estão sobrepostos, diminuindo o contraste entre a borda e o plano de fundo.

O problema pode ser solucionado através da utilização de um algoritmo de detecção de bordas mais sofisticado, que utilize, por exemplo, um limiar adaptativo (vide Tópico 4.3.2).

Na 2ª situação o problema ocorre quando o limiar de similaridade é menor que o ideal. Nesse caso, o algoritmo considera erroneamente que duas superelipses distintas são na verdade o mesmo objeto.

Para corrigir esse problema, propõe-se estudar outras soluções para a avaliação do grau de similaridade entre as superelipses encontradas.

A 3ª situação, por sua vez, deriva da escolha de um limiar de pontuação muito elevado. Como essa escolha foi feita de maneira empírica para cada imagem, é interessante desenvolver um método sistemático para a determinação do limiar ótimo.

### **Falsos Positivos**

Mesmo que os resultados apresentados nas Figuras 53 e 54 não explicitem esse tipo de problema, uma configuração incorreta dos parâmetros do algoritmo pode provocar a detecção de falsos positivos (i.e. casulos detectados em locais onde os mesmos não existem).

As causas deste problema e, conseqüentemente, as soluções, estão ligada aos mesmos fatores que geram os falsos negativos.

**Forma Geométrica dos Casulos: Ideal x Real (vide Figura 54)**

A principal premissa para realizar uma segmentação através do método da transformada de Hough é a de que o objeto a ser detectado possui uma forma geométrica bem definida.

Entretanto, ao lidar com objetos que não são artificiais, sempre haverá uma variação entre a forma ideal e a forma real, o que pode trazer problemas a depender da aplicação.

No caso dos casulos, uma maneira de mitigar esse problema seria incluir a variação do parâmetro  $n$  no algoritmo, fazendo com que a curva da superelipse tivesse mais possibilidades de se ajustar ao objeto a ser detectado.

**Erro nos Parâmetros da Superelipse (vide Figura 54)**

Em alguns casos, apesar da superelipse ser detectada, a estimação de seus parâmetros é feita de maneira incorreta.

Isso normalmente está associado a 2 fatores:

1. Uma etapa de detecção de bordas imperfeita;
2. Falha na etapa de segmentação, que “confunde” os pontos de borda de dois casulos diferentes.

A solução para a 1ª situação é idêntica à solução para o problema dos falsos negativos.

Para a 2ª situação, uma maneira de melhorar os resultados seria diminuir o nível de randomização utilizado na Etapa 02.03 do algoritmo.

A Tabela 5 traz uma compilação dos resultados encontrados para todas as imagens analisadas (conforme convenção de nomenclatura definida nas Figuras 38 e 39):

Tabela 5 – Resumo dos resultados da segmentação - Método da *Transformada de Hough*<sup>2</sup>

SEGMENTAÇÃO PELO MÉTODO DA <i>TRANSFORMADA DE HOUGH</i>				
#	IMAGEM	TAXA DE ACERTOS		TEMPO DE EXECUÇÃO
		ABSOLUTA	RELATIVA	
01	B01.01	23/23	100,00 %	20,77 s
02	B01.02	23/23	100,00 %	52,55 s
03	B01.03	11/23	47,83 %	52,25 s
04	B02.01	30/30	100,00 %	29,73 s
05	B02.02	30/30	100,00 %	68,90 s
06	B02.03	20/30	66,67 %	66,54 s
07	B03.01	20/20	100,00 %	18,91 s
08	B03.02	20/20	100,00 %	45,64 s
09	B03.03	17/20	85,00 %	52,37 s
10	B04.01	25/25	100,00 %	23,57 s
11	B04.02	25/25	100,00 %	55,58 s
12	B04.03	18/25	72,00 %	58,05 s
13	D01.01	23/23	100,00 %	21,12 s
14	D01.02	22/23	95,65 %	46,14 s
15	D01.03	12/23	52,17 %	50,43 s
16	D02.01	26/26	100,00 %	24,87 s
17	D02.02	25/26	96,15 %	44,65 s
18	D02.03	19/26	73,08 %	64,66 s
19	D03.01	30/31	96,77 %	25,83 s
20	D03.02	29/31	93,55 %	54,33 s
21	D03.03	14/31	45,16 %	55,77 s
22	D04.01	27/27	100,00 %	23,95 s
23	D04.02	26/27	96,30 %	54,58 s
24	D04.03	22/27	81,48 %	86,72 s
MÉDIA POR CENÁRIO				
CENÁRIO 01		204/205	99,51 %	23,59 s
CENÁRIO 02		200/205	97,56 %	52,80 s
CENÁRIO 03		133/205	64,88 %	60,85 s

Fonte: Autoria própria

<sup>2</sup>Para garantir um tempo de execução aceitável, todos os resultados expostos nesse trabalho foram obtidos através de simulações realizadas com o parâmetro **randomização** ativo.

Devido à característica de aleatoriedade dessa etapa do algoritmo, existe uma pequena variação relativa à reprodutibilidade dos resultados encontrados.

Como pode-se observar, para cada uma das imagens foram analisados dois parâmetros de desempenho:

- A taxa de acertos;
- O custo computacional (i.e. tempo de execução).

Ao analisar os resultados por cenário, fica evidente que o algoritmo funciona extremamente bem quando os casulos estão distantes uns dos outros, atingindo uma taxa de acertos próxima à 100 %.

De maneira semelhante, os resultados obtidos para o caso em que os casulos se tocam também são excelentes, visto que a taxa de acertos foi acima de 97 %.

Finalmente, para o cenário em que os casulos estão sobrepostos, a taxa de acertos atingida foi acima da média, girando em torno de 65 %.

Em ambos os casos, é importante destacar que ainda existe espaço para otimização do algoritmo: uma melhoria nos resultados pode ser alcançada através do estudo e implementação das sugestões expostas para resolução dos problemas encontrados.

Em relação ao tempo de execução do algoritmo, os valores encontrados são relativamente altos, da ordem de algumas dezenas de segundos. Deste modo, para uma aplicação em tempo real, como por exemplo uma inspeção feita na linha de produção, seria necessário reduzir esse tempo (mas esse não foi o foco do trabalho desenvolvido).

Uma análise da influência de cada etapa do algoritmo no tempo total de execução revela que cerca de 90 % desse tempo é gasto na etapa de processamento. Sendo assim, a otimização desse parâmetro passa, necessariamente, por uma melhoria das etapas relacionadas à transformada de Hough.

## 5.7 Comparação dos Resultados

A Tabela 6 faz uma comparação dos resultados encontrados, por algoritmo de segmentação, para cada um dos 3 cenários utilizados (conforme convenção de nomenclatura definida na Figura 37):

Tabela 6 – Resumo dos resultados da segmentação - Comparação entre os métodos utilizados

COMPARAÇÃO ENTRE OS MÉTODOS DE SEGMENTAÇÃO			
MÉTODO DE SEGMENTAÇÃO	TAXA DE ACERTOS		TEMPO DE EXECUÇÃO
	ABSOLUTA	RELATIVA	
CENÁRIO 01			
THRESHOLD	202/205	98,54 %	6,83 s
WATERSHED	202/205	98,54 %	3,99 s
TRANSFORMADA DE HOUGH	204/205	99,51 %	23,59 s
CENÁRIO 02			
THRESHOLD	61/205	29,76 %	3,45 s
WATERSHED	177/205	86,34 %	4,13 s
TRANSFORMADA DE HOUGH	200/205	97,56 %	52,80 s
CENÁRIO 03			
THRESHOLD	31/205	15,12 %	2,55 s
WATERSHED	88/205	42,93 %	3,81 s
TRANSFORMADA DE HOUGH	133/205	64,88 %	60,85 s

Fonte: Autoria própria

Analisando a taxa de acertos para o caso em que os casulos não se tocam (i.e. Cenário 01), pode-se concluir que o desempenho dos 3 algoritmos é bastante satisfatório.

Os resultados do Cenário 02 deixam claro que o algoritmo de segmentação por *threshold*, da maneira como foi implementado, não é capaz de isolar os casulos que se tocam. Esse problema pode ser contornado com a utilização de um dos outros dois métodos implementados.

Finalmente, no caso específico da ocorrência de oclusão parcial dos objetos a serem detectados (i.e. Cenário 03), apenas o método de segmentação pela transformada de Hough é capaz de fornecer uma saída correta.

Para todos os cenários analisados, o algoritmo de segmentação pelo método da transformada de Hough foi o que alcançou a maior taxa de acertos, seguido pelo método do *watershed* e, por último, pelo método do *threshold*.

Em relação ao tempo de execução, essa ordem se inverte, sendo o método da transformada de Hough, de longe, o mais custoso do ponto de vista computacional.

## 6 Considerações finais

Este capítulo apresenta um apanhado geral do trabalho desenvolvido, visando expor os resultados obtidos, as contribuições realizadas e as principais propostas de trabalhos posteriores.

### 6.1 Conclusões

Dado o exposto, pode-se considerar que os objetivos do trabalho foram atingidos, visto que foi possível:

- Criar um banco de imagens dos casulos do bicho-da-seda;
- Analisar, de maneira sistemática, as características do problema;
- Comparar os diferentes métodos de segmentação de imagens (com base em suas principais características);
- Identificar os métodos de segmentação de imagens mais adequados à resolução do problema;
- Implementar os algoritmos identificados e comparar seus respectivos desempenhos.

O teste das hipóteses definidas no Capítulo 1 revelou que:

#### **Hipótese 01 → Validada**

Os resultados obtidos demonstram que, sob condições específicas de trabalho, *todos os 3 métodos clássicos de segmentação de imagem implementados podem ser utilizados na detecção de casulos do bicho-da-seda.*

#### **Hipótese 02 → Parcialmente Validada**

Para avaliar se o desempenho dos métodos implementados é suficiente ou não para utilização em uma aplicação industrial, *é preciso analisar junto ao cliente todas as possíveis soluções.*

Por exemplo, caso a inspeção possa ser feita *offline*, as restrições em relação ao tempo de execução do algoritmo serão menos rigorosas (o que flexibiliza as decisões de projeto).

Os resultados encontrados demonstram que qualquer um dos algoritmos pode ser utilizado na situação definida no Cenário 01.

Já no Cenário 02, o método do *threshold* não é capaz de isolar os casulos que se tocam, tornando-se inadequado à tarefa.

Finalmente, para o Cenário 03, apenas o método da transformada de Hough é capaz de detectar os casulos que estão parcialmente sobrepostos.

Para todos os cenários analisados, o algoritmo de segmentação pelo método da transformada de Hough foi o que alcançou a maior taxa de acertos, seguido pelo método do *watershed* e, por último, pelo método do *threshold*.

Em relação ao tempo de execução, essa ordem se inverte, sendo o método da transformada de Hough, de longe, o mais custoso do ponto de vista computacional.

## 6.2 Contribuições

Além das contribuições já elencadas (Seção 6.1), acredita-se que talvez a maior contribuição do trabalho tenha sido a identificação da possibilidade de modelar um casulo do bicho-da-seda através de uma superelipse e, em seguida, a validação desse modelo.

## 6.3 Atividades Futuras de Pesquisa

Em continuidade a este trabalho, sugere-se as seguintes ações futuras:

### **Sistema de Detecção, Classificação e Retirada dos Casulos**

- Projeto do algoritmo de classificação dos casulos (em execução);
- Especificação do sistema de aquisição de imagens;
- Especificação do *hardware* que possuirá o algoritmo de visão computacional embarcado;
- Projeto do sistema de condicionamento;
- Projeto do sistema de retirada dos casulos defeituosos (i.e. atuador);
- Integração de todo o sistema.

### **Algoritmos**

- Utilizar uma linguagem mais eficiente e mais adequada à ser embarcada (e.g. C, C++);
- Implementar técnicas de paralelismo (Paralelismo Algorítmico x Paralelismo de Dados);
- Estudar a arquitetura de processador mais adequada ao problema (incluindo a possibilidade da utilização de processadores gráficos);
- Implementar as sugestões de melhoria dos algoritmos de segmentação já sugeridas no Capítulo 5.

## Modelagem

- Utilizar alguma técnica de ajuste de curvas para determinar o valor do parâmetro  $n$  que mais se adequa aos casulos.

Além disso, de modo semelhante ao que foi feito para o caso da superelipse (Seção 5.6), [Gielis \(2003\)](#) demonstra que, partindo da fórmula de um círculo, pode-se chegar à uma equação relativamente simples e capaz de descrever uma miríade de formas geométricas presentes na natureza, conhecida como **Superfórmula**.

Sendo assim, outra alternativa interessante de trabalho futuro seria explorar a utilização da técnica da transformada de Hough aliada à modelagem por meio de superfórmulas. Unir a robustez da segmentação pela transformada de Hough à simplicidade e versatilidade da modelagem por superfórmulas pode se revelar como um grande avanço no campo da segmentação de imagens.

## Referências

- BARBER, E. *Prehistoric Textiles: The Development of Cloth in the Neolithic and Bronze Ages with Special Reference to the Aegean*. Princeton University Press, 1992. ISBN 978-0-691-00224-8. Disponível em: <<http://tinyurl.com/barber1992>>. Citado na página 24.
- BOADO, C. *Importance of Silk*. 2012. Disponível em: <<http://tinyurl.com/ArtSilk2012>>. Citado na página 25.
- BRATAC. *Vídeo Institucional*. 2002. Disponível em: <<http://www.youtube.com/watch?v=1qm10b2ahFw>>. Citado nas páginas 16 e 33.
- Centro Ciência Viva de Bragança. *Ciclo de vida do bicho-da-seda*. 2014. Disponível em: <<http://www.braganca.cienciaviva.pt/seda/>>. Citado na página 30.
- COOK, J. D. *Three algorithms for converting color to grayscale*. 1999. Disponível em: <<http://www.johndcook.com/>>. Citado na página 41.
- DAVIES, E. R. *Computer and Machine Vision: Theory, Algorithms, Practicalities*. 4th. ed. [S.l.]: Academic Press, 2012. ISBN 978-0-123-86908-1. Citado nas páginas 37, 42, 48, 52, 53, 54, 55, 62, 66, 67, 68, 71, 72, 73 e 74.
- DEMANT, C.; STREICHER-ABEL, B.; GARNICA, C. *Industrial Image Processing: Visual quality control in manufacturing*. 2nd. ed. [S.l.]: Springer, 2013. ISBN 978-3-642-33904-2. Citado nas páginas 19, 21 e 50.
- Encyclopedia Britannica. *Sericulture*. 2014. Disponível em: <<http://global.britannica.com/EBchecked/topic/535632/sericulture>>. Citado na página 29.
- Encyclopedia Britannica. *Silk*. 2014. Disponível em: <<http://global.britannica.com/EBchecked/topic/544449/silk>>. Citado na página 24.
- Excellentware Brasil. Relatório de Prospecção Tecnológica, *Machine Vision: Inspeção Automatizada na Indústria*. 2004. Citado nas páginas 50 e 53.
- FAO. *Food and Agriculture Organization Corporate Statistical Database: Faostat domains: Livestock primary and livestock processed*. 2012. Disponível em: <<http://faostat3.fao.org/faostat-gateway/go/to/home/E>>. Citado nas páginas 26, 27, 129 e 130.
- FIEP, A. *Embrapii aprova projeto de indústria paranaense*. 2013. Disponível em: <<http://global.britannica.com/EBchecked/topic/544449/silk>>. Citado na página 16.
- FILHO, O. M.; NETO, U. V. *Processamento Digital de Imagens*. 1a. ed. [S.l.]: Brasport, 1999. ISBN 978-8-574-52009-4. Citado nas páginas 40, 42, 44, 46, 48 e 49.
- FISHER, R. et al. *Dictionary of Computer Vision and Image Processing*. 1st. ed. [S.l.]: John Wiley & Sons, 2005. ISBN 978-0-470-01526-1. Citado nas páginas 38, 42, 43, 124 e 127.
- GIELIS, J. A generic geometric transformation that unifies a wide range of natural and abstract shapes. *American journal of botany*, v. 90, n. 3, p. 333–338, 2003. Citado na página 119.

- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 2nd. ed. [S.l.]: Prentice hall, 2002. ISBN 978-0-201-18075-6. Citado nas páginas 35, 37, 38, 43, 44, 45, 56, 57, 58, 60, 61, 63 e 65.
- HO, P.-G. P. *Image Segmentation*. 1st. ed. [S.l.]: InTech, 2011. ISBN 978-9-533-07228-9. Citado na página 56.
- HORNBERG, A. *Handbook of Machine Vision*. 1st. ed. [S.l.]: Wiley-VCH, 2006. ISBN 978-3-527-40584-8. Citado nas páginas 12, 39 e 53.
- INDIA, C. S. B. M. of Textiles Government of. *Silk*. 2013. Disponível em: <<http://www.csb.gov.in/silk-sericulture/silk/>>. Citado na página 26.
- International Sericultural Commission. *Evolution of Silk*. 2013. Disponível em: <[http://inserco.org/en/evolution\\_of\\_silk](http://inserco.org/en/evolution_of_silk)>. Citado na página 24.
- Jornal de Londrina. *Paraná faz o melhor fio de seda do mundo*. 2013. Disponível em: <<http://tinyurl.com/JornalLondrina>>. Citado nas páginas 27 e 29.
- JÄHNE, B.; HAUBECKER, H. *Computer Vision and Applications: A Guide for Students and Practitioners*. 1st. ed. [S.l.]: Academic Press, 2000. ISBN 978-0-123-79777-3. Citado na página 50.
- KIRYATI, N.; ELDAR, Y.; BRUCKSTEIN, A. M. A probabilistic hough transform. *Pattern recognition*, Elsevier, v. 24, n. 4, p. 303–316, 1991. Citado na página 74.
- KLETTE, R. *Concise Computer Visino*. 1st. ed. [S.l.]: Springler, 2014. ISBN 978-1-4471-6319-0. Citado na página 57.
- KUSNAMAN, D. *Development Planning and Project Cycle Analysis for Sericulture in Central Java*. Tese de Doutorado, 2004. Citado na página 31.
- LEE, Y. W. *Silk Reeling and Testing Manual*. Food and Agriculture Organization of the United Nations, 1999. (FAO agricultural services bulletin, nº 136). ISBN 978-9-251-04293-9. Disponível em: <<http://www.fao.org/docrep/x2099e/x2099e00.htm>>. Citado nas páginas 31 e 33.
- LONG, D. *ellipse.m*. 1998. Disponível em: <<http://www.mathworks.com/matlabcentral/fileexchange/289-ellipse-m>>. Citado na página 106.
- MAINTZ, T. Digital and medical image processing. *Lecture Notes, Utrecht University*, 2005. Citado nas páginas 35, 36, 41, 58, 62, 70 e 78.
- MATHWORKS. *Correcting Nonuniform Illumination*. 2014. Disponível em: <<http://www.mathworks.com/help/images/examples/correcting-nonuniform-illumination.html>>. Citado na página 44.
- MATHWORKS. *Marker-Controlled Watershed Segmentation*. 2014. Disponível em: <<http://www.mathworks.com/help/images/examples/marker-controlled-watershed-segmentation.html>>. Citado na página 93.
- SHAPIRO, L. G.; STOCKMAN, G. C. *Computer Vision*. 1st. ed. [S.l.]: Prentice Hall, 2001. ISBN 978-0-130-30796-5. Citado nas páginas 48 e 50.

SHIH, F. Y. *Image Processing and Pattern Recognition: Fundamentals and techniques*. 1st. ed. [S.l.]: John Wiley & Sons, 2010. ISBN 978-0-470-40461-4. Citado na página 38.

SILVEIRA, C. B. *Sete benefícios conquistados através da Automação Industrial*. 2014. Disponível em: <<http://www.citisystems.com.br/sete-beneficios-automacao-industrial/>>. Citado na página 19.

SIMONOVSKY, M. *Ellipse Detection Using 1D Hough Transform*. 2013. Disponível em: <<http://www.mathworks.com/matlabcentral/fileexchange/33970-ellipse-detection-using-1d-hough-transform>>. Citado na página 106.

SRIDHAR, N. *Cycle of Silkworm*. 2010. Disponível em: <<http://tinyurl.com/SilkwormCycle>>. Citado na página 29.

TESINOVA, P. *Advances in Composite Materials - Analysis of Natural and Man-Made Materials*: Chapter 6 - composite materials from natural resources: Recent trends and future potentials. 1st. ed. [S.l.]: InTech, 2011. ISBN 978-9-533-07449-8. Citado na página 28.

Texere Silk. *Benefits of Silk*. 2012. Disponível em: <[http://texeresilk.com/article/benefits\\_of\\_silk](http://texeresilk.com/article/benefits_of_silk)>. Citado na página 24.

The Textile Museum. *An Introduction to Textile Terms*. 1997. Citado na página 24.

TIMES, T. N. Y. *The Reinvention of Silk*. 2011. Disponível em: <<http://tinyurl.com/TheNYTimes2011>>. Citado na página 25.

TKALCIC, M.; TASIC, J. Colour spaces: Perceptual, historical and applicational background. *EUROCON 2003. Computer as a Tool. The IEEE Region 8*, 2003. Citado nas páginas 39 e 40.

WEISSTEIN, E. W. *Superellipse*. 2014. Disponível em: <<http://mathworld.wolfram.com/Superellipse.html>>. Citado na página 102.

XIE, Y.; JI, Q. A new efficient ellipse detection method. In: IEEE. *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. [S.l.], 2002. v. 2, p. 957–960. Citado nas páginas 72, 75, 104, 105 e 108.

ZANETTI, R. Notas de aula de ent 110 - sericicultura. 2013. Citado na página 34.

# Apêndices

## APÊNDICE A – Dicionários

Durante a etapa de revisão bibliográfica, feita em grande parte em língua inglesa, houve dificuldade na compreensão de alguns termos específicos relacionados ao tema do trabalho.

Ciente desta dificuldade e visando facilitar os trabalhos posteriores, elaborou-se uma lista não-exaustiva desses termos. A lista foi dividida em duas partes, de acordo com o tema:

- Produção da Seda;
- Processamento de Imagens & Visão Computacional<sup>1</sup>.

É importante destacar que:

- Nem todas as palavras da lista foram utilizadas no texto da monografia (mas todas são comumente encontradas na bibliografia disponível);
- O critério de seleção das palavras foi subjetivo e arbitrário (cabendo ao autor avaliar os graus de dificuldade e especificidade de cada termo).

---

<sup>1</sup>Uma lista mais refinada, elaborada por especialistas da área e contendo mais de 2 500 termos, pode ser consultada em (FISHER et al., 2005).

## A.1 Produção da Seda

<i>Animal Husbandry</i> .....	Pecuária
<i>Automatic Reeling Machine (ARM)</i> .....	Máquina Automática de Enrolamento
<i>Bale</i> .....	Fardo
<i>Book</i> .....	Conjunto de objetos (neste caso novelos de seda) embalados e protegidos por uma capa ou cobertura
<i>Caterpillar</i> .....	Lagarta
<i>Cocoon</i> .....	Casulo
<i>Cottage Industry</i> .....	Indústria pequena e normalmente informalmente organizada
<i>Degum, to</i> .....	<i>Tirar a goma</i> ; Descolar <sup>2</sup>
<i>Denier</i> .....	Unidade de medida para densidade linear de fibras (g/9000.m)
<i>Drape</i> .....	Panejamento
<i>Dye, to</i> .....	Tingir
<i>Fabric</i> .....	Tecido
<i>Loom</i> .....	Máquina de Tear
<i>Moth</i> .....	Mariposa
<i>Mould</i> .....	Mofo
<i>Moult, to</i> .....	Trocar de pele
<i>Mulberry (or Morus)</i> .....	Amora
<i>Peel, to</i> .....	Descascar; Puxar a ponta de algo para desenrolar
<i>Rear, to</i> .....	Criar
<i>Reel, to</i> .....	Enrolar; Bobinar
<i>Ripe</i> .....	Maduro
<i>Satin</i> .....	Cetim
<i>Sheen</i> .....	Brilho
<i>Silk</i> .....	Seda
<i>Silkworm</i> .....	Bicho-da-Seda
<i>Skein (or Hank)</i> .....	Novelo; Meada
<i>Thread</i> .....	Linha; Fio
<i>Throw, to</i> .....	Trançar o fio de seda bobinado para produzir a fibra têxtil
<i>Warp</i> .....	Fios que formam um tecido no sentido de seu comprimento
<i>Weave, to</i> .....	Tecer
<i>Weft</i> .....	Fios que formam um tecido no sentido perpendicular ao seu comprimento
<i>Wool</i> .....	Lã
<i>Yarn</i> .....	Fibra Têxtil

<sup>2</sup>Neste contexto, se refere ao processo de amolecer e/ou remover parte da *sericina* do casulo do bicho-da-seda.

## A.2 Processamento de Imagens & Visão Computacional

<i>Active Contours/Snakes</i> .....	Contornos Ativos/Cobras; Nome de uma técnica de segmentação de imagens
<i>Affine Projection</i> .....	Projeção Afim
<i>Artifact</i> .....	Artefato <sup>3</sup>
<i>Background/Foreground</i> .....	Plano de Fundo/Plano Frontal
<i>Binary Image</i> .....	Imagem Binária (i.e. composta apenas por 2 cores: Branco e Preto)
<i>Blob</i> .....	Gota
<i>Catchment/Drainage Basin</i> .....	Bacia de Captação/Drenagem
<i>Clustering</i> .....	Clusterização; Agrupamento
<i>(Closing/Opening)-by-Reconstruction</i> .....	(Fechamento/Abertura)-por-Reconstrução
<i>Commercial Off-the-Shelf (COTS)</i> .....	Componentes Comerciais “de Prateleira”
<i>Connected Components</i> .....	Componentes Conectados
<i>Convex Hull</i> .....	Envoltória Convexa; Envelope Convexo
<i>Dam</i> .....	Barragem; Represa
<i>Dilation</i> .....	Dilatação
<i>Erosion</i> .....	Erosão
<i>Feature</i> .....	Característica; Atributo <sup>4</sup>
<i>Flood, to</i> .....	Inundar; Alagar
<i>Frame</i> .....	Quadro
<i>Hue</i> .....	Matiz; Tonalidade
<i>Landscape</i> .....	Paisagem
<i>Likelihood</i> .....	Probabilidade; Verossimilhança
<i>Look-Up Table (LUT)</i> .....	Tabela de Referência
<i>Machine Vision</i> .....	Visão de Máquina; Visão Artificial; Sistema Automatizado de Inspeção Visual
<i>Match, to</i> .....	Comparar; Confrontar
<i>Parameter Space</i> .....	Espaço de Parâmetros
<i>Pruning</i> .....	Poda
<i>Ridge</i> .....	Tergo; Cumeada; Cumeeira; Formação Geográfica Oposta ao Vale
<i>Rubber Band</i> .....	Elástico; Tira de Borracha
<i>Rugged</i> .....	Robusto
<i>Scrutinize, to</i> .....	Examinar minuciosamente
<i>Skeletonization</i> .....	“Esqueletização”
<i>Skeleton by Influence Zones (SKIZ)</i> .....	Esqueleto por Zonas de Influência
<i>Smooth, to</i> .....	Suavizar

---

<i>Thickening</i> .....	Espessamento
<i>Thinning</i> .....	Afinamento; Desbaste
<i>Threshold</i> .....	Limiar
<i>Top Hat</i> .....	Cartola
<i>Watershed</i> .....	Bacia Hidrográfica

---

<sup>3</sup>Neste contexto, o **artefato** é um erro na percepção ou representação de alguma informação visual ou sonora, introduzido pelos equipamentos ou técnicas utilizadas.

<sup>4</sup>É difícil traduzir a palavra *feature* de maneira precisa. De acordo com (FISHER et al., 2005), *feature* pode ser:

- Uma característica notável de algo (e.g. nariz e olhos de uma face; quinas e bordas de uma mesa);
- Um atributo derivado de um objeto/forma (e.g. brilho, textura, circularidade).

# Anexos

## ANEXO A – Informações de Produção

Para melhorar a visualização das informações, parte dos dados utilizados na construção das Figuras 3 e 4 foram omitidos.

As tabelas abaixo (Tabelas 9 e 10) trazem os dados completos, tal qual obtidos no site da FAO.

Tabela 9 – *Ranking* mundial de produção de casulos do bicho-da-seda

CASULOS DO BICHO-DA-SEDA (2012)			
POSIÇÃO	PAÍS	PRODUÇÃO (TON.)	PRODUÇÃO (%)
1º	China	370 002	65,66%
2º	Índia	151 000	26,80%
3º	Uzbequistão	25 500	4,53%
4º	Tailândia	4 600	0,82%
5º	Irã	3 300	0,59%
6º	Brasil	3 219	0,57%
7º	Vietnã	2 500	0,44%
8º	Coreia do Norte	900	0,16%
9º	Romênia	790	0,14%
10º	Afeganistão	700	0,12%
11º	Japão	202	0,04%
12º	Camboja	160	0,03%
13º	Turquia	151	0,03%
14º	Egito	110	0,02%
15º	Quirguistão	100	0,02%
16º	Espanha	60	0,01%
17º	Itália	50	0,01%
18º	Madagascar	50	0,01%
19º	Líbano	40	0,01%
20º	Nepal	31	0,01%
21º	Bulgária	20	<0,01%
22º	Grécia	10	<0,01%
23º	Azerbaijão	3	<0,01%
24º	Síria	1	<0,01%
25º	França	<1	<0,01%
26º	Myanmar	<1	<0,01%
27º	Polônia	<1	<0,01%
28º	Coreia do Sul	<1	<0,01%

Fonte: Adaptado de (FAO, 2012)

Tabela 10 – Ranking mundial de produção de seda crua

SEDA CRUA (2012)			
POSIÇÃO	PAÍS	PRODUÇÃO (TON.)	PRODUÇÃO (%)
1º	China	126 001	77,94%
2º	Índia	20 410	12,63%
3º	Vietnã	7 517	4,65%
4º	Romênia	2 100	1,30%
5º	Tailândia	1 600	0,99%
6º	Uzbekistão	1 200	0,74%
7º	Irã	900	0,56%
8º	Brasil	500	0,31%
9º	Coreia do Norte	400	0,25%
10º	Japão	280	0,17%
11º	Tajiquistão	200	0,12%
12º	Turcomenistão	200	0,12%
13º	Indonésia	120	0,07%
14º	Afeganistão	50	0,03%
15º	Quirguistão	50	0,03%
16º	Turquia	50	0,03%
17º	Camboja	30	0,02%
18º	Espanha	15	0,01%
19º	Itália	12	0,01%
20º	Líbano	10	0,01%
21º	Bulgária	5	<0,01%
22º	Grécia	5	<0,01%
23º	Egito	3	<0,01%
24º	Coreia do Sul	3	<0,01%
25º	França	<1	<0,01%

Fonte: Adaptado de (FAO, 2012)