



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

Victor Ariel Leal Sobral

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

Salvador
Dezembro, 2015

Victor Ariel Leal Sobral

**Visão Computacional aplicada ao Controle de Qualidade de
Casulos do Bicho-da-Seda: *Subsistema de Classificação 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
Dezembro, 2015

Ficha catalográfica elaborada pela Biblioteca da Faculdade de Tecnologia SENAI CIMATEC

S677v Sobral, Victor Ariel Leal

Visão computacional aplicada ao controle de qualidade de casulos do bicho-da-seda: subsistema de segmentação de imagens / Victor Ariel Leal Sobral – Salvador, 2015.

67 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. Descritores de imagens. 3. Classificação de imagens. 4. Casulo do bicho-da-seda. I. Faculdade de Tecnologia SENAI CIMATEC. II. Alves, Bruno Guimarães Costa. III. Título.

CDD 629.8

Agradecimentos

Agradeço ao SENAI CIMATEC pela oportunidade de realizar este gratificante trabalho, com inúmeras aplicações práticas na indústria. Agradeço à empresa BRATAC por ceder as amostras de casulos e as imagens de casulos da sua linha de produção. Agradeço à Bruno Guimarães Costa Alves, por toda a paciência e disponibilidade ao longo da execução deste trabalho. Agradeço a todos os professores e colegas da Especialização em Automação em Controle do SENAI CIMATEC pelo apoio e frutíferas discussões.

Resumo

O controle de qualidade na indústria da fiação de seda é tradicionalmente realizado de maneira manual. Com o objetivo de automatizar a etapa de seleção de casulos adequados para a produção de seda, propõe-se um sistema de visão computacional para substituir a inspeção visual realizada, por exemplo, na linha de produção da empresa BRATAC. Este trabalho tem como objetivo o desenvolvimento de um algoritmo piloto para o subsistema de classificação de imagens de casulos de bicho-da-seda. Para o desenvolvimento e avaliação deste algoritmo de seleção, três bancos de dados de imagens de casulos adequados e defeituosos foram formados, algoritmos de extração de descritores de forma, textura e SIFT foram desenvolvidos e a similaridade entre descritores foi avaliada em relação a sua capacidade de classificar casulos em um grupo adequado à produção de seda e um grupo de casulos defeituosos. Os algoritmos foram desenvolvidos na plataforma Matlab e mostraram resultados promissores para a classificação de casulos de bicho-da-seda.

Palavras-chaves: casulo do bicho-da-seda, produção de seda, classificação de imagens, visão computacional, descritores de imagens, avaliação de similaridade.

Abstract

The quality control in the silk fiber industry is traditionally done in manual fashion. With the objective to automate the visual inspection selection of adequate cocoons in the silk production, it is proposed a computer vision system to substitute visual inspection done, as example, in the BRATAC company production line. This work's objective is the development of a pilot algorithm for the silkworm cocoons image classification subsystem. To the development and evaluation of this selection algorithm, three image databases of adequate and defective cocoons were formed; texture, shape and SIFT descriptors extraction algorithms were developed and descriptor similarity was evaluated in its ability to classify cocoons in a group adequate to silk production and a defective group. The algorithms were developed in Matlab platform and promising results were shown to to the silkworm cocoon classification.

Key-words: silkworm cocoon, image classification, computer vision, image descriptors, similarity evaluation.

Lista de ilustrações

Figura 1 – Automatização da linha de produção de fios de seda	9
Figura 2 – Etapas da visão computacional aplicada à classificação de casulos	12
Figura 3 – Comparação de casulos quanto à forma	14
Figura 4 – Extração dos parâmetros para o descritor elíptico de Fourier	15
Figura 5 – Descritores de forma	17
Figura 6 – Casulos defeituosos com manchas	17
Figura 7 – Casulos defeituosos com manchas: Quadros RGB	18
Figura 8 – Espaço de escalas por Diferença de Gaussianas	21
Figura 9 – Descritor SIFT: histograma de gradientes	22
Figura 10 – Distância Euclidiana	23
Figura 11 – Distância de Mahalanobis	24
Figura 12 – Banco de dados 1: casulos adequados e defeituosos	25
Figura 13 – Banco de dados 2: casulos adequados e defeituosos	26
Figura 14 – Banco de dados 3: casulos adequados e defeituosos	26
Figura 15 – Cadeia de processamento de dados para classificação de casulos	27
Figura 16 – Banco de dados 1: segmentação dos casulos	28
Figura 17 – Banco de dados 2: segmentação dos casulos	28
Figura 18 – Banco de dados 3: segmentação dos casulos	28
Figura 19 – Média das distâncias mínimas para os descritores SIFT	30
Figura 20 – Distância de Mahalanobis para os descritores de Forma	30
Figura 21 – Distância de Mahalanobis para os descritores de Textura	31
Figura 22 – Médias das distâncias Euclidianas mínimas para os descritores SIFT	31
Figura 23 – Classificação de casulos considerados adequados	32
Figura 24 – Quadro de classificação de casulos considerados adequados	32
Figura 25 – Classificação de casulos considerados defeituosos	33
Figura 26 – Quadro de classificação de casulos considerados defeituosos	33
Figura 27 – Distância de Mahalanobis para os descritores de Forma - BD2	34
Figura 28 – Distância de Mahalanobis para os descritores de Textura - BD2	34
Figura 29 – Médias das distâncias Euclidianas mínimas para os descritores SIFT - BD2	35
Figura 30 – Classificação de casulos adequados (imagem 1) - BD2	35
Figura 31 – Classificação de casulos adequados (imagem 2) - BD2	36
Figura 32 – Classificação de casulos defeituosos - BD2	36
Figura 33 – Quadro da classificação dos casulos defeituosos- BD2	37
Figura 34 – Classificação automática dos casulos - BD2	37
Figura 35 – Quadro da classificação automática dos casulos - BD2	38
Figura 36 – Distância de Mahalanobis para os descritores de Forma - BD3	38

Figura 37 – Distância de Mahalanobis para os descritores de Textura - BD3	39
Figura 38 – Médias das distâncias Euclidianas mínimas para os descritores SIFT - BD3 .	39
Figura 39 – Classificação de casulos adequados (imagem 1) - BD3	40
Figura 40 – Classificação de casulos adequados (imagem 2) - BD3	40
Figura 41 – Classificação de casulos defeituosos - BD3	41
Figura 42 – Quadro da classificação dos casulos defeituosos- BD3	41
Figura 43 – Classificação automática dos casulos - BD3	42
Figura 44 – Quadro da classificação automática dos casulos - BD3	42

Lista de abreviaturas e siglas

BRATAC	<i>Brazil Takushoku Kumiai</i>
CIMATEC	Centro Integrado de Manufatura e Tecnologia
FIEB	Federação das Indústrias do Estado da Bahia
JP(E)G	<i>Joint Photographic Experts Group</i>
NIST	<i>National Institute of Standards and Technology</i>
RGB	<i>Red, Green and Blue</i>
SEMATECH	<i>Semiconductor Manufacturing Technology</i>
SENAI	Serviço Nacional de Aprendizagem Industrial
SIFT	<i>Scale Invariant Feature Transform</i>

1 Introdução

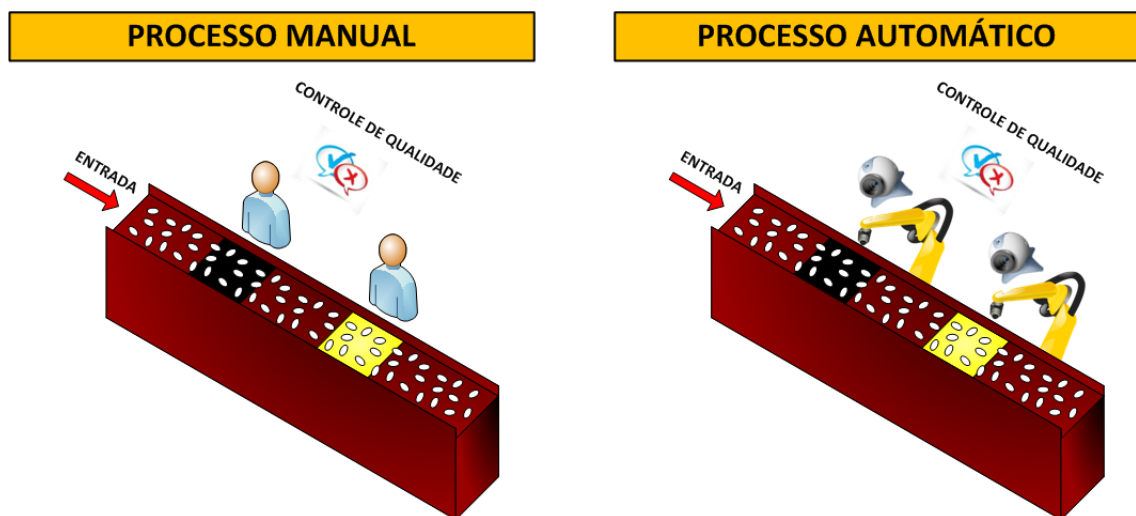
1.1 Contexto do trabalho

A empresa Fiação de Seda BRATAC (situada em Londrina/PR) atualmente investe em projetos de inovação em parceria com o SENAI CIMATEC na Bahia (FIEP, 2013). Dentre as diversas oportunidades de melhorias no processo produtivo da empresa BRATAC, pesquisadores do SENAI CIMATEC identificaram a oportunidade de aplicação de um sistema de visão computacional para o controle de qualidade dos casulos que são utilizados na produção do fio de seda.

Um dos processos tradicionalmente empregados na produção de fios de seda é a seleção dos casulos de bicho-da-seda considerados adequados para a produção. Casulos com defeito de formação, mofados ou danificados durante o processo de extração são selecionados manualmente e rejeitados na linha de produção da BRATAC (BRATAC, 2002). O controle de qualidade automatizado poderia diminuir os custos da empresa com pessoal e aumentar a produção, mantendo um padrão de qualidade aceitável (SILVEIRA, 2014).

Para automatizar esse processo, um sistema de visão computacional pode substituir os operadores humanos na identificação dos casulos defeituosos e um sistema robótico pode substituir a tarefa de descartar estes casulos na linha de produção. A figura 1 representa a automação proposta para a linha de produção da BRATAC.

Figura 1 – Automação da linha de produção de fios de seda



Fonte: (BOMFIM, 2015)

Devido à complexidade deste projeto de automação, o estudo pode ser dividido em duas partes: o projeto do sistema de atuação robotizada na linha de produção e o projeto do sistema de visão computacional. O sistema de visão computacional por sua vez pode ser dividido em três etapas: aquisição, segmentação e classificação.

Este trabalho tem como foco o estudo de uma das etapas do sistema de visão computacional, a classificação dos casulos. Outro estudante do curso de especialização em automação e controle no SENAI CIMATEC abordou no seu trabalho de conclusão o subsistema de segmentação de imagens (BOMFIM, 2015). Recomenda-se a leitura deste trabalho em segmentação como material complementar sobre a produção do fio de seda, visão computacional e compreensão dos desafios relacionados especificamente à segmentação de casulos de bicho-da-seda.

1.2 Objetivos

O objetivo geral deste trabalho é *avaliar a capacidade de diferentes descritores de classificar imagens de casulos de bicho-da-seda em casulos considerados adequados para a produção do fio de seda e casulos defeituosos*.

Os objetivos específicos são:

- Criar bancos de imagens dos casulos do bicho-da-seda;
- Desenvolver algoritmos simples de segmentação dos casulos do bicho-da-seda;
- Desenvolver algoritmos de extração de descritores dos casulos de bicho-da-seda;
- Desenvolver algoritmos para avaliação do grau de similaridade entre descritores;
- Desenvolver algoritmos de prova de conceito para a seleção automática de casulos defeituosos;

1.3 Organização da Monografia

A presente monografia possui quatro capítulos: Introdução; Fundamentação Teórica; Metodologia e Resultados Experimentais; e Conclusão. A seguir é apresentado o conteúdo de cada um destes capítulos.

No capítulo 1, Introdução, é apresentado o contexto do trabalho, os seus objetivos e a estrutura da monografia.

No capítulo 2, Fundamentação teórica, é apresentado resumidamente um modelo de um sistema de visão computacional e suas partes constituintes: aquisição, segmentação e classificação. Em seguida são apresentados os tipos e como são calculados os descritores que serão avaliados: descritor de forma, de textura e SIFT. Finalmente os métodos de cálculo de similaridade empregados são apresentados: distância euclidiana e distância de Mahalanobis.

No capítulo 3, Metodologia e Resultados Experimentais, são apresentados os procedimentos e as ferramentas utilizadas para a formação do banco de dados de imagens e o desenvolvimento dos algoritmos de classificação das imagens de casulos.

No capítulo 4, Conclusão e trabalhos futuros, é apresentada uma visão geral do trabalho realizado, dos resultados e das próximas etapas necessárias para o desenvolvimento do sistema de visão computacional.

2 Fundamentação Teórica

2.1 Visão Computacional

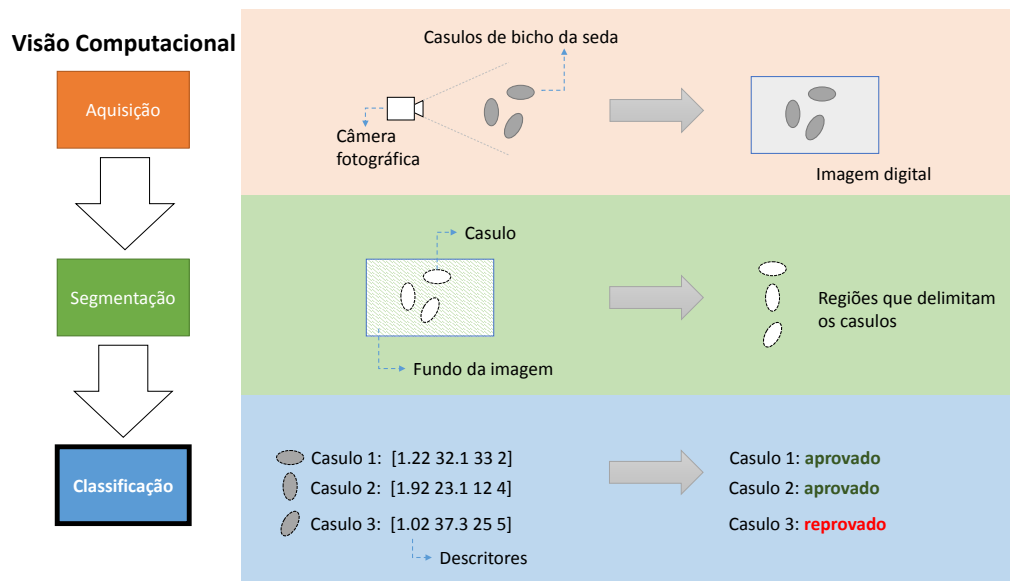
O campo da ciência que trata da aplicação de técnicas de processamento de imagens em problemas práticos é conhecido como visão computacional. Com a popularização de dispositivos de alto poder computacional e sensores de imagem, é cada vez mais comum encontrar soluções baseadas em visão computacional na indústria. Neste trabalho será apresentado um estudo exploratório de processamento de imagem para controle de qualidade com aplicação na indústria de fiação de seda.

O processamento de imagens para o controle de qualidade exige basicamente 3 etapas:

- Aquisição;
- Segmentação;
- Classificação;

A Figura 2 representa um sistema de visão computacional aplicado à seleção de casulos de bicho da seda.

Figura 2 – Etapas da visão computacional aplicada à classificação de casulos



Fonte: Autoria própria

2.1.1 Aquisição

Na etapa de aquisição, uma câmera irá converter a luz proveniente de uma cena num conjunto de dados que representa as informações captadas. Nesta etapa os diversos parâmetros da câmera influenciam na imagem resultante fornecida, por exemplo: resolução da imagem, foco utilizado, distância entre a câmera e o objeto em estudo, balanço do tom de branco, tempo de exposição e a sensibilidade do sensor a faixas do espectro eletromagnético.

A etapa de aquisição pode então ser modelada como um processo caixa-preta onde as grandezas de entrada e saída são dadas por:

Entrada

Luz proveniente da cena (grandezas físicas).

Saída

Imagem digital (ex: matrizes R, G e B de 640 por 480 pixels).

2.1.2 Segmentação

Na etapa de segmentação, a imagem é processada para que o objeto de interesse seja selecionado e extraído do restante da imagem. Este processo de recorte é muito importante para diminuir a interferência do restante da cena na análise do objeto de interesse e assim reduzir os requisitos da etapa de classificação. Para que uma segmentação seja bem sucedida é importante que o algoritmo utilizado explore informações da cena ou do objeto procurado, utilizando estes recursos para aumentar a sua robustez.

Do mesmo modo que a etapa anterior, a etapa de segmentação pode ser modelada como um processo caixa-preta onde as grandezas de entrada e saída são dadas por:

Entrada

Imagem digital adquirida na etapa de aquisição.

Saída

Imagem dos objetos de interesse, recortados da cena.

2.1.3 Classificação

A última etapa é a classificação, onde as imagens dos objetos que foram segmentados da cena são processadas para a extração de parâmetros que os descrevem e para a atribuição dos grupos que cada objeto pertence. Como na etapa de segmentação, é importante a utilização de algoritmos que explorem informações sobre os objetos, aumentando assim a robustez e capacidade de classificação do algoritmo.

Os parâmetros de entrada e saída da etapa de classificação são dados por:

Entrada

Imagem com regiões de interesse recortadas da cena, gerada pela etapa de segmentação.

Saída

Informações quantitativas sobre parâmetros do objeto e a classe que o mesmo pertence.

O enfoque principal deste trabalho será dado a etapa de classificação aplicada ao controle de qualidade de casulos de bicho da seda. Como as etapas de aquisição e segmentação influenciam consideravelmente a etapa classificação, alguns conceitos sobre a metodologia e parâmetros de aquisição e segmentação utilizados serão abordados.

2.2 Descritores

2.2.1 Descritor Elíptico de Fourier

Um dos aspectos facilmente identificáveis que possibilitam a distinção entre casulos considerados adequados e inadequados para a produção de fios de seda é o formato destes casulos. Na figura 3, o casulo da esquerda é adequado para a produção do fio de seda enquanto o casulo da direita é defeituoso. Portanto, um descritor de forma pode ser um parâmetro bastante útil para diferenciar estes grupos de casulos. Dentre os descritores de forma mais populares na literatura, destacam-se dois grupos de descritores, os baseados no contorno do objeto e os baseados na região delimitada pelo objeto (NIXON; AGUADO, 2002).

Figura 3 – Comparação de casulos quanto à forma



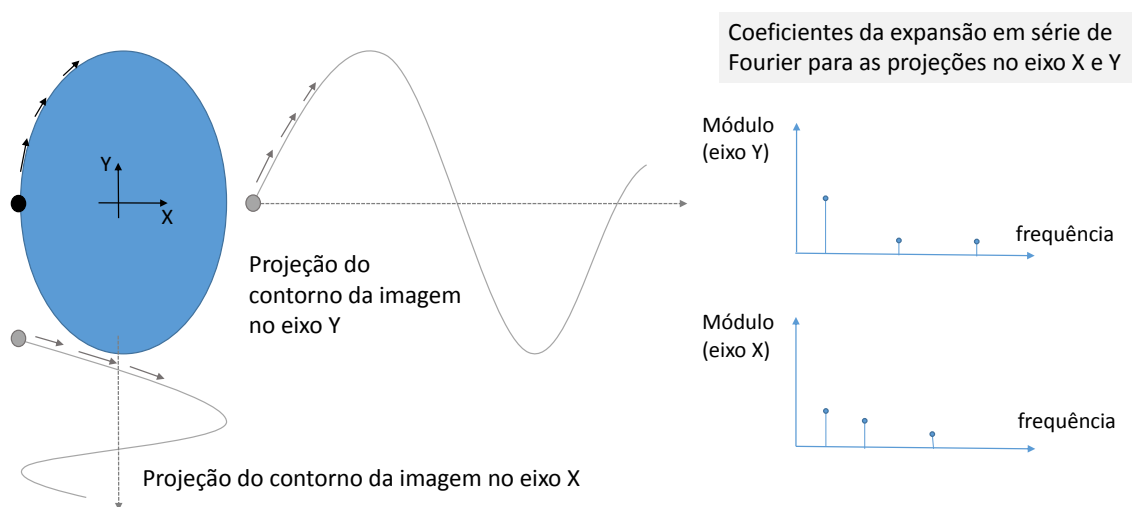
Fonte: Autoria própria

Os descritores baseados em contorno descrevem a forma utilizando a posição dos pixels na região de fronteira do objeto, construindo cadeias de códigos com a informação da posição relativa de cada pixel ou construindo uma curva que é representada por uma aproximação em série de Fourier (NIXON; AGUADO, 2002). Os descritores baseados na região delimitada pelo

objeto podem ser obtidos pelos parâmetros básicos da região (área, perímetro, compactação e dispersão) ou no cálculo de momentos para o conjunto dos pixels delimitados pela região (NIXON; AGUADO, 2002). Para este estudo, optou-se por um descritor baseado no contorno da imagem para a construção de um descritor de forma, o descritor elíptico de Fourier. Este descritor possui duas vantagens principais em relação aos demais, ele possui uma relação clara com o formato do objeto e ele pode ser adaptado para ser invariante à posição, rotação e escala do objeto.

O descritor elíptico de Fourier é obtido pela expansão em série de Fourier das projeções do contorno no eixo X e Y. O número de coeficientes da expansão define a precisão da representação, porém a presença de ruído na imagem e no processo de segmentação inviabiliza a utilização de um número elevado de coeficientes. Por avaliação empírica, nota-se que a utilização de em torno de 10 coeficientes é suficiente para se obter uma aproximação satisfatória. A figura 4 a seguir ilustra a representação da projeção nos eixos X e Y do contorno da imagem e a equação (2.1) representa a expansão em série de Fourier das curvas geradas por estas projeções.

Figura 4 – Extração dos parâmetros para o descritor elíptico de Fourier



Fonte: Autoria própria

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \frac{1}{2} \begin{pmatrix} a_{x0} \\ a_{y0} \end{pmatrix} + \sum_{k=1}^{\infty} \begin{pmatrix} a_{xk} & b_{xk} \\ a_{yk} & b_{yk} \end{pmatrix} \begin{pmatrix} \cos(k\omega t) \\ \sin(k\omega t) \end{pmatrix} \quad (2.1)$$

Onde:

$$\begin{aligned} a_{xk} &= \frac{2}{m} \sum_{i=1}^m x_i \cos(k\omega i\tau) & b_{xk} &= \frac{2}{m} \sum_{i=1}^m x_i \sin(k\omega i\tau) \\ a_{yk} &= \frac{2}{m} \sum_{i=1}^m y_i \cos(k\omega i\tau) & b_{yk} &= \frac{2}{m} \sum_{i=1}^m y_i \sin(k\omega i\tau) \end{aligned} \quad (2.2)$$

Considerando que (x, y) são as coordenadas da amostra pertencente ao contorno do objeto, m é o número total de amostras, k é o índice do coeficiente da expansão em série de fourier e t é um parametro auxiliar que identifica cada pixel do contorno. Para a implementação do algoritmo, considerou-se $\omega = 1$ e $\tau = \frac{2\pi}{m}$.

A última etapa para a criação do descritor elíptico de Fourier é a criação de um vetor que é invariante à posição, rotação e escala. Para isto ignora-se o primeiro termo da expansão (relacionados a a_0 e b_0) pois este está ligado ao centroide do objeto e, portanto, à posição. Quanto ao efeito da rotação no descritor, pode-se obter descritores invariantes à rotação pela utilização da magnitude do coeficiente complexo ($A_k = \sqrt{a_{xk}^2 + a_{yk}^2}$). Finalmente realiza-se a normalização dos demais coeficientes em relação ao segundo termo para se obter um descritor invariante à escala ($\frac{|A_k|}{|A_1|}$).

Para reduzir o tamanho do vetor descritor, soma-se os coeficientes normalizados para cada harmônica. A equação resultante (2.3) define o cálculo de cada termo k do descritor elíptico de fourier D_k .

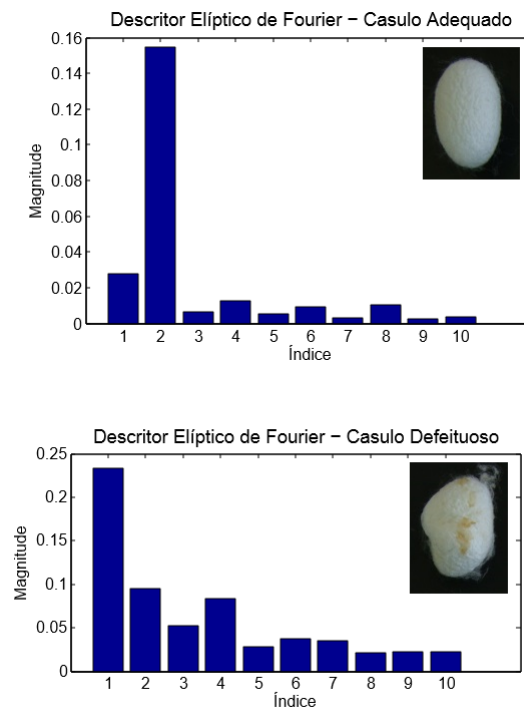
$$D_k = \frac{\sqrt{a_{xk}^2 + a_{yk}^2}}{\sqrt{a_{x1}^2 + a_{y1}^2}} + \frac{\sqrt{b_{xk}^2 + b_{yk}^2}}{\sqrt{b_{x1}^2 + b_{y1}^2}} = \frac{|A_k|}{|A_1|} + \frac{|B_k|}{|B_1|} \quad (2.3)$$

O descritor elíptico de Fourier é então calculado pela realização dos seguintes passos:

1. A partir do objeto segmentado, extrair um vetor de pixels representando o contorno do objeto.
2. Com a posição dos pixels do contorno, descrever as duas curvas de projeção, nos eixos X e Y como ilustrado pela figura 3.
3. Calcular a expansão em série de Fourier para as duas curvas de projeção.
4. Calcular o descritor como um vetor invariante à posição, rotação e escala do objeto. Os termos do descritor são descritos pela equação (2.3).

A figura 5 a seguir ilustra os descritores extraídos de duas imagens de referência.

Figura 5 – Descritores de forma



Fonte: Autoria própria

2.2.2 Descritor de Textura

A textura é descrita como uma propriedade de uma região da imagem que exibe um determinado padrão. Casulos que, por razões diversas, apresentam manchas são considerados inadequados para a produção do fio de seda. A figura 6 ilustra alguns exemplos de casulos com manchas. Portanto, assim como para o descritor de forma, um descritor de textura também pode ser um parâmetro importante para diferenciar casulos com um padrão de textura de referência considerado adequado dos demais inadequados. Existem duas abordagens populares para a extração de descritores de textura, a estatística e a estrutural (NIXON; AGUADO, 2002).

Figura 6 – Casulos defeituosos com manchas

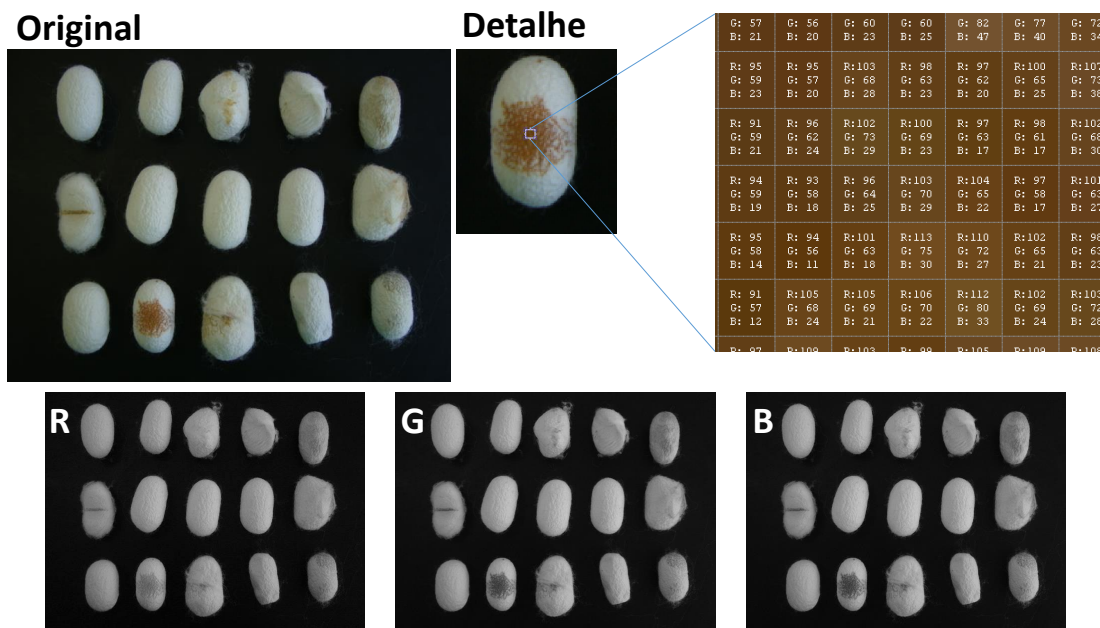


Fonte: Autoria própria

Uma textura grosseira pode ser comparada com uma textura fina em função da frequência e amplitude da variação de intensidade dos pixels numa imagem. A abordagem estatística busca descrever estas texturas como as propriedades da distribuição de intensidade dos pixels como, por exemplo, a média e a variância da intensidade dos pixels. Por outro lado, a abordagem estrutural considera a textura como uma estrutura de referência que se repete espacialmente na imagem (NIXON; AGUADO, 2002).

Devido à coloração avermelhada de algumas das manchas encontradas, o quadro azul da imagem dos casulos é o que fornece o melhor contraste para as manchas. A figura 7 ilustra os quadros RGB para a imagem de teste do banco de dados 2, mostrando os valores dos pixels para uma região típica de mancha. Por esta razão o equivalente ao quadro azul em tons de cinza foi selecionado como imagem para a extração dos parâmetros de textura.

Figura 7 – Casulos defeituosos com manchas: Quadros RGB



Fonte: Autoria própria

Uma maneira simples de se obter descritores de textura é se calcular alguns parâmetros estatísticos do histograma de uma determinada região (SCHWARTZ; SIQUEIRA; PEDRINI, 2012). Seis parâmetros estatísticos baseados no histograma da imagem dos casulos foram selecionados para a extração do descritor de textura: média, variância, curtose, obliquidade, energia e entropia. Estes parâmetros são baseados em estatística de primeira ordem, ou seja, a correlação da intensidade dos pixels dois a dois não é levada em conta.

A média fornece um valor de intensidade de referência para o casulo, que é afetado pela presença de manchas. Por outro lado, esse descritor também é bastante sensível a variações de iluminação na fotografia. A variância fornece um valor relacionado à dispersão em torno da

média dos diversos valores de intensidade dos pixels. Este descritor está relacionado à rugosidade do casulo e também pode ser afetado pela variação de iluminação e presença de regiões de sombra no casulo.

Curtose é uma outra medida estatística da dispersão de valores de intensidade em torno da média e está relacionada à forma de pico ou achatamento numa distribuição normal (NIST/SEMATECH, 2015). Este descritor é complementar à variância e também é suscetível às variações na iluminação. A obliquidade é uma medida estatística relacionada à assimetria da distribuição normal (NIST/SEMATECH, 2015). Este descritor pode indicar características de textura complementares à curtose e à variância, porém regiões de sombra podem afetar este parâmetro.

A normalização do histograma da imagem fornece uma aproximação para a função densidade de probabilidade para os valores de intensidade dos pixels. A equação (2.4) fornece a probabilidade P_i para cada intensidade i de um pixel. A seguir as equações (2.5) até (2.8) descrevem o cálculo dos parâmetros estatísticos para m pixels de intensidade I , assim como o cálculo destes componentes a partir do histograma de N regiões de valores de intensidade B_N com o número de ocorrências C_i para cada uma destas regiões.

$$P_i = \frac{C_i}{\sum_{k=1}^N C_k} \quad (2.4)$$

$$\mu = \frac{1}{m} \sum_{k=1}^m I_k = \sum_{k=1}^N B_k P_k \quad (2.5)$$

$$\sigma^2 = \frac{1}{m} \sum_{k=1}^m (I_k - \mu)^2 = \sum_{k=1}^N P_k (B_k - \mu)^2 \quad (2.6)$$

$$S = \frac{\frac{1}{m} \sum_{k=1}^m (I - \mu)^3}{\sigma^3} = \frac{\sum_{k=1}^N P_k (B_k - \mu)^3}{\sigma^3} \quad (2.7)$$

$$K = \frac{\frac{1}{m} \sum_{k=1}^m (I - \mu)^4}{\sigma^4} = \frac{\sum_{k=1}^N P_k (B_k - \mu)^4}{\sigma^4} \quad (2.8)$$

Energia é uma medida relacionada ao quadrado da ocorrência relativa de um pixel. É um parâmetro útil para caracterizar a rugosidade de uma textura, porém, assim como os demais parâmetros, é bastante suscetível a mudanças de iluminação.

O parâmetro entropia está ligado a quantidade de informação numa imagem. Quanto mais complexa a distribuição dos pixels, maior será este parâmetro. Regiões com pixels de intensidade constante resultam em parâmetros de entropia baixos.

A seguir o cálculo da energia e da entropia para um histograma dividido em L faixas onde P_i é o número de pixels da imagem que tem intensidade na faixa i .

$$Energia = \sum_{i=1}^N P_i * P_i \quad (2.9)$$

$$Entropia = - \sum_{i=0}^N P_i * \log_2 P_i \quad (2.10)$$

Todos estes parametros baseados no histograma da imagem segmentada formam um vetor de descritores de textura que serão utilizados para classificar os casulos de bicho-da-seda.

2.2.3 Descritor SIFT

SIFT (*Scale Invariant Feature Transform*) é um algoritmo desenvolvido por David Lowe (LOWE, 2004) que envolve a detecção e a descrição de regiões de interesse numa imagem. Este algoritmo tem sido extensivamente utilizado na área de visão computacional e realiza de modo eficiente a identificação de padrões locais relacionados a objetos específicos.

A primeira etapa do algoritmo SIFT é a identificação de regiões de máximo no espaço de escala, ou seja, pontos que possuem mais informação e possam estar relacionados a assinaturas da imagem. A detecção destes pontos é equivalente a detecção de bordas e estes são candidatos a regiões de interesse que terão suas características descritas.

Para construção de um espaço de escala, deve-se obter um conjunto de laplacianas de gaussianas de uma imagem em diferentes dimensões e um modo computacionalmente eficiente de obter estas imagens filtradas é por diferenças de gaussianas. A diferença de gaussianas pode ser interpretada como uma aproximação de um filtro passa-faixa pela conexão em cascata de dois filtros passa-baixa.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (2.11)$$

Onde,

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (2.12)$$

A Laplaciana pode ser aproximada por:

$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \quad (2.13)$$

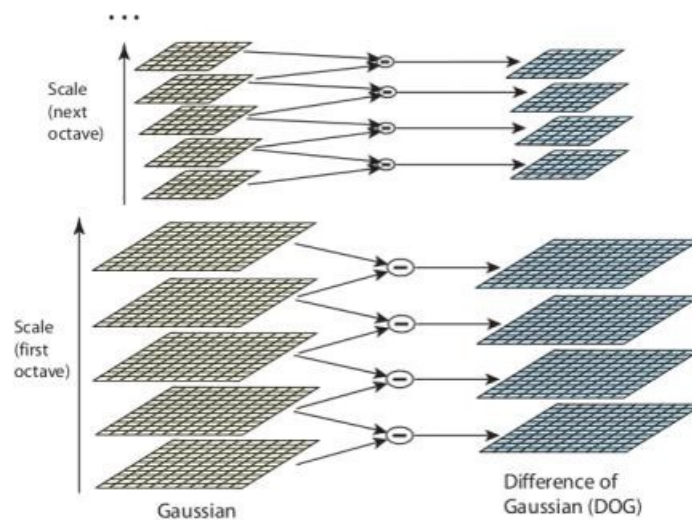
$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G \quad (2.14)$$

Logo, a diferença de gaussianas é expressa como:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (2.15)$$

Logo, para cada escala são geradas imagens filtradas por um kernel gaussiano em diferentes valores de sigma, seguido da realização da diferença destas imagens. A figura 8 a seguir ilustra o processo de geração do espaço de escalas.

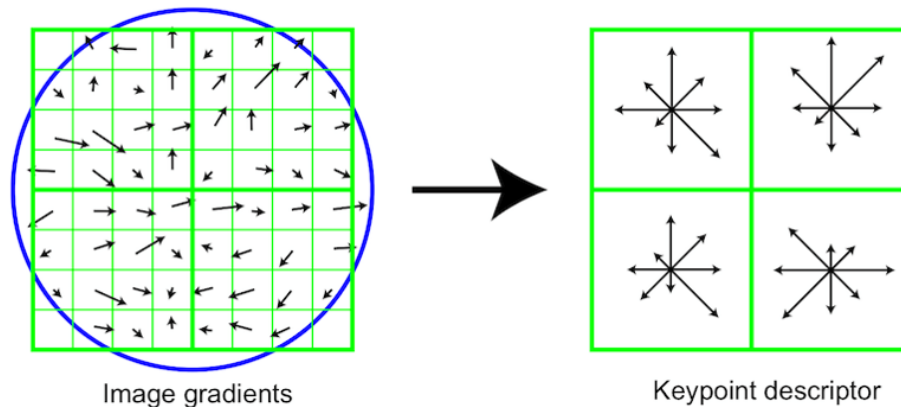
Figura 8 – Espaço de escalas por Diferença de Gaussianas



Fonte: (LOWE, 2004)

Após selecionar as regiões de interesse, excluindo regiões redundantes e outras possivelmente causadas por ruído, é realizado um histograma espacial dos gradientes em 8 direções para 16 regiões diferentes em torno da região selecionada. Para cada região de interesse é formado um descritor utilizando estes histogramas, sendo, portanto, o descritor SIFT um vetor de 128 posições. A figura 9 a seguir ilustra a formação dos descritores SIFT como o histograma de gradientes locais. Para a classificação dos casulos, foi formado um banco de dados de referência, contendo descritores de características extraídas de casulos considerados adequados.

Figura 9 – Descritor SIFT: histograma de gradientes



Fonte: (LOWE, 2004)

Uma vantagem do descritor SIFT é o nível de automatização do algoritmo, permitindo a extração de características locais automaticamente e assim a comparação de imagens sem a necessidade de descrever exatamente qual característica é a mais relevante. Por outro lado, essa automatização acaba diminuindo a sensibilidade à interpretação dos resultados.

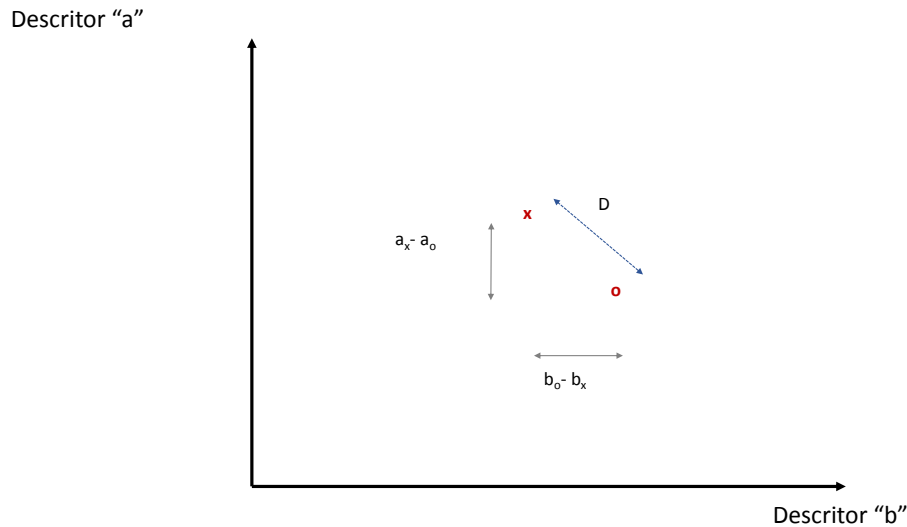
2.3 Cálculos de similaridade

Após extrair os descritores das imagens, existe o problema de identificar se esse descritor representa um casulo adequado ou não à produção do fio de seda. Para este fim, foi selecionado um grupo de imagens de casulos considerados adequados, ou seja, um grupo de referência para comparar casulos de teste e decidir se estes são adequados ou não.

2.3.1 Distância Euclidiana

A distância euclidiana é a forma mais simples de calcular a similaridade entre dois vetores e consiste na raiz quadrada da soma dos quadrados das distâncias em cada dimensão. A distância euclidiana é usualmente utilizada para calcular distâncias espaciais em duas ou três dimensões, mas pode ser estendida para qualquer número de dimensões, bastando para isso adicionar os termos referentes a cada dimensão no somatório, como indicado na equação (2.16). A figura 10 ilustra a distância euclidiana entre duas amostras, representadas por um círculo e um x.

Figura 10 – Distância Euclidiana



Fonte: Autoria própria

$$D_{euclidiana} = \sqrt{\sum (\text{descritor}_{amostra1} - \text{descritor}_{amostra2})^2} \quad (2.16)$$

No caso da figura 10, a distância euclidiana seria calculada como:

$$D = \sqrt{(a_x - a_o)^2 + (b_x - b_o)^2} \quad (2.17)$$

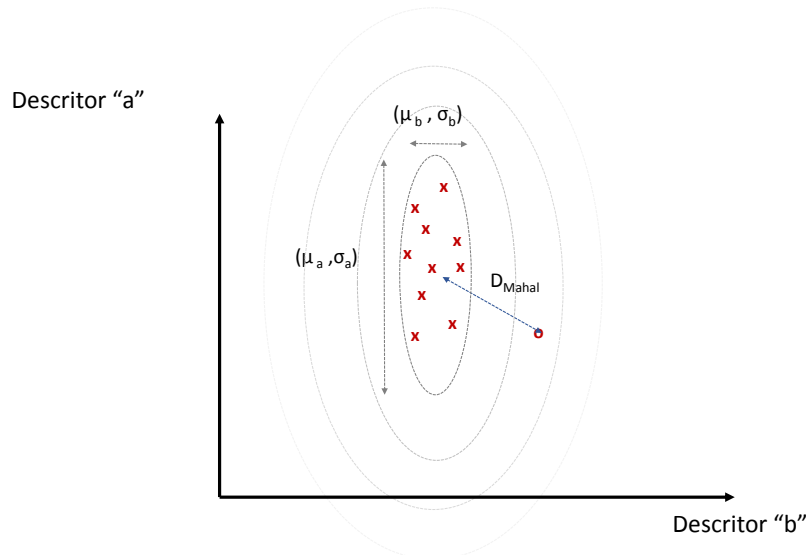
Para os descritores SIFT, onde cada vetor representa apenas uma característica local da imagem, decidiu-se pela utilização do método do vizinho mais próximo. Neste método, calcula-se a distância euclidiana entre a amostra de teste e cada um dos elementos do grupo de referência, selecionando então a menor distância. Para todos os descritores da imagem de teste, calcula-se o vizinho mais próximo e depois realiza-se a média destes valores. Esta média representa então o grau de similaridade da imagem de teste com o grupo de casulos bons. Este método foi escolhido por permitir uma melhor comparação para descritores locais que podem não estar representando a mesma característica.

2.3.2 Distância de Mahalanobis

Para comparar o casulo de teste com o conjunto de casulos bons, foi utilizado um método estatístico chamado distância de Mahalanobis. Este método utiliza a informação de dispersão do grupo de referência para normalizar a distância em relação ao desvio padrão das amostras para todas as dimensões. A figura 11 ilustra o cálculo da distância entre a amostra de teste representada por um círculo e um grupo de referência representados por x. Digamos que a

dispersão no eixo do descritor “a” é maior do que para o eixo do descritor “b”, então a distância entre o centroide da distribuição das amostras de referência para a amostra de teste tem um peso maior para desvios ao longo do eixo do descritor “b”. As elipses representam posições equidistantes do centroide da distribuição de amostras representadas por x , segundo a distância de Mahalanobis.

Figura 11 – Distância de Mahalanobis



Fonte: Autoria própria

A equação a seguir mostra a definição da distância de Mahalanobis de um vetor qualquer x para uma distribuição de vetor de médias μ e matriz de covariância S .

$$D_{Mahal}(x) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)} \quad (2.18)$$

Caso a matriz de covariância seja diagonal, o cálculo da distância se simplifica para a distância euclidiana normalizada pelo desvio padrão de cada dimensão.

$$D_{Mahal}(x) = \sqrt{\sum \frac{(x - \mu)^2}{\sigma}} \quad (2.19)$$

A distância de Mahalanobis foi utilizada para calcular a similaridade entre o casulo de teste e o grupo de referência para os descritores de forma e textura.

3 Metodologia e Resultados Experimentais

3.1 Formação dos bancos de dados de imagens

Uma cadeia de visão computacional completa foi desenvolvida para avaliar a capacidade dos descritores selecionados em diferenciar casulos considerados adequados à produção do fio de seda aos não adequados. Foram formados bancos de dados com imagens de casulos e estas imagens foram segmentadas e classificadas por algoritmos na plataforma Matlab.

A primeira etapa de criação do sistema de visão computacional foi a formação de um banco de dados de imagens de casulos considerados adequados e inadequados para a produção do fio de seda. O banco de dados 1 foi formado a partir de duas imagens enviadas da linha de produção da empresa Bratac. Uma das imagens contém 19 casulos considerados adequados e a outra imagem contém 20 casulos defeituosos. As imagens a seguir na figura 12 formam o banco de dados 1. Este banco de dados foi utilizado apenas como prova de conceito para o algoritmo de classificação.

Figura 12 – Banco de dados 1: casulos adequados e defeituosos



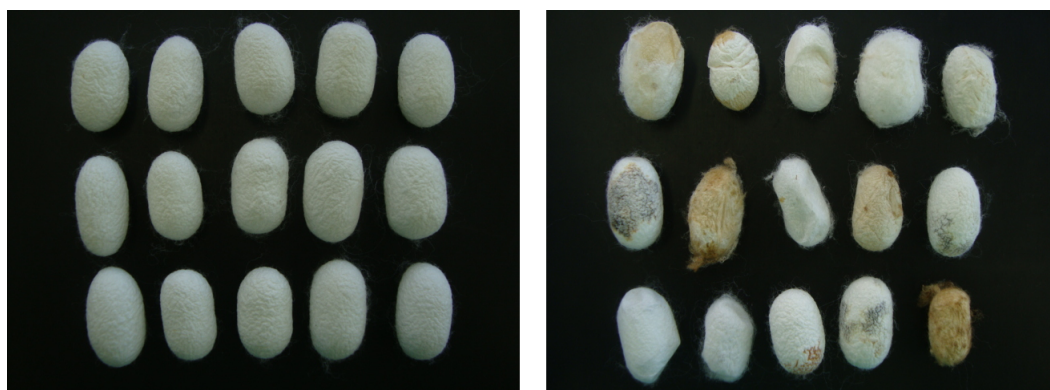
Fonte: Autoria própria

Estas imagens foram fotografadas com um celular modelo iphone 5 e salvas em formato jpeg com resolução 480x640 pixels. O tempo de exposição foi 1/20 segundos para os casulos adequados (em sensibilidade ISO-320) e 1/17 segundos para os defeituosos (em sensibilidade ISO-400). Não foi utilizado Flash em nenhuma das fotografias e o balanço da cor branca foi ajustado automaticamente.

Os bancos de dados 2 e 3 foram formados a partir de fotos de um conjunto de casulos enviados pela empresa Bratac para o SENAI CIMATEC. A diferença entre os bancos de dados 2 e 3 é unicamente o tempo de exposição da câmera digital utilizada para fotografar os casulos, sendo este tempo de exposição maior para o casulo 3. Noventa casulos considerados adequados

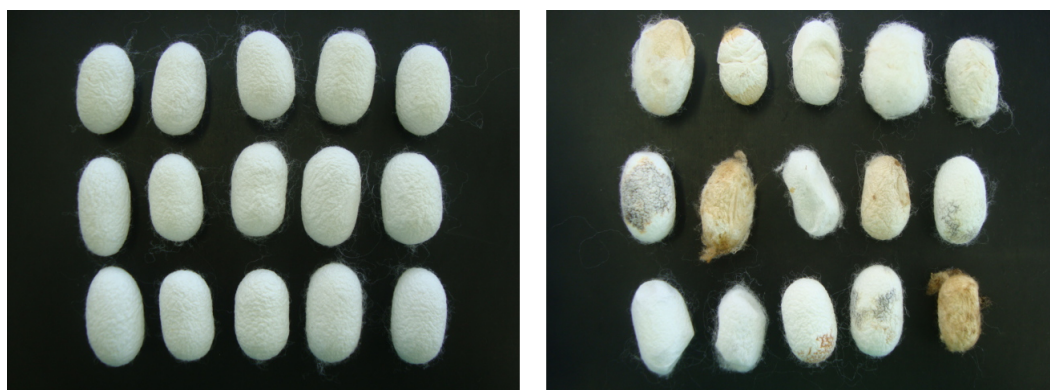
e quinze casulos inadequados foram fotografados em grupos de quinze unidades. Uma imagem de teste de quinze casulos, contendo uma mistura de casulos adequados e inadequados, foi utilizada como teste para validar a capacidade de seleção do algoritmo. Todos os casulos foram fotografados apenas uma única vez em cada banco de dados. As figuras 13 e 14 ilustram os bancos de dados 2 e 3 respectivamente.

Figura 13 – Banco de dados 2: casulos adequados e defeituosos



Fonte: Autoria própria

Figura 14 – Banco de dados 3: casulos adequados e defeituosos



Fonte: Autoria própria

Estas imagens foram fotografadas com uma câmera Sony modelo DSC-W130 e salvas em formato jpeg com resolução 640x480 pixels. O tempo de exposição foi entre 1/250 até 1/160 segundos para os casulos do banco de dados 2 e entre 1/160 até 1/100 segundos para os casulos do banco de dados 3. A sensibilidade utilizada para o sensor foi ISO-125. Não foi utilizado Flash em nenhuma das fotografias e o balanço da cor branca foi ajustado automaticamente.

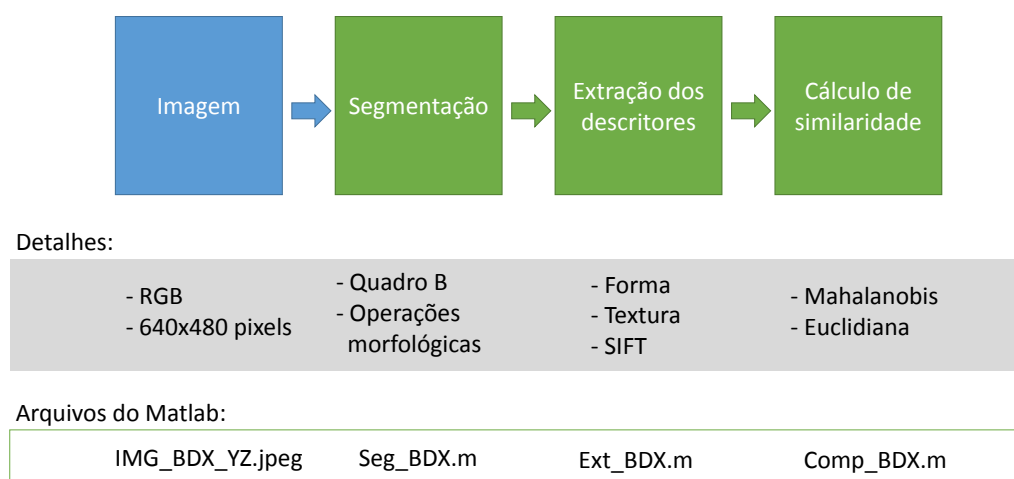
A tabela 1 a seguir resume o número total de casulos de referência, avaliação e identificação utilizados para cada banco de dados.

Tabela 1 – Distribuição dos grupos de imagens de casulos de bicho-da-seda

Imagens	Referência	Avaliação (Bons)	Avaliação (Defeituosos)	Identificação
Banco 1	10 casulos	19 casulos	20 casulos	nenhum
Banco 2	60 casulos	30 casulos	15 casulos	15 casulos
Banco 3	60 casulos	30 casulos	15 casulos	15 casulos

Utilizando estes bancos de dados, uma cadeia de processamento realizada no software Matlab foi realizada para classificar os casulos. A figura 15 a seguir representa a cadeia de processamento e os arquivos utilizados.

Figura 15 – Cadeia de processamento de dados para classificação de casulos

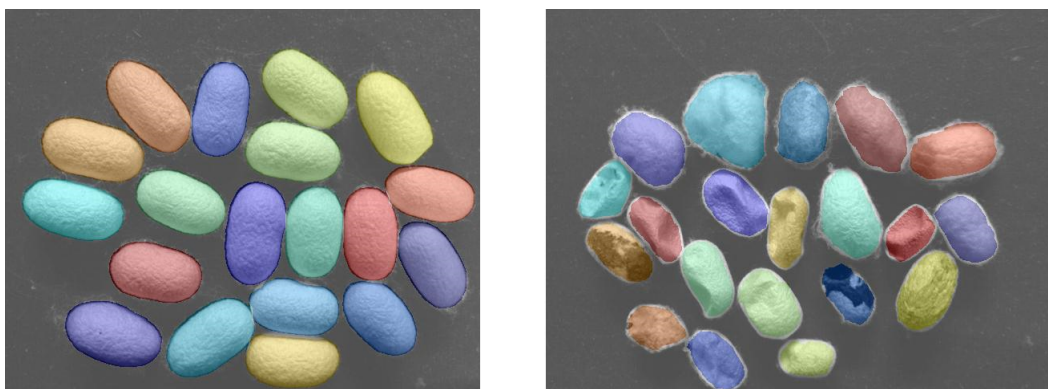


Fonte: Autoria própria

3.2 Segmentação das imagens

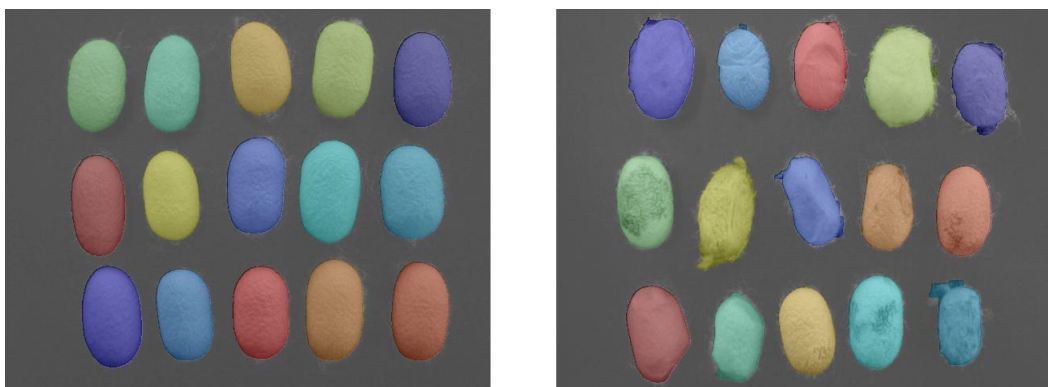
Para a parte de segmentação da imagem, foram utilizadas somente operações morfológicas para as imagens do banco de dados 2 e 3, onde os casulos não se tocam. Para o banco de dados 1 foi necessário utilizar o algoritmo watershed pois foi constatado que utilizar somente operações morfológicas não é suficiente para separar as regiões de maneira eficaz quando os casulos estão se tocando. Como a etapa de segmentação não é o foco deste trabalho, os algoritmos utilizados não serão discutidos em profundidade (Para mais detalhes sobre processos de segmentação ver (BOMFIM, 2015)). As imagens a seguir (Figuras 16, 17 e 18) ilustram o resultado do processo de segmentação, onde cada coloração representa uma diferente região segmentada.

Figura 16 – Banco de dados 1: segmentação dos casulos



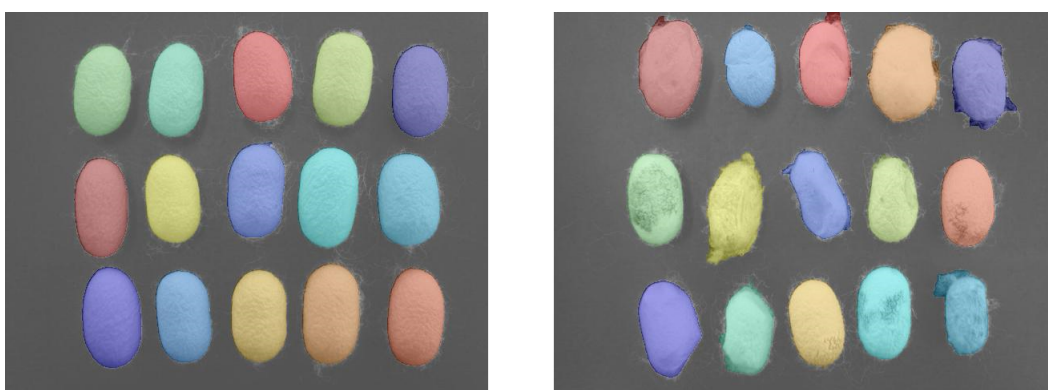
Fonte: Autoria própria

Figura 17 – Banco de dados 2: segmentação dos casulos



Fonte: Autoria própria

Figura 18 – Banco de dados 3: segmentação dos casulos



Fonte: Autoria própria

3.3 Extração dos descritores

Após a segmentação, um algoritmo para extração dos descritores foi utilizado. Para o descritor de forma, o descritor elíptico de Fourier foi extraído com base no algoritmo ilustrado no livro (NIXON; AGUADO, 2002). Foram adotados dez coeficientes da expansão em série de Fourier como uma aproximação satisfatória da forma do casulo, considerando que os coeficientes mais elevados apenas acrescentam ruído à descrição de forma e que as limitações no algoritmo de segmentação também limitam a relevância de termos de ordem elevada.

O descritor de textura é calculado a partir do histograma da imagem, obtendo os parâmetros média, variância, curtose, obliquidade, energia e entropia. Estes parâmetros são derivados da estatística de primeira ordem (SCHWARTZ; SIQUEIRA; PEDRINI, 2012) (como discutido no capítulo "Fundamentação Teórica", subseção "Descritores", parte "Descritores de textura"). Estes descritores podem identificar defeitos relacionados a manchas ou diferenças de rugosidade da superfície do casulo, porém também são afetados por variações de iluminação, presença de sombras e pela segmentação do casulo.

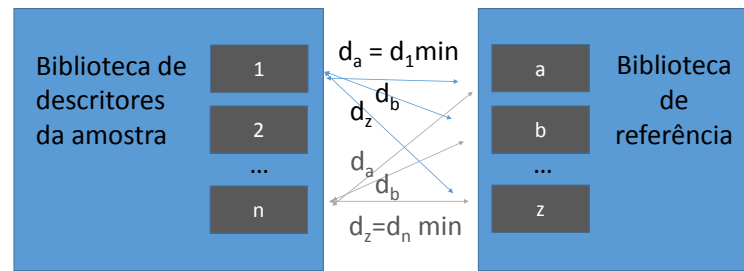
O descritor SIFT é calculado com a ajuda de uma biblioteca em código aberto para Matlab conhecida como VLFeat (VEDALDI; FULKERSON, 2008). Esta biblioteca é mantida por pesquisadores de diversas universidades como Oxford, KU Leuven, Czech Technical University in Prague, UCLA, Universidade de Surrey e Universidade de Grenoble Alpes. Para cada imagem de casulo após a segmentação, foi aplicado a função `vl_sift()`, retornando um conjunto de descritores SIFT.

3.4 Cálculo de Similaridade e Classificação

Para cada grupo de referência de imagens de casulos bons, foi formado um conjunto de descritores de forma, textura e SIFT, contendo todos os descritores dos casulos individuais destas imagens. A classificação dos casulos é então obtida pela avaliação da similaridade entre os descritores do casulo em avaliação e o grupo de descritores extraídos dos casulos de referência. Para os descritores de forma e textura é utilizada a distância de Mahalanobis para avaliar a similaridade entre os descritores do casulo de avaliação e o grupo de referência. Para o descritor SIFT, é utilizada a média dos vizinhos mais próximos entre os descritores do casulo de avaliação e todos os casulos do grupo de referência. A figura 19 a seguir ilustra como foi calculado o grau de similaridade de uma amostra de avaliação para o grupo de referência para o descritor SIFT.

A fase final da classificação é utilizar esse parâmetro de similaridade para dividir os casulos avaliados em dois grupos, os casulos considerados adequados para a produção de fios de seda e os casulos defeituosos e, portanto, inadequados para a produção. Uma das maneiras de realizar esta separação é escolher um limiar para o grau de similaridade entre o casulo sob avaliação e o grupo de referência, onde casulos são considerados adequados se o parâmetro de

Figura 19 – Média das distâncias mínimas para os descritores SIFT



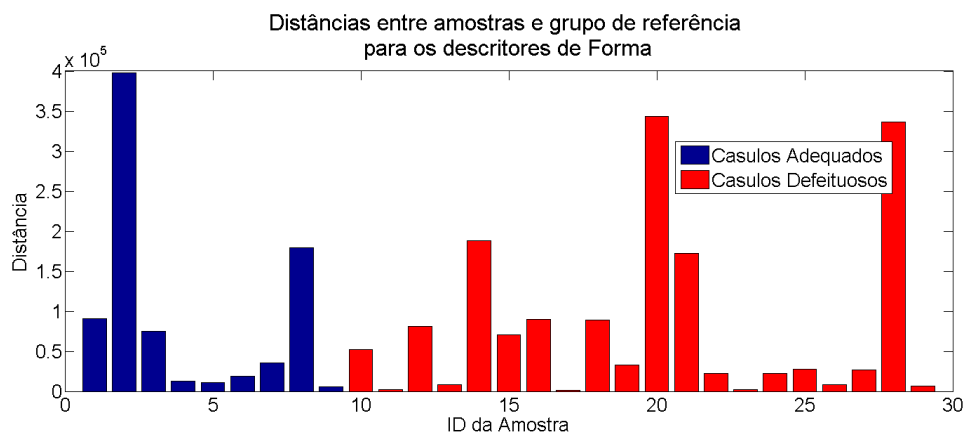
$$D = (d_1 \text{ min} + d_2 \text{ min} + \dots + d_n \text{ min})/n$$

Fonte: Autoria própria

similaridade (distância de Mahalanobis ou a média da distância euclidiana dos vizinhos mais próximos) é menor do que um determinado valor. Este valor pode ser determinado de maneira empírica, escolhendo-se um valor de limiar que não produz falsos positivos para um conjunto de casulos de teste. Outra possibilidade é assumir que as distâncias dos casulos de teste para o grupo de referência formam uma distribuição gaussiana e que a escolha de um limiar de três vezes o valor do desvio padrão desta distância garante que a maioria dos casulos bons serão considerados adequados.

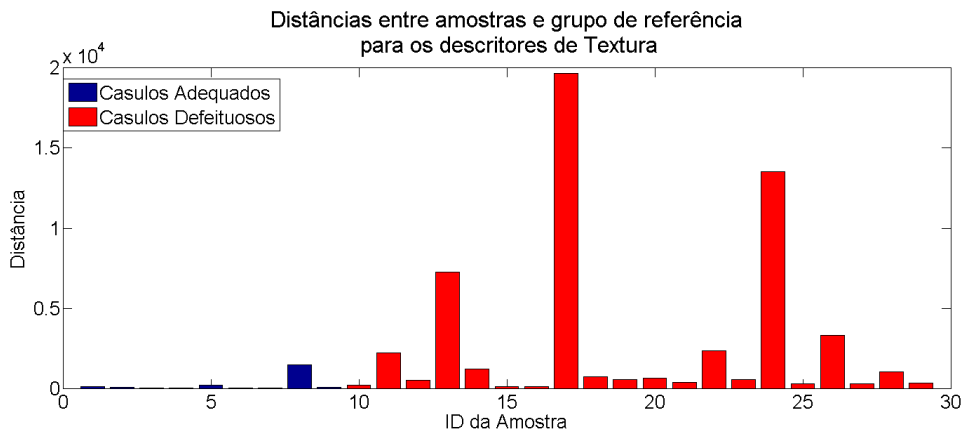
As figuras 20, 21 e 22 a seguir ilustram as distâncias calculadas entre o grupo de referência e o grupo de avaliação para os descritores de textura, forma e SIFT para o banco de dados 1. O grupo de referência foi formado com 10 casulos da imagem dos casulos bons e os casulos de avaliação contam com 9 casulos adequados e 20 casulos inadequados.

Figura 20 – Distância de Mahalanobis para os descritores de Forma



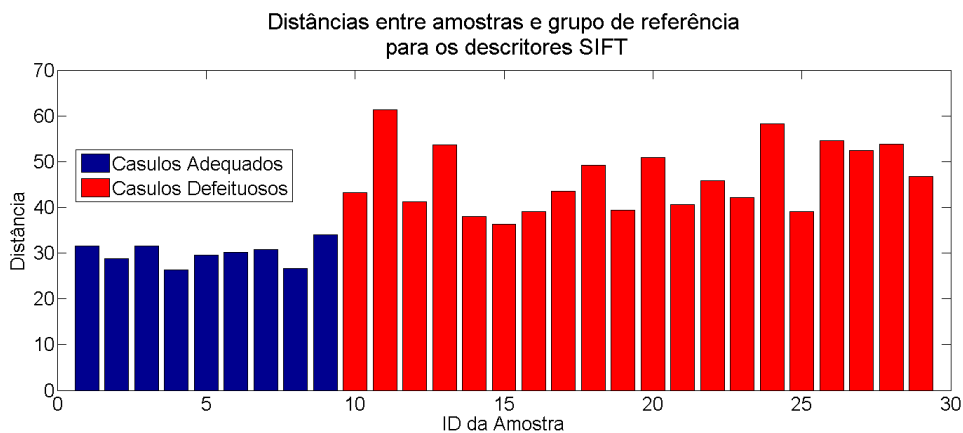
Fonte: Autoria própria

Figura 21 – Distância de Mahalanobis para os descritores de Textura



Fonte: Autoria própria

Figura 22 – Médias das distâncias Euclidianas mínimas para os descritores SIFT



Fonte: Autoria própria

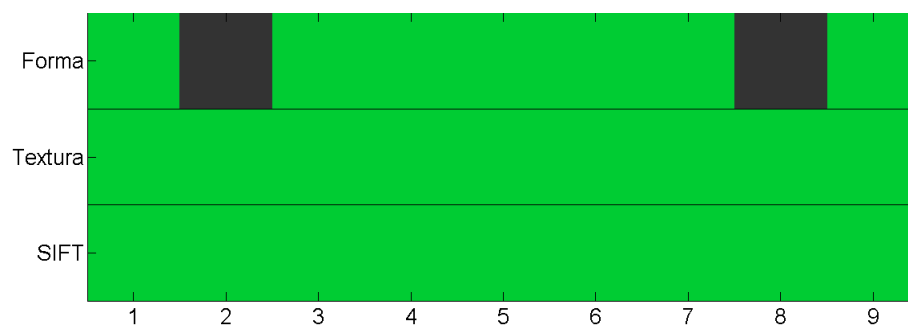
Definindo empiricamente, por exemplo, um limiar de 100000 para o descritor de forma, 1500 para o descritor de textura e 35 para o descritor SIFT, podemos testar a capacidade de seleção destes descritores para os grupos de avaliação utilizados anteriormente. Na figura 23 é ilustrada a classificação dos casulos adequados em relação ao grupo de referência (o quadro verde significa que o casulo foi classificado como adequado e o quadro vermelho significa que o casulo foi classificado como defeituoso) e no quadro da figura 24 é apresentado para cada casulo, quais descritores selecionam o casulo como adequado (verde) ou defeituoso (cinza). Verifica-se que o presente limiar de forma seleciona dois casulos adequados como defeituosos (Número 2 e 8), isso pode estar relacionado à presença de regiões de sombra que provocam uma segmentação equivocada destes casulos e geram portanto um formato discrepante para estes casulos. Outra possibilidade é que a segmentação por watershed para separar os casulos 6 e 8 possa ter contribuído para uma segmentação diferenciada do casulo 8.

Figura 23 – Classificação de casulos considerados adequados



Fonte: Autoria própria

Figura 24 – Quadro de classificação de casulos considerados adequados



Fonte: Autoria própria

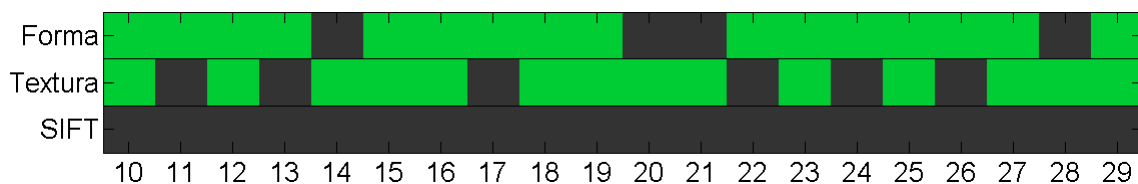
A seguir a imagem e o quadro de classificação para os casulos defeituosos, ilustrados nas figuras 25 e 26.

Figura 25 – Classificação de casulos considerados defeituosos



Fonte: Autoria própria

Figura 26 – Quadro de classificação de casulos considerados defeituosos



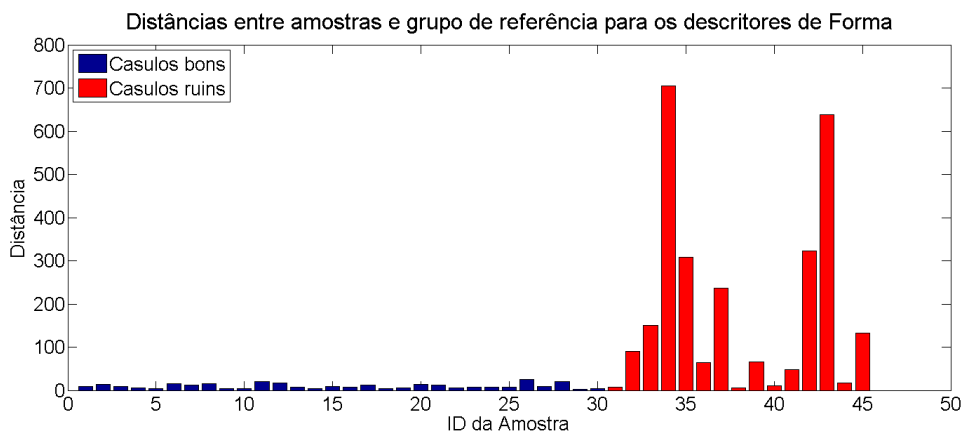
Fonte: Autoria própria

Pode-se notar que o descritor SIFT permite a seleção completa entre os casulos adequados e os casulos inadequados para este banco de dados, enquanto os descritores de textura e forma não tem desempenho satisfatório. Como existiam poucos casulos no grupo de referência, os cálculos utilizando a distância de Mahalanobis não são muito efetivos, logo, a medida de similaridade para os descritores de textura e forma não foi satisfatória. Por outro lado, a comparação de semelhança por distância euclidiana é realizada entre apenas duas amostras e não sofre a mesma influência que o cálculo da distância de Mahalanobis em relação ao número de amostras de referência, fornecendo, portanto, resultados mais confiáveis. O descritor SIFT é normalmente utilizado com uma única imagem de referência, podendo identificar uma característica de um objeto mesmo este estando presente em uma cena complexa. Outro fator que pode ter influenciado o

desempenho do descritor SIFT é o fato do algoritmo de segmentação dos casulos adequados e dos casulos inadequados serem diferentes e o SIFT pode ter identificado características das bordas dos casulos que são diferentes para os dois grupos de imagens. Este banco de dados serviu ao propósito de testar a cadeia completa de extração de descritores e cálculo de similaridade e fornecer resultados preliminares para o sistema de visão computacional.

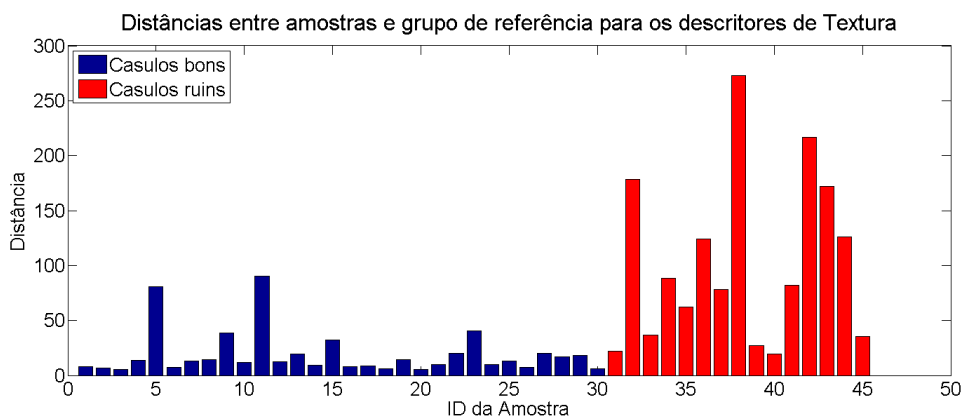
As figuras 27, 28 e 29 a seguir ilustram as distâncias calculadas entre o grupo de referência e o grupo de avaliação para os descritores de textura, forma e SIFT para o banco de dados 2. O grupo de referência foi formado com imagens de 60 casulos considerados adequados enquanto o grupo de avaliação conta com 30 imagens de casulos adequados e 15 de casulos inadequados.

Figura 27 – Distância de Mahalanobis para os descritores de Forma - BD2



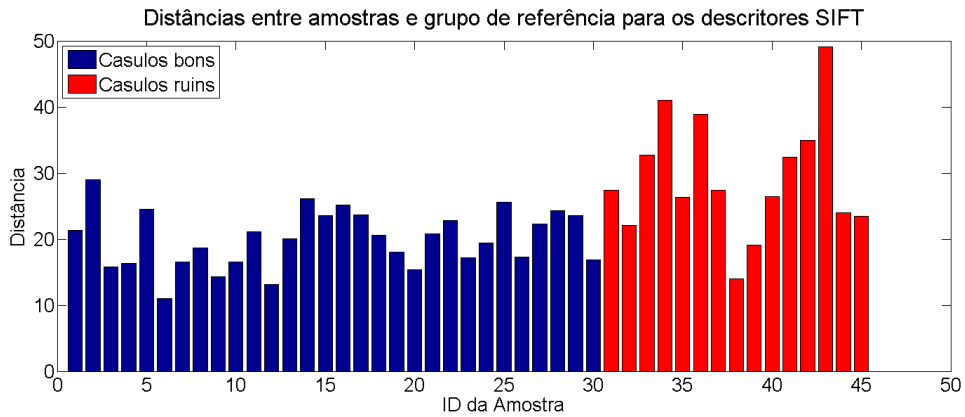
Fonte: Autoria própria

Figura 28 – Distância de Mahalanobis para os descritores de Textura - BD2



Fonte: Autoria própria

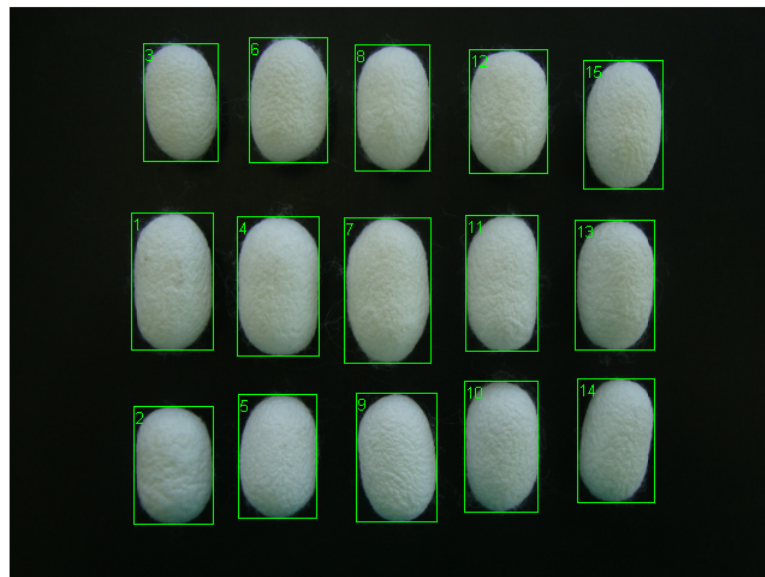
Figura 29 – Médias das distâncias Euclidianas mínimas para os descritores SIFT - BD2



Fonte: Autoria própria

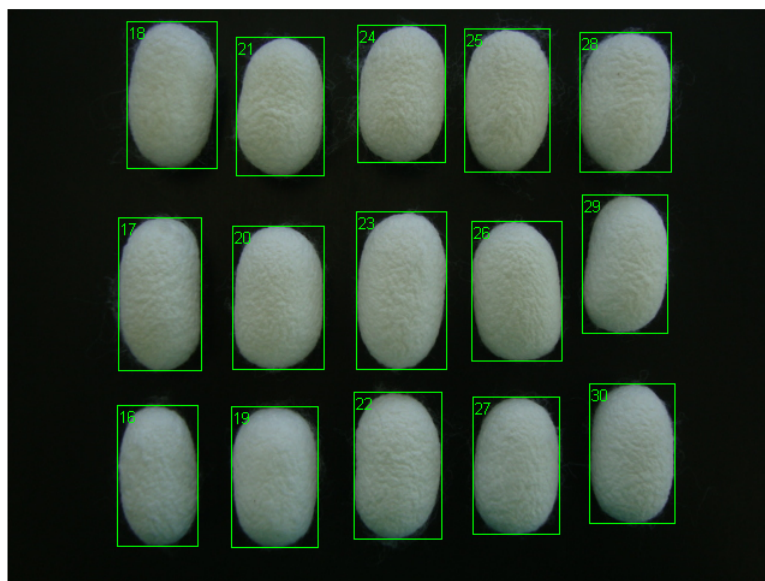
A seguir definiu-se o limiar de distância para a seleção de casulos do banco de dados 2 em 26 para o descritor de forma, 91 para o descritor de textura e 30 para o descritor SIFT e verificou-se a capacidade de seleção de cada descritor como feito anteriormente para o banco de dados 1.

Figura 30 – Classificação de casulos adequados (imagem 1) - BD2



Fonte: Autoria própria

Figura 31 – Classificação de casulos adequados (imagem 2) - BD2



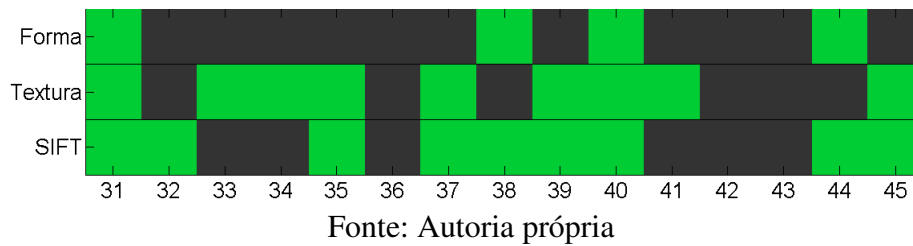
Fonte: Autoria própria

Figura 32 – Classificação de casulos defeituosos - BD2



Fonte: Autoria própria

Figura 33 – Quadro da classificação dos casulos defeituosos- BD2



Para o banco de dados 2, os descritores de forma e textura apresentaram um bom desempenho para realizar a classificação dos casulos considerados adequados e dos casulos inadequados. É importante notar que os descritores de forma e textura têm atuação complementar e juntos selecionam 13 casulos do total de 15 caulos (identificação de 86.7% dos casulos). O descritor SIFT selecionou características redundantes ao descritor de forma para a presente imagem.

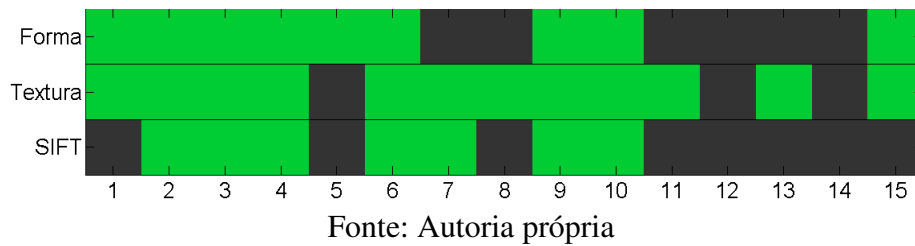
Finalmente, para demonstrar a capacidade de seleção do algoritmo desenvolvido, utilizou-se uma imagem com casulos adequados e defeituosos para a identificação automática dos casulos que possuíssem baixa semelhança com o banco de dados de referência. A seguir a imagem resultante do algoritmo de classificação para o banco de dados 2. Os limiares escolhidos empiricamente foram: Forma = 16; textura = 320; e SIFT = 24.

Figura 34 – Classificação automática dos casulos - BD2



Fonte: Autoria própria

Figura 35 – Quadro da classificação automática dos casulos - BD2



O algoritmo utilizado mostra a capacidade de classificar corretamente 100% dos casulos de teste para esta imagem. Neste caso o descritor SIFT e o descritor de forma são suficientes para diferenciar os casulos adequados dos inadequados e o descritor de textura apresenta informação redundante ao descritor SIFT. Mais uma vez a combinação dos descritores mostra uma capacidade robusta de classificação da maior parte das imagens de caulos.

As figuras 36, 37 e 38 a seguir ilustram as distâncias calculadas entre o grupo de referência e o grupo de avaliação para os descritores de textura, forma e SIFT para o banco de dados 3. O grupo de referência foi formado com imagens de 60 casulos considerados adequados enquanto o grupo de avaliação conta com 30 imagens de casulos adequados e 15 de casulos inadequados.

Figura 36 – Distância de Mahalanobis para os descritores de Forma - BD3

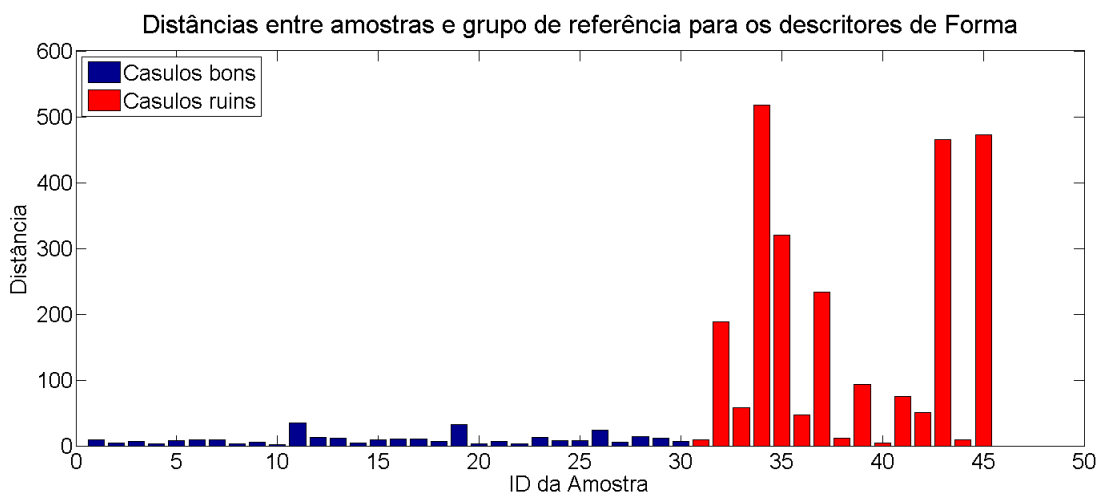
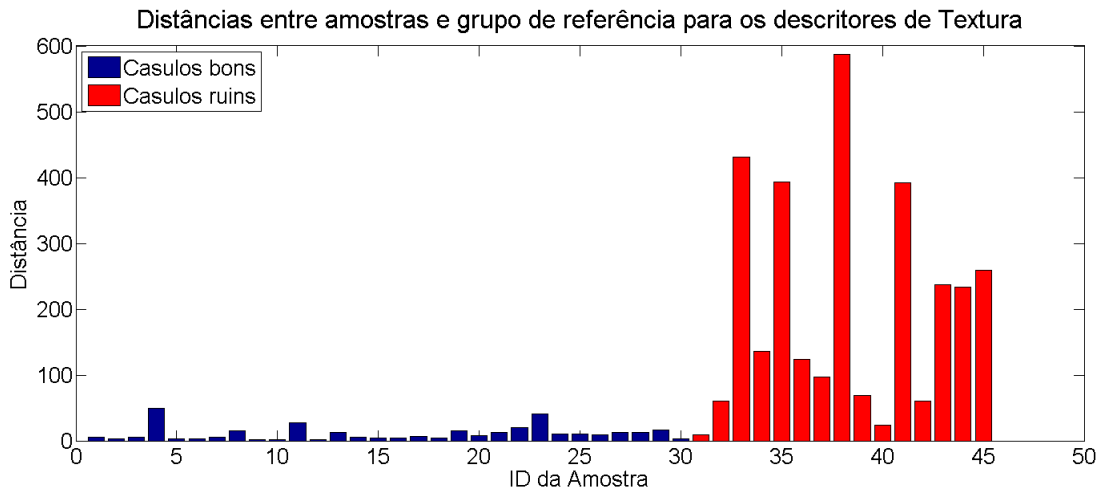
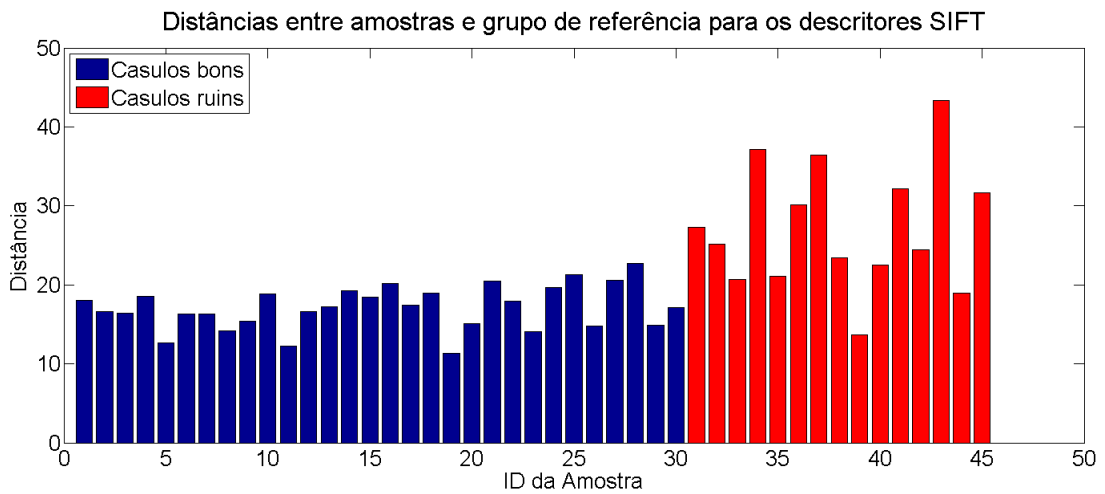


Figura 37 – Distância de Mahalanobis para os descritores de Textura - BD3



Fonte: Autoria própria

Figura 38 – Médias das distâncias Euclidianas mínimas para os descritores SIFT - BD3



Fonte: Autoria própria

A seguir definiu-se o limiar de distância para a seleção de casulos do banco de dados 3 em 35 para o descritor de forma, 51 para o descritor de textura e 23 para o descritor SIFT e verificou-se a capacidade de seleção de cada descritor como feito anteriormente para os bancos de dados 1 e 2.

Figura 39 – Classificação de casulos adequados (imagem 1) - BD3



Fonte: Autoria própria

Figura 40 – Classificação de casulos adequados (imagem 2) - BD3



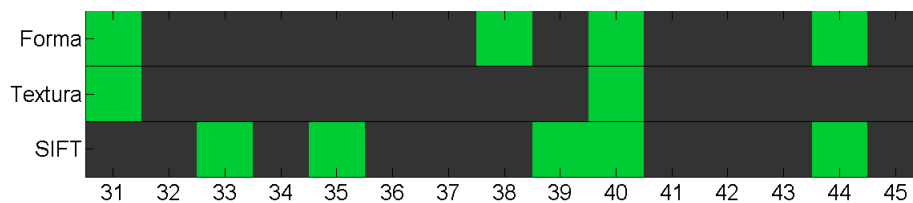
Fonte: Autoria própria

Figura 41 – Classificação de casulos defeituosos - BD3



Fonte: Autoria própria

Figura 42 – Quadro da classificação dos casulos defeituosos- BD3



Fonte: Autoria própria

Como esperado, o desempenho do algoritmo de classificação para o banco de dados 3 é similar ao banco de dados 2 pois os mesmos casulos foram fotografados para ambos os bancos de dados. Fotografias com um maior tempo de exposição ressaltam diferenças na textura dos casulos e, portanto, aumentam a eficiência dos descritores de textura e SIFT.

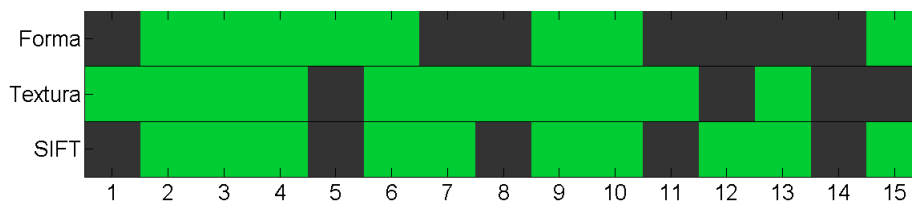
Assim como foi realizado para o banco de dados 2, utilizou-se uma imagem com casulos adequados e defeituosos para a identificação automática dos casulos que possuíssem baixa semelhança com o banco de dados de referência. As figuras 43 e 44 ilustram o resultado do algoritmo de classificação para o banco de dados 3. Os limiares escolhidos empiricamente foram: Forma = 14; textura = 300; e SIFT = 25 .

Figura 43 – Classificação automática dos casulos - BD3



Fonte: Autoria própria

Figura 44 – Quadro da classificação automática dos casulos - BD3



Fonte: Autoria própria

O aumento do tempo de exposição aumentou o contraste entre os casulos e o fundo da imagem, facilitando a segmentação dos casulos portanto a eficiência do descritor de forma. O aumento do tempo de exposição também ressaltou aspectos relacionados com a textura como superfície dos casulos e a presença de manchas, melhorando também o desempenho do descritor de textura. Os descritores de forma e textura são suficientes para classificar todos casulos da imagem de teste, sendo o descritor SIFT redundante com o descritor de forma para esta imagem.

De uma maneira geral, as imagens de teste do banco de dados 2 e 3 demonstram um resultado promissor para a classificação de casulos de bichos da seda por processamento de imagem, utilizando descritores de textura, forma e SIFT.

4 Conclusão

O presente trabalho apresentou o desenvolvimento de um algoritmo de classificação de imagens de casulos de bichos da seda para selecionar casulos adequados à produção da seda. Para avaliar descritores de forma, textura e SIFT na classificação dos casulos, uma cadeia de visão computacional foi desenvolvida em Matlab e três bancos de dados serviram como prova de conceito para o algoritmo de classificação.

No primeiro banco de dados, utilizando apenas duas imagens enviadas por uma empresa na área de fiação de seda, foi possível desenvolver um algoritmo piloto com as partes de segmentação e extração de descritores. Como o número de casulos disponíveis neste banco de dados é reduzido, não foi possível avaliar satisfatoriamente o cálculo de similaridade pela distância de Mahalanobis.

No segundo e terceiro banco de dados, grupos de casulos de referência de 60 casulos foram utilizados, permitindo uma melhor avaliação dos cálculos de similaridade. Quanto maior for o número de casulos utilizados para gerar o grupo de referência, maior será a robustez do cálculo de similaridade pois mais variações possíveis de forma e textura dentro do padrão considerado adequado serão levados em conta. Em outras palavras, a matriz covariância do grupo de referência se aproximará da matriz covariância do conjunto contendo o universo de todos os casulos possíveis que seriam considerados adequados.

As imagens de teste mostraram uma excelente capacidade de classificação do algoritmo, existindo inclusive valores de limiar que fornecem a separação completa entre casulos adequados e defeituosos. Esta é uma boa indicação que os descritores utilizados podem ser suficientes para a classificação dos casulos, porém um maior número de imagens de teste é necessário para avaliar a robustez e desempenho do algoritmo antes de aplica-lo à uma linha de produção.

Como trabalhos futuros, é interessante desenvolver um sistema de aquisição de imagens para a linha de produção dos casulos de bicho da seda. Este sistema deve fotografar casulos num ambiente controlado, ou seja, com iluminação e parâmetros da câmera de aquisição ajustados para fornecer uma classificação otimizada dos casulos.

É também necessário um estudo sobre a integração das etapas de aquisição, segmentação e classificação no sistema de visão computacional, como por exemplo estudando o impacto de métodos de segmentação alternativos ([BOMFIM, 2015](#)) na classificação dos casulos.

Finalmente o sistema de visão computacional deve passar por exaustivos testes para avaliar a robustez da classificação de casulos, mostrando a capacidade de seleção de casulos para grandes bancos de dados de imagens.

Referências

- BOMFIM, R. E. C. *Visão Computacional aplicada ao Controle de Qualidade de Casulos do Bicho-da-Seda: Subsistema de Segmentação de Imagens*. Monografia de Especialização, 2015. Citado nas páginas 9, 10, 27 e 43.
- BRATAC. *Vídeo Institucional*. 2002. Disponível em: <<http://www.youtube.com/watch?v=1qm10b2ahFw>>. Citado na página 9.
- FIEP, A. *Embrapii aprova projeto de indústria paranaense*. 2013. Disponível em: <<http://www.agenciafiep.com.br/noticia/embrapii-aprova-projeto-de-industria-paranaense/>>. Citado na página 9.
- LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, v. 2, n. 60, p. 91–110, 2004. Citado nas páginas 20, 21 e 22.
- NIST/SEMATECH. *NIST/SEMATECH e-Handbook of Statistical Methods*. 2015. Disponível em: <<http://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm>>. Citado na página 19.
- NIXON, M. S.; AGUADO, A. S. *Feature Extraction and Image Processing*. 1th. ed. [S.l.]: Newnes, 2002. ISBN 0750650788. Citado nas páginas 14, 15, 17, 18 e 29.
- SCHWARTZ, W. R.; SIQUEIRA, F. Roberti de; PEDRINI, H. Evaluation of feature descriptors for texture classification. *Journal of Electronic Imaging*, v. 21, n. 2, p. 023016–1–023016–17, 2012. Disponível em: <<http://dx.doi.org/10.1117/1.JEI.21.2.023016>>. Citado nas páginas 18 e 29.
- 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 9.
- VEDALDI, A.; FULKERSON, B. *VLFeat: An Open and Portable Library of Computer Vision Algorithms*. 2008. Disponível em: <<http://www.vlfeat.org/>>. Citado na página 29.

Anexos

```

1 % Algoritmo de Segmentação das imagens do Banco de Dados 1 (Seg_BD1.m)
2
3 %% Primeira parte: Segmentacao dos casulos bons
4
5 % Faz a leitura da imagem base para os casulos bons e salva na variável Ib
6 Ib=imread('Imagens/BD1/IMG_BD1_B01.jpg');
7
8 % Converte a imagem de RGB para tons de cinza
9 Gb=rgb2gray(Ib);
10
11 % Faz a detecção de borda utilizando o método Canny, threshold superior é
12 % 0.4 e o inferior é 0.4*0.4=0.16 e salva na variável Xb
13 Xb=edge(Gb,'canny',0.4);
14
15 % remove os grupos de pixels com área inferior a 25 pixels
16 Xb = bwareaopen(Xb,25);
17
18 % Define os parametros para a dilatação da imagem para fechar os segmentos
19 % do contorno. A forma de referencia é linear a 0 e 90 graus.
20 se90 = strel('line', 2, 90);
21 se0 = strel('line', 2, 0);
22
23 % É realizada a dilatação seguida pelo preenchimento dos espaços limitados
24 % pelo contorno fechado.
25 Xb = imdilate(Xb, [se90 se0]);
26 Xb = imfill(Xb,'holes');
27
28 % É realizada a transformação da distancia euclidiana da imagem, para
29 % selecionar o centro dos casulos e ponto de partida para o algoritmo
30 % watershed que servirá para separar os casulos que estão se tocando. As
31 % operações a seguir são realizadas para garantir o funcionamento do
32 % algoritmo watershed, por exemplo fixando um unico minimo local.
33 D = bwdist(~Xb);
34 D = -D;
35 D(~Xb) = -Inf;
36
37 % Realiza uma operação de erosão para selecionar os centros das áreas dos
38 % casulos
39 se = strel('disk',15,0);
40 Z=imerode(D,se);
41
42 %Impõe as regiões centrais das áreas dos casulos como mínimo local.
43 Z=imregionalmin(Z);
44 D=imimposemin(D,~Z);
45
46 % Calculo da função Watershed para definir a fronteira das regiões entre
47 % os casulos
48 L = watershed(D, 8);
49
50 % Salva a fronteira calculada pelo watershed na variável Bx
51 Bx=im2bw(label2rgb(L,'white',[0 0 0]));
52
53 % Ajusta os parametros da dilatação para lineares em 0 e 90 graus
54 se90 = strel('line', 1, 90);
55 se0 = strel('line', 1, 0);
56
57 % Realiza a dilatação das linhas representando a fronteira entre as imagens
58 % dos casulos
59 Bx = imdilate(~Bx, [se90 se0]);
60

```

```

1 % Algoritmo de Segmentação das imagens do Banco de Dados 2 (Seg_BD2.m)
2
3 %% Primeira parte: Segmentacao dos casulos bons
4
5 IMJPG=cell(1,6);
6
7 for x=1:6
8     arquivo=sprintf('Imagens/BD2/IMG_BD2_B%d.jpg',x);
9     IMJPG{1, x}=arquivo;
10 end
11
12 % Inicializa o tamanho dos grupos ()
13 TamGrupos=[];
14
15 % Salva o número de grupos
16 NumGrupos=length(IMJPG);
17
18 % Informa ao usuário o número de arquivos de imagem que serão processados
19 disp('---')
20 str=sprintf('%d arquivos de imagem de casulos bons serão segmentados...',length
(IMJPG));
21 disp(str)
22 disp(' ')
23
24 for i=1:length(IMJPG)
25
26     % Informa a imagem que está sendo processada
27     str=sprintf('Segmentando a imagem número %d...',i);
28     disp(str)
29
30     % Faz a leitura da imagem base para os casulos bons e salva na variável I
31     Ib=imread(char(IMJPG(i)));
32
33     % Utiliza o quadro vermelho para padronizar com a segmentacao dos
34     % casulos ruins, onde trabalhar com o quadro vermelho diminui a
35     % influencia dos defeitos
36     Gb=Ib(:,:,1);
37
38     % Faz a detecção de borda utilizando o método Canny, threshold superior é
39     % 0.3 e o inferior é 0.3*0.3=0.09 e salva na variável Xb
40     Xb=edge(Gb,'canny',0.3);
41
42     % Define os parametros para a dilatação da imagem para fechar os segmentos
43     % do contorno. A forma de referencia é linear a 0 e 90 graus.
44     se90 = strel('line', 3, 90);
45     se0 = strel('line', 3, 0);
46
47     % É realizada a dilatação seguida pelo preenchimento dos espaços limitados
48     % pelo contorno fechado.
49     Xb = imdilate(Xb, [se90 se0]);
50     Xb = imfill(Xb,'holes');
51
52     % remove os grupos de pixels com área inferior a 200 pixels
53     Xb = bwareaopen(Xb,200);
54
55     % reduz a area segmentada para a imagem
56     Xb = imerode(Xb, [se90 se0]);
57
58     % remove os grupos de pixels com área inferior a 25 pixels
59     Xb = bwareaopen(Xb,200);

```



```

1 % Algoritmo de Segmentação das imagens do Banco de Dados 3 (Seg_BD3.m)
2
3 %% Primeira parte: Segmentacao dos casulos bons
4
5 IMJPG=cell(1,6);
6
7 for x=1:6
8     arquivo=sprintf('Imagens/BD3/IMG_BD3_B%d.jpg',x);
9     IMJPG{1, x}=arquivo;
10 end
11
12 % Inicializa o tamanho dos grupos ()
13 TamGrupos=[];
14
15 % Salva o número de grupos
16 NumGrupos=length(IMJPG);
17
18 % Informa ao usuário o número de arquivos de imagem que serão processados
19 disp('---')
20 str=sprintf('%d arquivos de imagem de casulos bons serão segmentados...',length
(IMJPG));
21 disp(str)
22 disp(' ')
23
24 for i=1:length(IMJPG)
25
26     % Informa a imagem que está sendo processada
27     str=sprintf('Segmentando a imagem número %d...',i);
28     disp(str)
29
30     % Faz a leitura da imagem base para os casulos bons e salva na variável I
31     Ib=imread(char(IMJPG(i)));
32
33     % Utiliza o quadro vermelho para padronizar com a segmentacao dos
34     % casulos ruins, onde trabalhar com o quadro vermelho diminui a
35     % influencia dos defeitos
36     Gb=Ib(:,:,1);
37
38     % Faz a detecção de borda utilizando o método Canny, threshold superior é
39     % 0.3 e o inferior é 0.3*0.3=0.09 e salva na variável Xb
40     Xb=edge(Gb,'canny',0.3);
41
42     % Define os parametros para a dilatação da imagem para fechar os segmentos
43     % do contorno. A forma de referencia é linear a 0 e 90 graus.
44     se90 = strel('line', 3, 90);
45     se0 = strel('line', 3, 0);
46
47     % É realizada a dilatação seguida pelo preenchimento dos espaços limitados
48     % pelo contorno fechado.
49     Xb = imdilate(Xb, [se90 se0]);
50     Xb = imfill(Xb,'holes');
51
52     % remove os grupos de pixels com área inferior a 200 pixels
53     Xb = bwareaopen(Xb,200);
54
55     % reduz a area segmentada para a imagem
56     Xb = imerode(Xb, [se90 se0]);
57
58     % remove os grupos de pixels com área inferior a 25 pixels
59     Xb = bwareaopen(Xb,200);

```

```

1 % Algoritmo de Extracao dos descritores do Banco de Dados 1 (Ext_BD1.m)
2
3 %% Extracao dos descritores dos casulos bons
4
5 % Carrega o mapa de segmentacao dos casulos bons
6 load('Segmentacao/BD1/seg_casulos_bons.mat')
7 L=Lx;
8
9 % Faz a leitura da imagem base para os casulos bons e salva na variável Ib
10 I=imread('Imagens/BD1/IMG_BD1_B01.jpg');
11
12 % Seleciona o quadro referente a cor azul na imagem pois esta destaca
13 % melhor os defeitos
14 G=I(:,:,3);
15
16 % Determina a pasta onde os arquivos dos descritores devem ser salvos
17 pasta=sprintf('BD1');
18
19 % Determina que estes casulos sao do tipo bom
20 tipo=sprintf('B');
21
22 % Executa script de extracao dos descritores
23 run('Extracao_Descritores.m')
24
25 % Exibe mensagem de finalizacao do algoritmo
26 disp('Descritores dos casulos bons extraídos com sucesso')
27 disp('--')
28
29 %% Extracao dos descritores dos casulos ruins
30
31 % Carrega o mapa de segmentacao dos casulos ruins
32 load('Segmentacao/BD1/seg_casulos_ruins.mat')
33 L=Lz;
34
35 % Faz a leitura da imagem base para os casulos bons e salva na variável I
36 I=imread('Imagens/BD1/IMG_BD1_R01.jpg');
37
38 % Seleciona o quadro referente a cor azul na imagem pois esta destaca
39 % melhor os defeitos
40 G=I(:,:,3);
41
42 % Determina a pasta onde os arquivos dos descritores devem ser salvos
43 pasta=sprintf('BD1');
44
45 % Determina que estes casulos sao do tipo bom
46 tipo=sprintf('R');
47
48 % Executa script de extracao dos descritores
49 run('Extracao_Descritores.m')
50
51 % Exibe mensagem de finalizacao do algoritmo
52 disp('Descritores dos casulos ruins extraídos com sucesso')
53 disp('--')
54

```

```

1 % Algoritmo de Extracao dos descritores do Banco de Dados 2 (Ext_BD2.m)
2
3 %% Extracao dos descritores dos casulos bons
4
5 for k=1:6
6
7     % Carrega o mapa de segmentacao dos casulos bons
8     arquivo=sprintf('Segmentacao/BD2/seg_casulos_bonsB0%d.mat',k);
9     load(arquivo)
10    L=Lx;
11
12    % Faz a leitura da imagem base para os casulos bons e salva na variável I
13    arquivo=sprintf('Imagens/BD2/IMG_BD2_B0%d.jpg',k);
14    I=imread(arquivo);
15
16    % Seleciona o quadro referente a cor azul na imagem pois esta destaca
17    % melhor os defeitos
18    G=I(:,:,3);
19
20    % Determina a pasta onde os arquivos dos descritores devem ser salvos
21    pasta=sprintf('BD2');
22
23    % Determina que estes casulos sao do tipo bom
24    tipo=sprintf('G%dB',k);
25
26    % Executa script de extracao dos descritores
27    run('Extracao_Descritores.m')
28
29    % Exibe mensagem de finalizacao do algoritmo
30    disp('Descritores dos casulos bons extraidos com sucesso')
31    disp('--')
32
33 end
34
35 %% Extracao dos descritores dos casulos ruins
36
37 % Carrega o mapa de segmentacao dos casulos ruins
38 load('Segmentacao/BD2/seg_casulos_ruins.mat')
39 L=Lz;
40
41 % Faz a leitura da imagem base para os casulos ruins e salva na variável I
42 I=imread('Imagens/BD2/IMG_BD2_R01.jpg');
43
44 % Seleciona o quadro referente a cor azul na imagem pois esta destaca
45 % melhor os defeitos
46 G=I(:,:,3);
47
48 % Determina a pasta onde os arquivos dos descritores devem ser salvos
49 pasta=sprintf('BD2');
50
51 % Determina que estes casulos sao do tipo ruim
52 tipo=sprintf('R');
53
54 % Executa script de extracao dos descritores
55 run('Extracao_Descritores.m')
56
57 % Exibe mensagem de finalizacao do algoritmo
58 disp('Descritores dos casulos ruins extraidos com sucesso')
59 disp('--')
60

```

```

1 % Algoritmo de Extracao dos descritores do Banco de Dados 3 (Ext_BD3.m)
2
3 %% Extracao dos descritores dos casulos bons
4
5 for k=1:6
6
7     % Carrega o mapa de segmentacao dos casulos bons
8     arquivo=sprintf('Segmentacao/BD3/seg_casulos_bonsB0%d.mat',k);
9     load(arquivo)
10    L=Lx;
11
12    % Faz a leitura da imagem base para os casulos bons e salva na variável I
13    arquivo=sprintf('Imagens/BD3/IMG_BD3_B0%d.jpg',k);
14    I=imread(arquivo);
15
16    % Seleciona o quadro referente a cor azul na imagem pois esta destaca
17    % melhor os defeitos
18    G=I(:,:,3);
19
20    % Determina a pasta onde os arquivos dos descritores devem ser salvos
21    pasta=sprintf('BD3');
22
23    % Determina que estes casulos sao do tipo bom
24    tipo=sprintf('G%dB',k);
25
26    % Executa script de extracao dos descritores
27    run('Extracao_Descritores.m')
28
29    % Exibe mensagem de finalizacao do algoritmo
30    disp('Descritores dos casulos bons extraidos com sucesso')
31    disp('--')
32
33 end
34
35 %% Extracao dos descritores dos casulos ruins
36
37 % Carrega o mapa de segmentacao dos casulos ruins
38 load('Segmentacao/BD3/seg_casulos_ruins.mat')
39 L=Lz;
40
41 % Faz a leitura da imagem base para os casulos ruins e salva na variável I
42 I=imread('Imagens/BD3/IMG_BD3_R01.jpg');
43
44 % Seleciona o quadro referente a cor azul na imagem pois esta destaca
45 % melhor os defeitos
46 G=I(:,:,3);
47
48 % Determina a pasta onde os arquivos dos descritores devem ser salvos
49 pasta=sprintf('BD3');
50
51 % Determina que estes casulos sao do tipo ruim
52 tipo=sprintf('R');
53
54 % Executa script de extracao dos descritores
55 run('Extracao_Descritores.m')
56
57 % Exibe mensagem de finalizacao do algoritmo
58 disp('Descritores dos casulos ruins extraidos com sucesso')
59 disp('--')
60

```

```

1 % Algoritmos generico para extrair Descritores de forma, textura e SIFT (Extracao_Descritores.m)
2
3 %% Script de extracao de descritores a partir das variaveis:
4 % L: mapa de regioes segmentadas
5 % G: imagem original da segmentacao em tons de cinza
6 % pasta: string representando a pasta onde os arquivos devem ser salvos
7 % tipo: string representando o tipo de casulo (bom, ruim ou teste)
8
9 % Adquire as informações sobre cada área.
10 stats=bwconncomp(L);
11
12 % Extrai informações dos contornos das áreas detectadas
13 [B1,L1,N1,A1]=bwboundaries(L);
14
15 % Inicializa vetor dos descritores de forma
16 FormDesc=zeros(1,10);
17
18 % Loop para extrair descritores de cada casulo e salvar resultados
19 for j=1:stats.NumObjects
20
21     % Extrai a mascara da área selecionada
22     mask=uint8(zeros(size(L)));
23     mask(stats.PixelIdxList{j})=1;
24
25     % Extrai o vetor dos descritores elipticos de forma
26     Vec_Form=EllipticDescrp(B1{j}', 11);
27     FormDesc=Vec_Form(2:end);
28
29     % Inicializa a biblioteca SIFT
30     run('C:\Users\Victor\Documents\MATLAB\vlfeat-0.9.18-bin\vlfeat-0.9.18\toolbox\vl_setup')
31
32     %Gera imagem de um casulo individual
33     Im_seg=mask.*G;
34
35     % Calcula os descritores SIFT (edgethresh 1.5 para casulos individuais)
36     [fa, SIFTDesc] = vl_sift(single(Im_seg), 'edgethresh', 30);
37
38     % Calcula o histograma do casulo, ignorando os pixels nulos
39     [h_da,bins]= imhist(Im_seg);
40     h_da(1)=0;
41
42     % Calcula os descritores de textura a partir do histograma
43     Pi=h_da/sum(h_da);
44     Mean_Hist=sum(Pi.*bins);
45     Std_Hist=sqrt(sum(Pi.*(bins-Mean_Hist).^2));
46     Skewness_Hist=(sum(Pi.*(bins-Mean_Hist).^3))/(Std_Hist^3);
47     Kurtosis_Hist=(sum(Pi.*(bins-Mean_Hist).^4))/(Std_Hist^4);
48     Entropy = -sum(Pi.*log2(max(Pi,min(Pi,(Pi~0)))));
49     Energy = sum(Pi.*Pi);
50
51     TextDesc =[ Mean_Hist, ...
52                Std_Hist, ...
53                Skewness_Hist, ...
54                Kurtosis_Hist, ...
55                Entropy, ...
56                Energy];
57
58     % Salva os descritores dos casulos na pasta indicada

```

```

1  %(Código adaptado, extraído do livro Feature Extraction And Image Processing - Mark
S. Nixon, Alberto S. Aguado - 2002)
2  %Elliptic Fourier Descriptors (EllipticDescrp.m)
3  function CE=EllipticDescrp(curve,n)
4      %n=num coefficients
5      %if n=0 then n=m/2
6      %Scale amplitude output
7
8      %Function from image
9      X=curve(1,:);
10     Y=curve(2,:);
11     m=size(X,2);
12
13     %Elliptic Fourier Descriptors
14     if(n==0)
15         n=floor(m/2);
16     end; %number of coefficients
17
18     %Fourier Coefficients
19     ax=zeros(1,n); bx=zeros(1,n);
20     ay=zeros(1,n); by=zeros(1,n);
21     t=2*pi/m;
22
23     for k=1:n
24         for i=1:m
25             ax(k)=ax(k)+X(i)*cos(k*t*(i-1));
26             bx(k)=bx(k)+X(i)*sin(k*t*(i-1));
27             ay(k)=ay(k)+Y(i)*cos(k*t*(i-1));
28             by(k)=by(k)+Y(i)*sin(k*t*(i-1));
29         end
30         ax(k)=ax(k)*(2/m);
31         bx(k)=bx(k)*(2/m);
32         ay(k)=ay(k)*(2/m);
33         by(k)=by(k)*(2/m);
34     end
35
36     %Invariant
37     CE=zeros(1,n);
38     for k=1:n
39         CE(k)=sqrt((ax(k)^2+ay(k)^2)/(ax(1)^2+ay(1)^2))+sqrt((bx(k)^2+by(k)^2)/(bx
(1)^2+by(1)^2));
40     end
41
42 end

```

```

1 % Apresentação dos Descritores do BD1 (Comp_BD1.m)
2
3 %% Cálculos de similaridade
4
5 clc
6 clear all
7
8 % Número de casulos bons que serão utilizados como referencia do total de
9 % 19 casulos
10 N_Ref=10;
11
12 % Inicialização das variáveis dos descritores de referencia dos casulos
13 % bons
14 FormDescT=[];
15 TextDescT=[];
16 SIFTDescT=[];
17
18 % Leitura dos arquivos salvos dos descritores
19 for i=1:N_Ref
20     % Leitura dos descritores de forma
21     arquivo=sprintf('Descritores/BD1/Forma/FormDesc_CasuloB%d',i);
22     load(arquivo);
23     FormDescT=[FormDescT FormDesc'];
24
25     % Leitura dos descritores de textura
26     arquivo=sprintf('Descritores/BD1/Textura/TextDesc_CasuloB%d',i);
27     load(arquivo);
28     TextDescT=[TextDescT TextDesc'];
29
30     % Leitura dos descritores SIFT
31     arquivo=sprintf('Descritores/BD1/SIFT/SIFTDesc_CasuloB%d',i);
32     load(arquivo);
33     SIFTDescT=[SIFTDescT SIFTDesc'];
34 end
35
36 % Inicialização das variáveis das distancias para os casulos de teste bons
37 distFormB=[];
38 distTextB=[];
39 distSIFTB=[];
40
41 % Leitura dos descritores dos casulos de teste bons e cálculo da
42 % similaridade com o vetor de referencia
43 for i=(N_Ref+1):19
44     % Leitura dos descritores de forma
45     arquivo=sprintf('Descritores/BD1/Forma/FormDesc_CasuloB%d',i);
46     load(arquivo);
47
48     % Leitura dos descritores de textura
49     arquivo=sprintf('Descritores/BD1/Textura/TextDesc_CasuloB%d',i);
50     load(arquivo);
51
52     % Leitura dos descritores SIFT
53     arquivo=sprintf('Descritores/BD1/SIFT/SIFTDesc_CasuloB%d',i);
54     load(arquivo);
55
56     % Cálculo das distancias entre o grupo de referencia e as amostras
57     [distForm,distText,distSIFT]=DistanceCalc(FormDesc,FormDescT,TextDesc,
TextDescT,SIFTDesc,SIFTDescT);
58
59     % Armazenamento das variáveis de distancia

```

```

1 % Apresentação dos Descritores do BD2 (Comp_BD2.m)
2
3 %% Cálculos de similaridade
4
5 clc
6 clear all
7
8 % Número de casulos bons que serão utilizados como referencia do total de
9 % 90 casulos
10
11 % Inicialização das variáveis dos descritores de referencia dos casulos
12 % bons
13 FormDescT=[];
14 TextDescT=[];
15 SIFTDescT=[];
16
17 % Leitura dos arquivos salvos dos descritores
18 for k=1:4
19     for i=1:15
20         % Leitura dos descritores de forma
21         arquivo=sprintf('Descritores/BD2/Forma/FormDesc_CasuloGdB%d',k,i);
22         load(arquivo);
23         FormDescT=[FormDescT FormDesc'];
24
25         % Leitura dos descritores de textura
26         arquivo=sprintf('Descritores/BD2/Textura/TextDesc_CasuloGdB%d',k,i);
27         load(arquivo);
28         TextDescT=[TextDescT TextDesc'];
29
30         % Leitura dos descritores SIFT
31         arquivo=sprintf('Descritores/BD2/SIFT/SIFTDesc_CasuloGdB%d',k,i);
32         load(arquivo);
33         SIFTDescT=[SIFTDescT SIFTDesc'];
34     end
35 end
36
37 % Inicialização das variáveis das distancias para os casulos de teste bons
38 distFormB=[];
39 distTextB=[];
40 distSIFTB=[];
41
42 % Leitura dos descritores dos casulos de teste bons e cálculo da
43 % similaridade com o vetor de referencia
44 for k=5:6
45     for i=1:15
46         % Leitura dos descritores de forma
47         arquivo=sprintf('Descritores/BD2/Forma/FormDesc_CasuloGdB%d',k,i);
48         load(arquivo);
49
50         % Leitura dos descritores de textura
51         arquivo=sprintf('Descritores/BD2/Textura/TextDesc_CasuloGdB%d',k,i);
52         load(arquivo);
53
54         % Leitura dos descritores SIFT
55         arquivo=sprintf('Descritores/BD2/SIFT/SIFTDesc_CasuloGdB%d',k,i);
56         load(arquivo);
57
58         % Cálculo das distancias entre o grupo de referencia e as amostras
59         [distForm,distText,distSIFT]=DistanceCalc(FormDesc,FormDescT,TextDesc,
TextDescT,SIFTDesc,SIFTDescT);

```



```

1 % Apresentação dos Descritores do BD3 (Comp_BD3.m)
2
3 %% Cálculos de similaridade
4
5 clc
6 clear all
7
8 % Número de casulos bons que serão utilizados como referencia do total de
9 % 90 casulos
10 %N_Ref=10;
11
12 % Inicialização das variáveis dos descritores de referencia dos casulos
13 % bons
14 FormDescT=[];
15 TextDescT=[];
16 SIFTDescT=[];
17
18 % Leitura dos arquivos salvos dos descritores
19 for k=1:4
20     for i=1:15
21         % Leitura dos descritores de forma
22         arquivo=sprintf('Descritores/BD3/Forma/FormDesc_CasuloGdB%d',k,i);
23         load(arquivo);
24         FormDescT=[FormDescT FormDesc'];
25
26         % Leitura dos descritores de textura
27         arquivo=sprintf('Descritores/BD3/Textura/TextDesc_CasuloGdB%d',k,i);
28         load(arquivo);
29         TextDescT=[TextDescT TextDesc'];
30
31         % Leitura dos descritores SIFT
32         arquivo=sprintf('Descritores/BD3/SIFT/SIFTDesc_CasuloGdB%d',k,i);
33         load(arquivo);
34         SIFTDescT=[SIFTDescT SIFTDesc'];
35     end
36 end
37
38 % Inicialização das variáveis das distancias para os casulos de teste bons
39 distFormB=[];
40 distTextB=[];
41 distSIFTB=[];
42
43 % Leitura dos descritores dos casulos de teste bons e cálculo da
44 % similaridade com o vetor de referencia
45 for k=5:6
46     for i=1:15
47         % Leitura dos descritores de forma
48         arquivo=sprintf('Descritores/BD3/Forma/FormDesc_CasuloGdB%d',k,i);
49         load(arquivo);
50
51         % Leitura dos descritores de textura
52         arquivo=sprintf('Descritores/BD3/Textura/TextDesc_CasuloGdB%d',k,i);
53         load(arquivo);
54
55         % Leitura dos descritores SIFT
56         arquivo=sprintf('Descritores/BD3/SIFT/SIFTDesc_CasuloGdB%d',k,i);
57         load(arquivo);
58
59         % Cálculo das distancias entre o grupo de referencia e as amostras
60         [distForm,distText,distSIFT]=DistanceCalc(FormDesc,FormDescT,TextDesc,

```

```

1 function [distForm,distText,distSIFT]=DistanceCalc(FormDesc,FormDescT,TextDesc,↵
TextDescT,SIFTDesc,SIFTDescT)
2 % Esta função calcula a distância entre a amostra e o vetor de casulos
3 % bons(DistanceCalc.m)
4
5 % Cálculo da distância de Mahalanobis entre os descritores de Forma e textura
6 distForm=mahal(FormDesc(2:end),(FormDescT(2:end,:))');
7 distText=mahal(TextDesc,TextDescT');
8
9 % Cálculo da distancia euclidiana mínima dos descritores SIFT
10 dmin=(sum((SIFTDesc(:,1)-SIFTDescT(:,1)).^2));
11 d_tot=0;
12
13 for k=1:size(SIFTDesc,2)
14     for p=1:size(SIFTDescT,2)
15         d=(sum((SIFTDesc(:,k)-SIFTDescT(:,p)).^2));
16
17         if d<dmin
18             dmin=d;
19         end
20     end
21     d_tot=d_tot+(dmin.^0.5)/size(SIFTDesc,2);
22 end
23 distSIFT=d_tot;
24 end

```

```

1 % Algoritmo para classificar casulos adequados do BD1 (Classif_AutomBD1B.m)
2
3 %%
4 % Carrega o resultado das comparações entre casulos de avaliação do BD1
5 run('Comp_BD1');
6
7 % Carrega a segmentação dos casulos de teste
8 load('Segmentacao/BD1/seg_casulos_bons.mat');
9 % Adquire as informações sobre cada área.
10 stats=bwconncomp(Lx);
11
12 % Carrega a imagem dos casulos de teste
13 figure,imshow('Imagens/BD1/IMG_BD1_B01.JPG');
14 hold
15
16 % % Define os limiares para os descritores com base nos gráficos de distancia
17 % % encontrados
18 Limiar_Forma=100000;
19 Limiar_Text=1500;
20 Limiar_SIFT=35;
21
22
23 % Seleciona os casulos considerados ruins e desenha caixa verde em torno
24 % deles
25 for y=1+10:stats.NumObjects
26
27     if((distFormB(y-10)>Limiar_Forma)|| (distTextB(y-10)>Limiar_Text)|| (distSIFTB(y-
28     % Obtem a mascara do casulo selecionado
29     mask=uint8(zeros(size(Lx)));
30     mask(stats.PixelIdxList{y})=1;
31
32     % Desenha uma caixa em torno da imagem
33     st = regionprops(mask, 'BoundingBox' );
34     rectangle('Position',[st.BoundingBox(1),st.BoundingBox(2),...
35     st.BoundingBox(3),st.BoundingBox(4)],'EdgeColor',...
36     'r','LineWidth',1 );
37
38     text(st.BoundingBox(1)+2, (st.BoundingBox(2)+10),num2str(y-10), 'color',
39 'red');
40
41     else
42     mask=uint8(zeros(size(Lx)));
43     mask(stats.PixelIdxList{y})=1;
44     % Desenha uma caixa em torno da imagem
45     st = regionprops(mask, 'BoundingBox' );
46     rectangle('Position',[st.BoundingBox(1),st.BoundingBox(2),...
47     st.BoundingBox(3),st.BoundingBox(4)],'EdgeColor',...
48     'g','LineWidth',1 );
49
50     text(st.BoundingBox(1)+2, (st.BoundingBox(2)+10),num2str(y-10), 'color',
51 'green');
52
53     end
54 end
55
56 % Avaliação dos descritores utilizados para a classificação de cada casulo
57

```

```

1 % Algoritmo para classificar casulos defeituosos do BD1 (Classif_AutomBD1R.m)
2
3 %%
4 % Carrega o resultado das comparações entre casulos de avaliação do BD1
5 run('Comp_BD1');
6
7 % Carrega a segmentação dos casulos de teste
8 load('Segmentacao/BD1/seg_casulos_ruins.mat');
9 % Adquire as informações sobre cada área.
10 stats=bwconncomp(Lz);
11
12 % Carrega a imagem dos casulos de teste
13 figure,imshow('Imagens/BD1/IMG_BD1_R01.JPG');
14 hold
15
16 % % Define os limiares para os descritores com base nos gráficos de distancia
17 % % encontrados
18 Limiar_Forma=100000;
19 Limiar_Text=1500;
20 Limiar_SIFT=35;
21
22
23 % Seleciona os casulos considerados ruins e desenha caixa verde em torno
24 % deles
25 for y=1:stats.NumObjects
26
27     if((distFormR(y)>Limiar_Forma)|| (distTextR(y)>Limiar_Text)|| (distSIFTR(y)
28 >Limiar_SIFT))
29         % Obtem a mascara do casulo selecionado
30         mask=uint8(zeros(size(Lz)));
31         mask(stats.PixelIdxList{y})=1;
32
33         % Desenha uma caixa em torno da imagem
34         st = regionprops(mask, 'BoundingBox' );
35         rectangle('Position',[st.BoundingBox(1),st.BoundingBox(2),...
36             st.BoundingBox(3),st.BoundingBox(4)], 'EdgeColor',...
37             'r','LineWidth',1 );
38
39         text(st.BoundingBox(1)+2, (st.BoundingBox(2)+10),num2str(y+9), 'color',
40 'red');
41
42     else
43         mask=uint8(zeros(size(Lz)));
44         mask(stats.PixelIdxList{y})=1;
45         % Desenha uma caixa em torno da imagem
46         st = regionprops(mask, 'BoundingBox' );
47         rectangle('Position',[st.BoundingBox(1),st.BoundingBox(2),...
48             st.BoundingBox(3),st.BoundingBox(4)], 'EdgeColor',...
49             'g','LineWidth',1 );
50
51         text(st.BoundingBox(1)+2, (st.BoundingBox(2)+10),num2str(y+9), 'color',
52 'green');
53
54     end
55
56 end
57
58 % Avaliação dos descritores utilizados para a classificação de cada casulo

```

```

1 function [distForm,distText,distSIFT]=DistanceCalc(FormDesc,FormDescT,TextDesc,↵
TextDescT,SIFTDesc,SIFTDescT)
2 % Esta função calcula a distância entre a amostra e o vetor de casulos
3 % bons(DistanceCalc.m)
4
5 % Cálculo da distância de Mahalanobis entre os descritores de Forma e textura
6 distForm=mahal(FormDesc(2:end),(FormDescT(2:end,:))');
7 distText=mahal(TextDesc,TextDescT');
8
9 % Cálculo da distancia euclidiana mínima dos descritores SIFT
10 dmin=(sum((SIFTDesc(:,1)-SIFTDescT(:,1)).^2));
11 d_tot=0;
12
13 for k=1:size(SIFTDesc,2)
14     for p=1:size(SIFTDescT,2)
15         d=(sum((SIFTDesc(:,k)-SIFTDescT(:,p)).^2));
16
17         if d<dmin
18             dmin=d;
19         end
20     end
21     d_tot=d_tot+(dmin.^0.5)/size(SIFTDesc,2);
22 end
23 distSIFT=d_tot;
24 end

```

```

1 % Algoritmo para classificar casulos adequados do BD2 (Classif_AutomBD2B.m)
2
3 %%
4 % Carrega o resultado das comparações entre casulos de avaliação do BD2
5 run('Comp_BD2');
6
7 % Carrega a segmentação dos casulos de teste
8 %load('Segmentacao/BD2/seg_casulos_bonsB05.mat');
9 load('Segmentacao/BD2/seg_casulos_bonsB06.mat');
10
11 % Adquire as informações sobre cada área.
12 stats=bwconncomp(Lx);
13
14 % Carrega a imagem dos casulos de teste
15 %figure,imshow('Imagens/BD2/IMG_BD2_B05.JPG');
16 figure,imshow('Imagens/BD2/IMG_BD2_B06.JPG');
17 hold
18
19 % % Define os limiares para os descritores com base nos gráficos de distancia
20 % % encontrados
21 Limiar_Forma=26;
22 Limiar_Text=91;
23 Limiar_SIFT=30;
24
25
26 % Seleciona os casulos considerados ruins e desenha caixa verde em torno
27 % deles
28 for y=1:stats.NumObjects
29
30     if((distFormB(y)>Limiar_Forma)|| (distTextB(y)>Limiar_Text)|| (distSIFTB(y) <
>Limiar_SIFT))
31         % Obtem a mascara do casulo selecionado
32         mask=uint8(zeros(size(Lx)));
33         mask(stats.PixelIdxList{y})=1;
34
35         % Desenha uma caixa em torno da imagem
36         st = regionprops(mask, 'BoundingBox' );
37         rectangle('Position',[st.BoundingBox(1),st.BoundingBox(2),...
38             st.BoundingBox(3),st.BoundingBox(4)], 'EdgeColor',...
39             'r', 'LineWidth',1 );
40
41         text(st.BoundingBox(1)+2, (st.BoundingBox(2)+10), num2str(y+15), 'color', <
'red');
42
43     else
44         mask=uint8(zeros(size(Lx)));
45         mask(stats.PixelIdxList{y})=1;
46         % Desenha uma caixa em torno da imagem
47         st = regionprops(mask, 'BoundingBox' );
48         rectangle('Position',[st.BoundingBox(1),st.BoundingBox(2),...
49             st.BoundingBox(3),st.BoundingBox(4)], 'EdgeColor',...
50             'g', 'LineWidth',1 );
51
52         text(st.BoundingBox(1)+2, (st.BoundingBox(2)+10), num2str(y+15), 'color', <
'green');
53
54     end
55
56
57 end

```

```

1 % Algoritmo para classificar casulos defeituosos do BD2 (Classif_AutomBD2R.m)
2
3 %%
4 % Carrega o resultado das comparações entre casulos de avaliação do BD2
5 run('Comp_BD2');
6
7 % Carrega a segmentação dos casulos de teste
8 load('Segmentacao/BD2/seg_casulos_ruins.mat');
9
10 % Adquire as informações sobre cada área.
11 stats=bwconncomp(Lz);
12
13 % Carrega a imagem dos casulos de teste
14 figure,imshow('Imagens/BD2/IMG_BD2_R01.JPG');
15 hold
16
17 % % Define os limiares para os descritores com base nos gráficos de distancia
18 % % encontrados
19 Limiar_Forma=26;
20 Limiar_Text=91;
21 Limiar_SIFT=30;
22
23
24 % Seleciona os casulos considerados ruins e desenha caixa verde em torno
25 % deles
26 for y=1:stats.NumObjects
27
28     if((distFormR(y)>Limiar_Forma)|| (distTextR(y)>Limiar_Text)|| (distSIFTR(y)
29 >Limiar_SIFT))
30         % Obtem a mascara do casulo selecionado
31         mask=uint8(zeros(size(Lz)));
32         mask(stats.PixelIdxList{y})=1;
33
34         % Desenha uma caixa em torno da imagem
35         st = regionprops(mask, 'BoundingBox' );
36         rectangle('Position',[st.BoundingBox(1),st.BoundingBox(2),...
37             st.BoundingBox(3),st.BoundingBox(4)],'EdgeColor',...
38             'r','LineWidth',1 );
39
40         text(st.BoundingBox(1)+2, (st.BoundingBox(2)+10),num2str(y+30), 'color',
41 'red');
42
43     else
44         mask=uint8(zeros(size(Lz)));
45         mask(stats.PixelIdxList{y})=1;
46         % Desenha uma caixa em torno da imagem
47         st = regionprops(mask, 'BoundingBox' );
48         rectangle('Position',[st.BoundingBox(1),st.BoundingBox(2),...
49             st.BoundingBox(3),st.BoundingBox(4)],'EdgeColor',...
50             'g','LineWidth',1 );
51
52         text(st.BoundingBox(1)+2, (st.BoundingBox(2)+10),num2str(y+30), 'color',
53 'green');
54
55     end
56
57 end
58
59 % Avaliação dos descritores utilizados para a classificação de cada casulo

```

```

1 %% Classificação Automática de Casulos BD2 (Classif_AutomBD2.m)
2
3 % Limpa terminal
4 clc
5 clear all
6
7 % Inicialização das variáveis dos descritores de referencia dos casulos
8 % bons
9 FormDescT=[];
10 TextDescT=[];
11 SIFTDescT=[];
12
13 % Leitura dos arquivos salvos dos descritores
14 for k=1:4
15     for i=1:15
16         % Leitura dos descritores de forma
17         arquivo=sprintf('Descritores/BD2/Forma/FormDesc_CasuloG%dB%d',k,i);
18         load(arquivo);
19         FormDescT=[FormDescT FormDesc'];
20
21         % Leitura dos descritores de textura
22         arquivo=sprintf('Descritores/BD2/Textura/TextDesc_CasuloG%dB%d',k,i);
23         load(arquivo);
24         TextDescT=[TextDescT TextDesc'];
25
26         % Leitura dos descritores SIFT
27         arquivo=sprintf('Descritores/BD2/SIFT/SIFTDesc_CasuloG%dB%d',k,i);
28         load(arquivo);
29         SIFTDescT=[SIFTDescT SIFTDesc'];
30     end
31 end
32
33 % Inicialização das variáveis das distancias para os casulos de teste ruins
34 distText_T=[];
35 distSIFT_T=[];
36 distForm_T=[];
37
38 % Leitura dos descritores dos casulos de teste ruins e cálculo da
39 % similaridade com o vetor de referencia
40 for i=1:15
41     %Leitura dos descritores de forma
42     arquivo=sprintf('Descritores/BD2/Forma/FormDesc_CasuloT%d',i);
43     load(arquivo);
44
45     % Leitura dos descritores de textura
46     arquivo=sprintf('Descritores/BD2/Textura/TextDesc_CasuloT%d',i);
47     load(arquivo);
48
49     % Leitura dos descritores SIFT
50     arquivo=sprintf('Descritores/BD2/SIFT/SIFTDesc_CasuloT%d',i);
51     load(arquivo);
52
53     % Cálculo das distancias entre o grupo de referencia e as amostras
54     [distForm,distText,distSIFT]=DistanceCalc(FormDesc,FormDescT,TextDesc,
TextDescT,SIFTDesc,SIFTDescT);
55
56     % Armazenamento das variáveis de distancia
57     distForm_T=[distForm_T distForm'];
58     distText_T=[distText_T distText'];
59     distSIFT_T=[distSIFT_T distSIFT'];

```



```

1 % Algoritmo para classificar casulos adequados do BD3 (Classif_AutomBD3B.m)
2
3 %%
4 % Carrega o resultado das comparações entre casulos de avaliação do BD3
5 run('Comp_BD3');
6
7 % Carrega a segmentação dos casulos de teste
8 %load('Segmentacao/BD3/seg_casulos_bonsB05.mat');
9 load('Segmentacao/BD3/seg_casulos_bonsB06.mat');
10
11 % Adquire as informações sobre cada área.
12 stats=bwconncomp(Lx);
13
14 % Carrega a imagem dos casulos de teste
15 %figure,imshow('Imagens/BD3/IMG_BD3_B05.JPG');
16 figure,imshow('Imagens/BD3/IMG_BD3_B06.JPG');
17 hold
18
19 % % Define os limiares para os descritores com base nos gráficos de distancia
20 % % encontrados
21 Limiar_Forma=35;
22 Limiar_Text=51;
23 Limiar_SIFT=23;
24
25
26 % Seleciona os casulos considerados ruins e desenha caixa verde em torno
27 % deles
28 for y=1:stats.NumObjects
29
30     if((distFormB(y)>Limiar_Forma)|| (distTextB(y)>Limiar_Text)|| (distSIFTB(y) <
>Limiar_SIFT))
31         % Obtem a mascara do casulo selecionado
32         mask=uint8(zeros(size(Lx)));
33         mask(stats.PixelIdxList{y})=1;
34
35         % Desenha uma caixa em torno da imagem
36         st = regionprops(mask, 'BoundingBox' );
37         rectangle('Position',[st.BoundingBox(1),st.BoundingBox(2),...
38             st.BoundingBox(3),st.BoundingBox(4)], 'EdgeColor',...
39             'r', 'LineWidth',1 );
40
41         text(st.BoundingBox(1)+2, (st.BoundingBox(2)+10), num2str(y+15), 'color',
'green');
42
43     else
44         mask=uint8(zeros(size(Lx)));
45         mask(stats.PixelIdxList{y})=1;
46         % Desenha uma caixa em torno da imagem
47         st = regionprops(mask, 'BoundingBox' );
48         rectangle('Position',[st.BoundingBox(1),st.BoundingBox(2),...
49             st.BoundingBox(3),st.BoundingBox(4)], 'EdgeColor',...
50             'g', 'LineWidth',1 );
51
52         text(st.BoundingBox(1)+2, (st.BoundingBox(2)+10), num2str(y+15), 'color',
'green');
53
54     end
55
56
57 end

```

```

1 % Algoritmo para classificar casulos defeituosos do BD3 (Classif_AutomBD3R.m)
2
3 %%
4 % Carrega o resultado das comparações entre casulos de avaliação do BD3
5 run('Comp_BD3');
6
7 % Carrega a segmentação dos casulos de teste
8 load('Segmentacao/BD3/seg_casulos_ruins.mat');
9
10 % Adquire as informações sobre cada área.
11 stats=bwconncomp(Lz);
12
13 % Carrega a imagem dos casulos de teste
14 figure,imshow('Imagens/BD3/IMG_BD3_R01.JPG');
15 hold
16
17 % % Define os limiares para os descritores com base nos gráficos de distancia
18 % % encontrados
19 Limiar_Forma=35;
20 Limiar_Text=51;
21 Limiar_SIFT=23;
22
23
24 % Seleciona os casulos considerados ruins e desenha caixa verde em torno
25 % deles
26 for y=1:stats.NumObjects
27
28     if((distFormR(y)>Limiar_Forma)|| (distTextR(y)>Limiar_Text)|| (distSIFTR(y)
29 >Limiar_SIFT))
30         % Obtem a mascara do casulo selecionado
31         mask=uint8(zeros(size(Lz)));
32         mask(stats.PixelIdxList{y})=1;
33
34         % Desenha uma caixa em torno da imagem
35         st = regionprops(mask, 'BoundingBox' );
36         rectangle('Position',[st.BoundingBox(1),st.BoundingBox(2),...
37             st.BoundingBox(3),st.BoundingBox(4)], 'EdgeColor',...
38             'r', 'LineWidth',1 );
39
40         text(st.BoundingBox(1)+2, (st.BoundingBox(2)+10), num2str(y+30), 'color',
41 'red');
42
43     else
44         mask=uint8(zeros(size(Lz)));
45         mask(stats.PixelIdxList{y})=1;
46         % Desenha uma caixa em torno da imagem
47         st = regionprops(mask, 'BoundingBox' );
48         rectangle('Position',[st.BoundingBox(1),st.BoundingBox(2),...
49             st.BoundingBox(3),st.BoundingBox(4)], 'EdgeColor',...
50             'g', 'LineWidth',1 );
51
52         text(st.BoundingBox(1)+2, (st.BoundingBox(2)+10), num2str(y+30), 'color',
53 'green');
54
55     end
56
57 end
58
59 % Avaliação dos descritores utilizados para a classificação de cada casulo

```

```

1 %% Classificação Automática de Casulos BD3 (Classif_AutomBD3.m)
2
3 % Limpa terminal
4 clc
5 clear all
6
7 % Inicialização das variáveis dos descritores de referencia dos casulos
8 % bons
9 FormDescT=[];
10 TextDescT=[];
11 SIFTDescT=[];
12
13 % Leitura dos arquivos salvos dos descritores
14 for k=1:4
15     for i=1:15
16         % Leitura dos descritores de forma
17         arquivo=sprintf('Descritores/BD3/Forma/FormDesc_CasuloGdB%d',k,i);
18         load(arquivo);
19         FormDescT=[FormDescT FormDesc'];
20
21         % Leitura dos descritores de textura
22         arquivo=sprintf('Descritores/BD3/Textura/TextDesc_CasuloGdB%d',k,i);
23         load(arquivo);
24         TextDescT=[TextDescT TextDesc'];
25
26         % Leitura dos descritores SIFT
27         arquivo=sprintf('Descritores/BD3/SIFT/SIFTDesc_CasuloGdB%d',k,i);
28         load(arquivo);
29         SIFTDescT=[SIFTDescT SIFTDesc'];
30     end
31 end
32
33 % Inicialização das variáveis das distancias para os casulos de teste ruins
34 distText_T=[];
35 distSIFT_T=[];
36 distForm_T=[];
37
38 % Leitura dos descritores dos casulos de teste ruins e cálculo da
39 % similaridade com o vetor de referencia
40 for i=1:15
41     %Leitura dos descritores de forma
42     arquivo=sprintf('Descritores/BD3/Forma/FormDesc_CasuloT%d',i);
43     load(arquivo);
44
45     % Leitura dos descritores de textura
46     arquivo=sprintf('Descritores/BD3/Textura/TextDesc_CasuloT%d',i);
47     load(arquivo);
48
49     % Leitura dos descritores SIFT
50     arquivo=sprintf('Descritores/BD3/SIFT/SIFTDesc_CasuloT%d',i);
51     load(arquivo);
52
53     % Cálculo das distancias entre o grupo de referencia e as amostras
54     [distForm,distText,distSIFT]=DistanceCalc(FormDesc,FormDescT,TextDesc,
TextDescT,SIFTDesc,SIFTDescT);
55
56     % Armazenamento das variáveis de distancia
57     distForm_T=[distForm_T distForm'];
58     distText_T=[distText_T distText'];
59     distSIFT_T=[distSIFT_T distSIFT'];

```