

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROJETO DE GRADUAÇÃO**



JÚLIA D'AVILA

**USO DE PROCESSAMENTO DIGITAL DE IMAGENS PARA
AVALIAÇÃO DE PERFIL QUÍMICO POR CROMATOGRAFIA
EM CAMADA DELGADA**

**VITÓRIA – ES
MAIO/2019**

JÚLIA D'AVILA

**USO DE PROCESSAMENTO DIGITAL DE IMAGENS PARA
AVALIAÇÃO DE PERFIL QUÍMICO POR CROMATOGRAFIA EM
CAMADA DELGADA**

Parte manuscrita do Projeto de Graduação da aluna **Júlia D'Avila**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheira Eletricista.

Orientador: Prof. Dr. Klaus Fabian Coco
Coorientador: Prof. Dr. Hildegardo Seibert França

VITÓRIA – ES
MAIO/2019

JÚLIA D'AVILA

**USO DE PROCESSAMENTO DIGITAL DE IMAGENS PARA
AVALIAÇÃO DE PERFIL QUÍMICO POR CROMATOGRAFIA EM
CAMADA DELGADA**

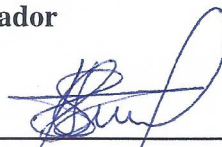
Parte manuscrita do Projeto de Graduação da aluna **Júlia D'Avila**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheira Eletricista.

Aprovado em 10 de maio de 2019.

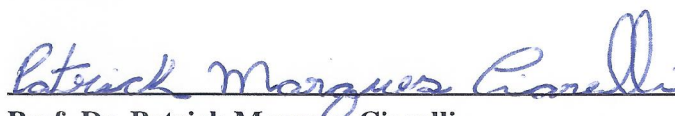
COMISSÃO EXAMINADORA:



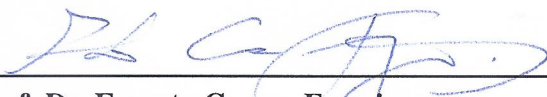
Prof. Dr. Klaus Fabian Côco
Universidade Federal do Espírito Santo
Orientador



Prof. Dr. Hildegardo Seibert França
Instituto Federal do Espírito Santo
Coorientador



Prof. Dr. Patrick Marques Ciarelli
Universidade Federal do Espírito Santo
Examinador



Prof. Dr. Ernesto Correa Ferreira
Instituto Federal do Espírito Santo
Examinador

*“Por vezes sentimos que aquilo que fazemos
não é senão uma gota de água no mar. Mas
o mar seria menor se lhe faltasse uma gota.”*

(Madre Teresa de Calcutá)

RESUMO

Como resultado do avanço tecnológico, novas ferramentas são criadas constantemente para facilitar e simplificar tarefas das mais comuns até as mais complexas. A cromatografia é um processo físico-químico, que embora tenha sido desenvolvida nos anos 1900, continua sendo utilizada nos dias atuais. Diversas técnicas foram desenvolvidas e, com o advento da tecnologia, esse processo tornou-se um dos melhores para caracterização de substâncias. Dentre as técnicas mais avançadas estão a Cromatografia Líquida de Alta Eficiência (CLAE), cromatografia de fase gasosa e nas mais simples está a Cromatografia em Camada Delgada (CCD), sendo essa atualmente realizada com procedimento mais manual. Para identificação das substâncias, a cromatografia sempre está atrelada a um método de caracterização e/ou identificação química, como espectrofotometria ultravermelha e infravermelha, espectrometria de massa e Ressonância Magnética Nuclear (RMN), porém, a técnica mais utilizada para caracterização é a CCD, por ser mais barata e rápida (DEGANI; CASS; VIEIRA, 1998). Uma das variáveis nessa técnica é justamente a obtenção do cromatograma, que é feita por revelação com um reagente químico e/ou físico, dando colorações e fatores de retardamento (R_f) característicos da substância em análise. Por esse motivo é desenvolvido neste trabalho uma eurística implementada em MATLAB® para identificar os padrões dos perfis cromatográficos da CCD obtidos a partir de plantas e, com esses dados, mapear a cor e disposição do perfil cromatográfico nas placas. A análise é feita por comparações para identificar se plantas diferentes apresentam perfis cromatográficos semelhantes ou não. Este estudo mostrou resultados comparativos satisfatórios, apresentando-se como uma alternativa para a análise da presença de substâncias em plantas, já que consegue reduzir significativamente o tempo necessário para a análise comparativa, além de auxílio às decisões com mais precisão, baseando-se nos valores exatos dos centros das manchas do cromatograma comparativo.

Palavras-chave: Cromatografia em camada delgada (CCD). Perfil cromatográfico. Processamento de imagem.

LISTA DE FIGURAS

Figura 1 – Exemplo de perfil cromatográfico de uma amostra de espinafre e as substâncias presentes na amostra	15
Figura 2 – Linha de aplicação	15
Figura 3 – O valor do fator de retardamento: a relação de ds e dm	16
Figura 4 – Distribuição das quantidades (figura inferior) de analitos em uma mancha (figura superior)	17
Figura 5 – Placa CDD com iluminação UV	19
Figura 6 – Perfil cromatográfico e seu perfil de intensidade	19
Figura 7 – Placa CCD e seu perfil de intensidade	20
Figura 8 – Etapas de um sistema de visão artificial	21
Figura 9 – Exemplo de imagem I com seis linhas e seis colunas, onde o <i>pixel</i> $I(3 \times 3)$ está em destaque, assim como sua vizinhança 3×3	23
Figura 10 – Seleção do perfil cromatográfico	26
Figura 11 – Perfil de intensidade - gerado pelo <i>ImageJ</i>	26
Figura 12 – Imagens descartadas	30
Figura 13 – Exemplo de imagem da placa de cromatografia a ser segmentada. A: Mancha do revelador; B: Corrida cromatográfica de uma amostra	31
Figura 14 – Aplicação do detector de bordas Canny. (a) Imagem binária da placa com as bordas detectadas através do método de Canny. (b) Perfil de intensidade de (a) na horizontal. (c) Perfil de intensidade de (a) na vertical	32
Figura 15 – Aplicação do detector de bordas de Sobel. (a) Imagem binária da placa com as bordas detectadas através do método de Sobel. (b) Perfil de intensidade de (a) na horizontal	32
Figura 16 – placa recortada e seu novo perfil de intensidade. (a) Região de interesse encontrada após detecção das bordas. (b) Perfil de intensidade de (a)	33
Figura 17 – As seis faixas obtidas a partir do perfil de intensidade da Figura 16	34
Figura 18 – Perfil de intensidade sem correção da linha de base	35
Figura 19 – Perfil de intensidade com correção da linha de base	35
Figura 20 – Faixa colorida seu perfil de intensidade com as manchas detectadas. (a) Faixa colorida e, em vermelho, os centros das manchas identificadas. (b) Perfil de intensidade de (a) e marcação de seus picos	36
Figura 21 – Uma mancha selecionada e seus perfis de intensidade horizontal e vertical	37
Figura 22 – As placas selecionadas e a comparação de suas substâncias (ou manchas). Limiares de R_f e de cor: 0,027 e 9	39
Figura 23 – Comparação 1	41
Figura 24 – Comparação 2	41
Figura 25 – Comparação 3	42

Figura 26 – Comparação 4	43
Figura 27 – Comparação 5	43

LISTA DE TABELAS

Tabela 1 – Alguns resultados obtidos a partir do código desenvolvido	40
--	----

LISTA DE ABREVIATURAS E SIGLAS

CCD	Cromatografia em Camada Delgada
CCF	Cromatografia em Camada Fina
CLAE	Cromatografia Líquida de Alta Eficiência
CMOS	<i>Complementary Metal-Oxide Semiconductor</i> – Semicondutor de Metal-Óxido Complementar
GMM	<i>Gaussian Mixture Model</i> – Modelo de Misturas Gaussianas
IFES	Instituto Federal do Espírito Santo
IV	Infravermelho
RGB	<i>Red, Green and Blue</i> – Vermelho, Verde e Azul
RMN	Ressonância Magnética Nuclear
SVA	Sistema de Visão Artificial
UFES	Universidade Federal do Espírito Santo
UV	Ultravioleta

LISTA DE SÍMBOLOS

R_f	Fator de retardamento / Fator de retenção
ds	Distância do ponto de aplicação da amostra até o centro da substância
dm	Distância do ponto de aplicação da amostra até a frente do solvente
R	Intensidade de Vermelho
G	Intensidade de Verde
B	Intensidade de Azul
C	Intensidade em nível de Cinza
I	Imagem no domínio espacial
(x,y)	Coordenadas de um ponto da imagem no domínio espacial
I'	Imagem filtrada no domínio espacial
$I'(x,y)$	Ponto (x,y) da imagem filtrada I' no domínio espacial
m	Quantidade de linhas da matriz da imagem I
n	Quantidade de colunas da matriz da imagem I
p	Quantidade ímpar de linhas da matriz de vizinhança centrada em $I(x,y)$
q	Quantidade ímpar de colunas da matriz de vizinhança centrada em $I(x,y)$
$I(x,y)$	Ponto (x,y) da imagem I no domínio espacial
$\nabla I(x,y)$	Vetor gradiente de $I(x,y)$
$\mathbf{G}(x,y)$	Vetor gradiente de $I(x,y)$
$\frac{\partial}{\partial x}$	Operador diferencial em x
$\frac{\partial}{\partial y}$	Operador diferencial em y
\mathbf{i}	Vetor direção em x
\mathbf{j}	Vetor direção em y
$ \mathbf{G}(x,y) $	Módulo do vetor gradiente Luminosidade
S_V	Máscara de Sobel vertical
S_H	Máscara de Sobel horizontal
$\theta(x,y)$	Orientação do do vetor gradiente
L	Luminosidade
a	Coordenada cromática verde-vermelho

b	Coordenada cromática azul-amarelo
D_E	Distância euclidiana
R_1	Intensidade de Vermelho do <i>pixel</i> 1
G_1	Intensidade de Verde do <i>pixel</i> 1
B_1	Intensidade de Azul do <i>pixel</i> 1
R_2	Intensidade de Vermelho do <i>pixel</i> 2
G_2	Intensidade de Verde do <i>pixel</i> 2
B_2	Intensidade de Azul do <i>pixel</i> 2
L_1	Luminosidade do <i>pixel</i> 1
a_1	Coordenada cromática verde-vermelho do <i>pixel</i> 1
b_1	Coordenada cromática azul-amarelo do <i>pixel</i> 1
L_2	Luminosidade do <i>pixel</i> 2
a_2	Coordenada cromática verde-vermelho do <i>pixel</i> 2
b_2	Coordenada cromática azul-amarelo do <i>pixel</i> 2

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Cromatografia em Camada Delgada	14
1.2	Deteccção nas placas cromatográficas	14
1.3	O fator de retardamento (R_f)	16
1.4	Área da mancha	16
1.5	Objetivos	17
1.5.1	Objetivo Geral	17
1.5.2	Objetivos específicos	17
1.6	Trabalhos Correlatos	18
2	EMBASAMENTO TEÓRICO	21
2.1	Aquisição	21
2.2	Pré-processamento	21
2.3	Conversão	22
2.4	Redimensionamento da imagem	22
2.5	Filtragem	22
2.5.1	Filtro da mediana	23
2.6	Deteccção de bordas	23
2.7	Perfil de Intensidade	25
2.8	Distância euclidiana	26
3	METODOLOGIA	28
4	EXPERIMENTOS E RESULTADOS	30
4.1	Base de dados	30
4.2	Segmentação dos perfis cromatográficos	30
4.3	Caracterização dos perfis cromatográficos	34
4.4	Comparação dos perfis cromatográficos	38
4.5	Código desenvolvido	39
4.6	Alguns resultados comparativos	40
5	CONCLUSÃO	44
5.1	Trabalhos futuros	44
	REFERÊNCIAS BIBLIOGRÁFICAS	46

	APÊNDICES	48
	APÊNDICE A – RESUMO DAS FUNÇÕES DESENVOLVIDAS . . .	49
A.1	Função principal: <i>main.m</i>	49
A.2	Função <i>detectaBordas.m</i>	49
A.3	Função <i>perfil.m</i>	50
A.4	Função <i>salvaFaixas.m</i>	50
A.5	Função <i>salvaDados.m</i>	50
A.6	Função <i>comparaFaixas.m</i>	50
	APÊNDICE B – CÓDIGOS DAS FUNÇÕES DESENVOLVIDAS . . .	51
B.1	Código <i>main.m</i>	51
B.2	Código <i>detectaBordas.m</i>	54
B.3	Código <i>perfil.m</i>	57
B.4	Código <i>salvaFaixas.m</i>	58
B.5	Código <i>salvaDados.m</i>	58
B.6	Código <i>comparaFaixas.m</i>	64

1 INTRODUÇÃO

O avanço tecnológico tem permitido aos computadores uma maior capacidade de processamento de dados e às câmeras digitais uma melhor resolução de imagens. A combinação dos dois tem possibilitado a cientistas e pesquisadores(as) a aplicação das técnicas de processamento digital de imagens. Atualmente, é possível se obter câmeras digitais com uma alta qualidade de imagem e diversas funcionalidades com preços acessíveis. Da mesma forma, os telefones inteligentes (*smartphones*) têm tido uma crescente melhoria em seu processamento e na qualidade das fotos obtidas.

Por consequência, essa acessibilidade tem facilitado a aplicação de ferramentas de processamento de imagens para detecção de objetos, pessoas e outras diversas aplicações. No domínio da química e/ou biologia pode-se citar o processamento de imagens em alguns estudos de quantificação de substância (concentração ou quantidade de partículas, por exemplo) por cromatografia de camada delgada, através de processamento das imagens dos cromatogramas obtidos, seja com fotos tiradas de câmeras fotográficas (HESS, 2007), de celulares (YU et al., 2016) ou de escâneres (VALVERDE; THIS; VIGNOLLE, 2007).

A cromatografia é uma técnica que permite a separação de solutos em uma mistura (DEGANI; CASS; VIEIRA, 1998; COSKUN, 2016). Nesse processo, a mistura passa por duas fases, conhecidas por estacionária e móvel. As diferentes propriedades das substâncias fazem com que elas interajam de maneiras diferentes com essas duas fases. No caso da separação, utiliza-se de propriedades da substância, como a polaridade, a solubilidade, a carga iônica, a massa e o volume. Além disso, a cromatografia pode estar conectada com outros métodos físicos de identificação como espectrofotometria ultravioleta (UV), infravermelha (IV), espectrometria de massas e ressonância magnética nuclear, dentre outros.

Um dos tipos mais comuns de cromatografia é a Cromatografia em Camada Fina, também chamada de Cromatografia em Camada Delgada (CCD), que é um dos tipos de cromatografia mais simples e baratos (SHERMA; FRIED, 2003) e que é frequentemente utilizada como método qualitativo de análise (HESS, 2007). Muitos procedimentos realizados em estudos de toxicologia ambiental, de química de alimentos, de pesticidas, assim como nas indústrias química, cosmética e farmacêutica utilizam a CCD para analisar os componentes de uma mistura (BELE; KHALE, 2010).

Além disso, tem-se o método quantitativo que é capaz de estabelecer a quantidade de analitos (componentes a serem analisados) presentes nas amostras. Porém, a determinação da quantidade e/ou concentração de substâncias químicas em amostras é usualmente feita por técnicas de cromatografia de alta eficiência (CÉSAR et al., 2007; JULIÃO et al., 2003). Visto que essas

técnicas são mais caras, Hess (2007) propõe um método barato para análises tanto qualitativas quanto quantitativas de CCD utilizando câmera digital.

Por essa razão, propõe-se aqui um método de análise automática das fotos das placas cromatográficas preparadas no laboratório de análises químicas do Instituto Federal do Espírito Santo (IFES) campus Vila Velha fazendo-se uso de algoritmos implementados em computador, uma vez que a extração e/ou identificação das camadas e de suas cores são mais precisas ao se utilizar ferramentas computacionais de processamento de imagens.

1.1 Cromatografia em Camada Delgada

A CCD é uma técnica de cromatografia líquido-sólida desenvolvida sobre uma placa, geralmente de vidro, onde tem-se uma fina camada de um material adsorvente, sendo normalmente utilizada a sílica-gel, conhecido como fase estacionária (DEGANI; CASS; VIEIRA, 1998). Na placa, identifica-se a linha onde as amostras são aplicadas. Ela é então colocada numa cuba cromatográfica que contém um solvente (ou uma mistura de solventes), também conhecido como a fase móvel. A quantidade de fase móvel deve ser tal que não ultrapasse a linha de aplicação. Então, por ação da capilaridade, a fase móvel ascende na placa carregando a mistura, onde alguns compostos vão sendo mais adsorvidos que outros devido a interação com a fase estacionária (DEGANI; CASS; VIEIRA, 1998).

Como algumas substâncias são incolores, é necessário algum processo de revelação (espectrofotometria ultravermelha (UV), por exemplo) para que se tornem visíveis na placa. Nela, algumas substâncias da mistura ficam mais retidas que outras, observa-se sua separação e obtém-se o seu perfil cromatográfico (DEGANI; CASS; VIEIRA, 1998). A classificação dos tipos de substâncias é vista na placa de CCD em diferentes cores e posições, sendo denominado cromatograma. A Figura 1 mostra o perfil cromatográfico de uma amostra de espinafre e a identificação de suas substâncias. O perfil cromatográfico, ou cromatograma, é semelhante para uma mesma amostra se mantida a mesma fase estacionária e as proporções dos solventes da fase móvel, além dos reveladores utilizados para caracterização das manchas.

1.2 Detecção nas placas cromatográficas

Neste estudo, a proposta é analisar fotos de placas cromatográficas, onde cada placa possui amostras de uma mesma planta. Cada amostra é representada por uma faixa com manchas coloridas, conhecida por perfil cromatográfico ou cromatograma. Além disso, na placa, tem-se uma linha preta (perpendicular a todas as faixas) que é o ponto de aplicação de cada amostra, tomada como referência (Figura 2).

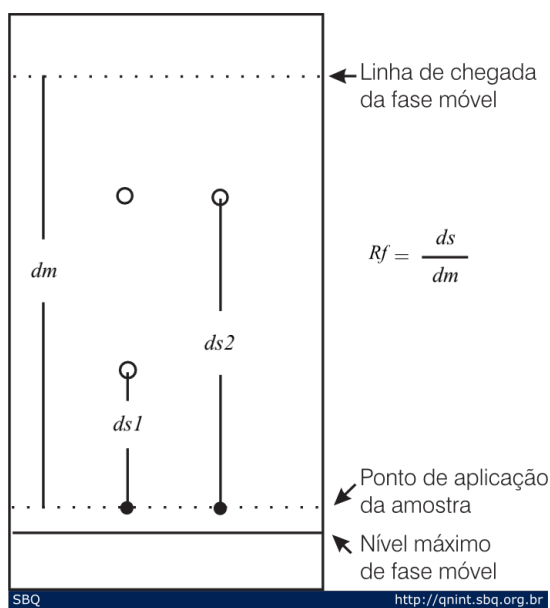
Utilizando o processamento de imagem, cada perfil será processado por vez e então cada mancha será detectada pela sua cor e pela distância da linha preta (referência) até seu centro, que será calculado pelo *script*.

1.3 O fator de retardamento (R_f)

O valor do fator de retardamento ou fator de retenção é dado pela razão da distância percorrida pelo centro da mancha sobre a distância percorrida pela fase móvel (IUPAC, 2014, p. 1304), ou seja, é a distância normalizada do centro de uma mancha até a linha de aplicação da amostra. Assim, é calculado o fator de retardamento para cada mancha, o que permite conhecer a posição de cada uma delas. A Figura 3 apresenta essa relação, onde, ds é a distância do ponto de aplicação ao centro da substância e dm é a distância do ponto de aplicação até a frente do solvente.

Esse valor é um parâmetro físico da substância e pode ser utilizado para indicar de qual se trata se: o solvente, a fase estacionária, as condições ambientais, dentre outros fatores, forem os mesmos (BELE; KHALE, 2010).

Figura 3 – O valor do fator de retardamento: a relação de ds e dm



Fonte: Degani, Cass e Vieira (1998) - Adaptado.

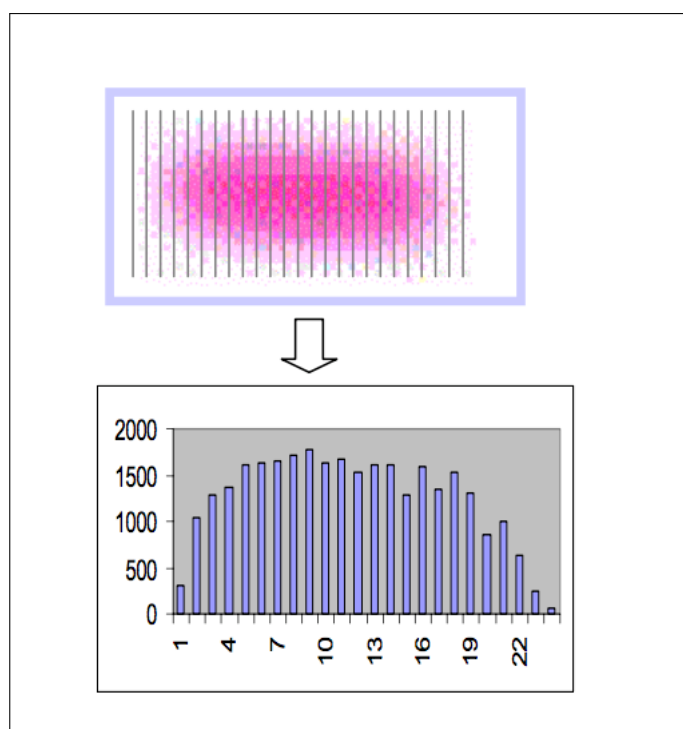
1.4 Área da mancha

As manchas encontradas na CCD tendem à distribuição gaussiana (VALENTE; AUGUSTO; RIEDO, 2003), isso quer dizer que a quantidade do analito tende a diminuir do centro até as bordas (Figura 4).

A área da mancha é proporcional à quantidade de pontos (ou *pixels*) nela presente, e também à quantidade de soluto (concentração da substância) na amostra. Este valor de área pode ser encontrado fazendo uma aproximação da curva pelo método de integração numérica trapezoidal e assim calculando sua área ou integrando a área sob a curva (VALENTE; AUGUSTO; RIEDO, 2003).

Essa área pode ser utilizada para encontrar a concentração da substância (ou analito) presente na amostra através de sua curva analítica ou curva de calibração.

Figura 4 – Distribuição das quantidades (figura inferior) de analitos em uma mancha (figura superior)



Fonte: Valente, Augusto e Riedo (2003).

1.5 Objetivos

1.5.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver um algoritmo utilizando técnicas de processamento de imagens para comparação automática de perfis cromatográficos obtidos de plantas medicinais pela técnica de cromatografia em camada delgada.

1.5.2 Objetivos específicos

- Desenvolver um algoritmo em plataforma de simulação computacional (MATLAB®), composto de sub-rotinas, capazes de:

- Identificar as linhas na placa;
 - Identificar os perfis cromatográficos;
 - Calcular o valor de R_f para cada mancha;
 - Identificar a cor no centro da mancha;
- Comparar o perfil cromatográfico, a cor e o R_f de cada faixa com os já existentes para identificação das substâncias.

1.6 Trabalhos Correlatos

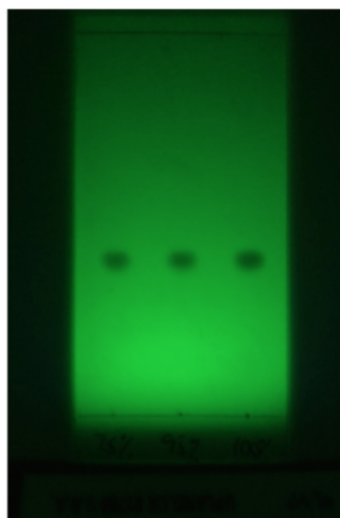
Em seu trabalho, Yu et al. (2016) propuseram a caracterização da autenticidade de drogas usando um celular. No estudo, foi criado um suporte 3D com encaixes para um celular, uma lâmpada UV alimentada por bateria e para uma placa de CCD. Além disso, foram desenvolvidos dois algoritmos de processamento de imagens para analisar as imagens tiradas pelo celular. O primeiro para processar as imagens internamente no celular e o segundo para analisá-las no MATLAB®. Como resultado, ambos algoritmos retornaram medidas equivalentes aos resultados advindos de um instrumento comercial (densitômetro).

Um exemplo de placa utilizada pelos autores pode ser vista na Figura 5. Inicialmente eles utilizaram um filtro Gaussiano (GONZALEZ; WOODS, 2002), que serve para suavizar a imagem e retirar os ruídos em altas frequências, porém, como consequências houve o “borramento” da linha base da imagem, dificultando sua detecção. Após a filtragem, aplicou-se um filtro para detecção automática de bordas, neste caso, para detectar as linhas de base e a linha de frente de solvente para identificar e recortar a região de interesse da imagem. Em seguida, aplicaram um limiar para separar o fundo dos pontos da marcação vistos na imagem. Os autores ainda utilizaram outros algoritmos para caracterização destes.

Já os autores Olech et al. (2012) utilizaram um *software* para realizar uma análise quantitativa preliminar das placas. Inicialmente, aplicaram o filtro da mediana (GONZALEZ; WOODS, 2002) para remoção de ruídos provenientes da câmera digital e, então, aplicaram um filtro passa altas, que elimina baixas frequências, para remover o efeito da iluminação não-homogênea. Depois traçaram o perfil de intensidade (ver seção 2.7) de uma faixa (Figura 6) para encontrar a área das manchas de interesse.

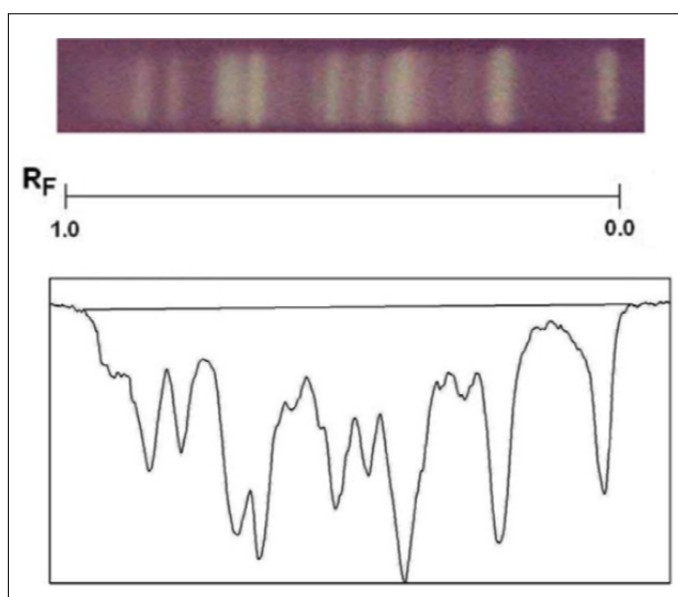
De acordo com Sousa, Mendonça e Campilho (2008), a detecção da placa é geralmente o primeiro passo para análise de imagens de cromatografia. Os autores fizeram análise de várias técnicas utilizadas por diversos autores e então apresentaram a abordagem utilizada por eles, baseada em GMM (*Gaussian Mixture Model*). Primeiro converteram a imagem para nível de cinza e aplicaram o filtro gaussiano passa-baixas para remover ruídos do fundo. A detecção

Figura 5 – Placa CDD com iluminação UV



Fonte: Yu et al. (2016).

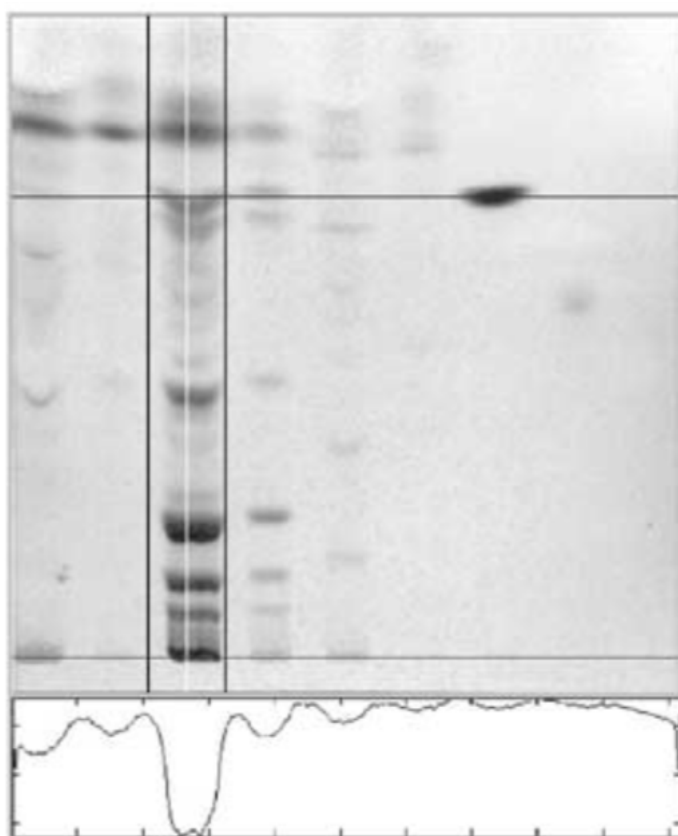
Figura 6 – Perfil cromatográfico e seu perfil de intensidade



Fonte: Olech et al. (2012).

automática das manchas foi realizada através do perfil de intensidade horizontal da imagem da placa (Figura 7) e, então, fizeram a projeção vertical do centro da mancha para obter uma estimativa do seu perfil de intensidade. Após, utilizaram a morfologia matemática aplicando a operação de *closing* (GONZALEZ; WOODS, 2002) para detectar a intensidade do fundo. Este resultado foi então usado para subtrair do perfil de intensidade aproximado da lâmina. Ao final, utilizaram a mistura de funções gaussianas para detectar cada mancha, onde cada uma estava associada à uma curva gaussiana. Desta forma, os valores de média, desvio padrão e valor de pico foram utilizados para caracterizar a posição e área da mancha correspondente.

Figura 7 – Placa CCD e seu perfil de intensidade

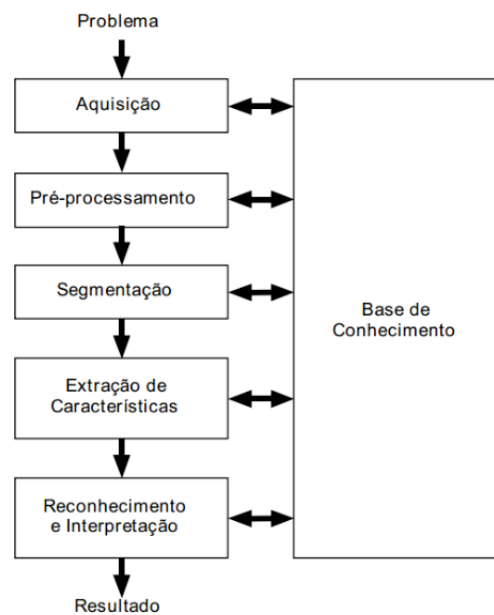


Fonte: Sousa, Mendonça e Campilho (2008) - Adaptado.

2 EMBASAMENTO TEÓRICO

De acordo com Filho e Neto (1999), um Sistema de Visão Artificial (SVA) é definido como um sistema capaz de processar e interpretar imagens de cenários reais. A Figura 8 apresenta as etapas desse sistema. Sendo as placas de cromatografia das plantas o cenário real, o algoritmo a ser desenvolvido será responsável por processar e interpretar suas imagens.

Figura 8 – Etapas de um sistema de visão artificial



Fonte: Filho e Neto (1999).

2.1 Aquisição

O dispositivo de carga acoplada (*Charge Coupled Device*) e o semicondutor de metal-óxido complementar (CMOS – *Complementary Metal-Oxide Semiconductor*) são sensores utilizados em câmeras digitais. Eles transformam a luz refletida em sinais elétricos que são então convertidos em bits por um conversor analógico-digital. Apesar desses sensores serem monocromáticos, eles são capazes de registrar a cor através de filtros de cor.

2.2 Pré-processamento

Antes da imagem seguir para as etapas subsequentes de processamento, pode ser necessário que ela passe por algum pré-processamento, seja para corrigir imperfeições advindas da aquisição, seja para auxiliar operações que serão feitas para extração das características.

2.3 Conversão

As imagens no padrão RGB possuem três matrizes de imagens. Estas três matrizes correspondem a três diferentes canais: vermelho (R – *Red*), verde (G – *Green*) e azul (B – *Blue*) (FILHO; NETO, 1999). Isso quer dizer que cada *pixel* da imagem apresenta três valores, cada um correspondente a um dos canais, onde cada canal é, em geral, de 8-bits, ou seja, para cada canal, cada *pixel* (ou ponto) possui um valor de 0 a 255.

Então, para otimizar o tempo de processamento, as imagens devem ser convertidas para escala de cinza, em que cada *pixel* da imagem possui um valor de 0 a 255, sendo 0 o valor correspondente ao preto e 255 ao branco.

Existem alguns métodos para conversão de imagens coloridas para escala de cinza. Na Equação (1), por exemplo, o valor da intensidade de cinza do *pixel* é dado pela média simples de seus valores de intensidade de vermelho, verde e azul (KLETTE, 2014)

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

onde:

C , R , G e B representam, respectivamente, os valores inteiros das intensidades nos canais cinza, vermelho, verde e azul.

2.4 Redimensionamento da imagem

Para ampliar e para reduzir uma imagem é necessário a aplicação da operação de (*upsample*) e de decimação (*downsample*), respectivamente. Como a decimação reduz o número de *pixels* em uma imagem, faz-se necessário recalcular o valor de intensidade desses novos pontos que depende de sua vizinhança. Para redimensionar imagens, os seguintes métodos são utilizados: vizinho mais próximo, interpolação bilinear e interpolação bi-cúbica (HAN, 2013).

A fim de obter resultados melhores, ou seja, de imagens mais suaves, utiliza-se a interpolação bi-cúbica. Esta interpolação inclui os 16 vizinhos mais próximos de um ponto e tende a preservar melhor detalhes finos do que nos métodos citados anteriormente. Contudo, seu custo computacional é mais elevado. Uma das funções utilizadas para calcular a intensidade dos novos pontos é a *B-spline* cúbica, mas também pode-se utilizar, por exemplo, polinômios de *Lagrange*.

2.5 Filtragem

O objetivo da filtragem é muitas vezes o de melhorar a qualidade das imagens. As técnicas de filtragem podem ser utilizadas para suavizar imagens e amenizar ruídos intrínsecos ao processo de captura de imagens digitais (filtros passa-baixas), como também para realçar detalhes em

imagens (filtros passa-altas).

Além disso, os filtros podem ser aplicados no domínio da frequência ou no espacial. No caso do domínio espacial, as técnicas de filtragem são aplicadas diretamente sobre os *pixels* da imagem por meio da convolução (FILHO; NETO, 1999). Já no domínio da frequência, a imagem é transformada para domínio de Fourier e é multiplicada por um filtro. Aplica-se então a inversa de Fourier ao produto da imagem para retornar ao domínio espacial.

2.5.1 Filtro da mediana

O filtro da média e o filtro da mediana são ambos utilizados para remoção de ruídos da imagem, porém o primeiro não é capaz de preservar bordas e detalhes da imagem. Já o da mediana é um método não-linear e apresenta bom desempenho em imagens que apresentam ruído impulsivo (FILHO; NETO, 1999).

Assim, o filtro da mediana é implementado da seguinte forma: Seja a imagem, em nível de cinza, uma matriz I de inteiros positivos com m linhas e n colunas. Dada as coordenadas (x,y) de um *pixel* em I , a vizinhança desse *pixel* é uma matriz $p \times q$ em torno do mesmo (Figura 9), onde p e q são dois números inteiros positivos e ímpares. O valor do *pixel* em (x,y) na imagem filtrada I' será dado pela mediana dos valores de sua vizinhança $p \times q$, ou seja, existem $p.q$ valores a serem ordenados, e o valor central dessa sequência será a intensidade do *pixel* em $I'(x,y)$.

Figura 9 – Exemplo de imagem I com seis linhas e seis colunas, onde o *pixel* $I(3,3)$ está em destaque, assim como sua vizinhança 3×3

20	122	222	67	12	221
25	213	232	56	34	12
76	123	50	12	32	1
128	45	51	42	21	11
23	9	54	67	23	22
246	2	29	49	23	22

Fonte: Produção do próprio autor.

2.6 Detecção de bordas

De acordo com Filho e Neto (1999), em uma imagem em nível de cinza, a borda é definida como o limite entre duas regiões onde os valores de intensidade são diferentes. Para detectar esta descontinuidade, faz-se uso do gradiente da imagem. Este é um vetor (Equação (2)) calculado a

partir da variação da imagem nas direções x e y (Equações (3) e (4))

$$\nabla I(x,y) = \mathbf{G}(x,y) = \frac{\partial I(x,y)}{\partial x} \mathbf{i} + \frac{\partial I(x,y)}{\partial y} \mathbf{j} \quad (2)$$

$$\frac{\partial I(x,y)}{\partial x} = \frac{I(x+1,y) - I(x-1,y)}{2} \quad (3)$$

$$\frac{\partial I(x,y)}{\partial y} = \frac{I(x,y+1) - I(x,y-1)}{2} \quad (4)$$

onde:

$\nabla I(x,y)$, $\mathbf{G}(x,y)$ é o vetor gradiente de $I(x,y)$;

$\frac{\partial}{\partial x}$ e $\frac{\partial}{\partial y}$ são operadores diferenciais em x e y ;

\mathbf{i} e \mathbf{j} determinam a direção dos eixos x e y .

Segundo Cabral (2018), para a realização da detecção de bordas em uma imagem em nível de cinza é necessário: filtrar a imagem para sua suavização e eliminação de ruídos, calcular suas derivadas parciais em x e y e o módulo do vetor gradiente (Equação (5)), efetuar a limiarização da imagem ressaltada para achar as bordas mais fortes e, então, pode-se rejeitar e incluir bordas.

$$|\mathbf{G}(x,y)| = \sqrt{\left(\frac{\partial I(x,y)}{\partial x}\right)^2 + \left(\frac{\partial I(x,y)}{\partial y}\right)^2} \quad (5)$$

Onde:

$|\mathbf{G}(x,y)|$ equivale ao módulo do vetor gradiente.

Existem diferentes métodos para detectar bordas, dentre os quais pode-se destacar: Sobel, Prewitt, Roberts e Canny (CABRAL, 2018).

O método de Sobel utiliza as seguintes máscaras (vertical e horizontal) da Equação (6)

$$\mathbf{S}_V = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \mathbf{S}_H = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (6)$$

Convolvendo a imagem com essas máscaras, obtém-se os gradientes vertical e horizontal, respectivamente.

O método Canny tem a vantagem de detectar bordas de largura de um *pixel* e fornece bordas em regiões de menor contraste. As etapas deste método são (CABRAL, 2018):

- Suavização da imagem;

- Cálculo do gradiente da imagem, além do módulo utiliza-se também a direção do vetor gradiente (Equação (7))

$$\theta(x, y) = \arctan \left(\frac{\partial I(x, y)}{\partial y} / \frac{\partial I(x, y)}{\partial x} \right) \quad (7)$$

- Para cada *pixel* é calculada a direção que melhor se aproxima da do gradiente. Se o contraste de um *pixel* for menor do que um de seus vizinhos na direção do gradiente, este é suprimido, ou seja, tem seu valor colocado em zero;
- Limiarização com histerese – neste caso são escolhidos dois limiares: um inferior e um superior. Se o *pixel* tem intensidade de gradiente maior que o limite superior, é uma borda. Então verifica-se os seus vizinhos na direção perpendicular ao gradiente, se suas intensidades de gradiente forem maiores que o limite inferior, são bordas.

2.7 Perfil de Intensidade

Para encontrar a distância do centro de uma mancha até a linha de aplicação será utilizado o *plot* do perfil cromatográfico, ou perfil de intensidade. Essa curva trará no eixo das abscissas o comprimento, e no eixo das ordenadas a soma das intensidades (em escala de cinza) dos pontos.

A Figura 10 apresenta um corte de um perfil cromatográfico delimitado pelo retângulo amarelo e a Figura 11 mostra o seu *plot* (perfil de intensidade). Este *plot* foi obtido através de um *software* livre de processamento de imagens chamado *ImageJ* que é usado, por exemplo, em Olech et al. (2012).

Pode-se notar que a área selecionada inclui a linha base e que esta pode ser vista no *plot* (Figura 11) como um grande pico para baixo. Há dois picos grandes para baixo, porém pode-se identificar a linha base como o primeiro, pois é mais estreito. Como visto na seção 2.3, este pico pra baixo significa que está mais próximo do preto do que do branco.

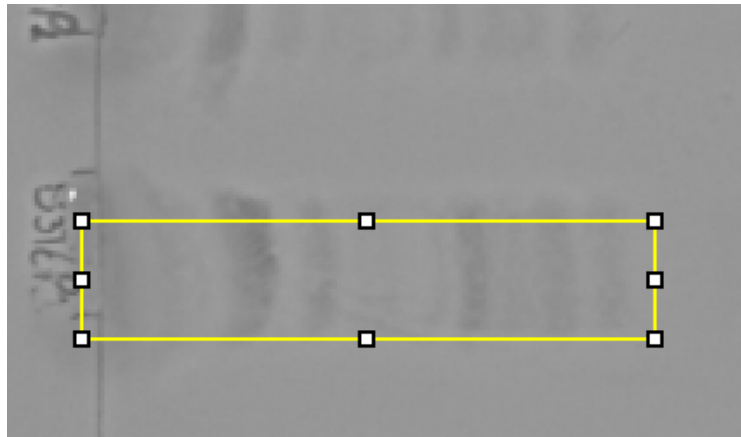
Diferente do que foi aplicado à Figura 10, a largura da área selecionada deve ser igual a maior largura da faixa (VALVERDE; THIS; VIGNOLLE, 2007). Dessa forma, a área também estará sendo contabilizada no perfil (ver seção 1.4).

Portanto, como cada mancha segue um perfil gaussiano, a sua posição pode ser dada pelo pico da curva. Logo, sabendo-se a posição da linha base, a posição da linha de frente da fase móvel, a posição aproximada das manchas e a posição do pico, pode-se encontrar o valor *Rf* de cada mancha através da Equação (8)

$$Rf = \frac{ds}{dm} \quad (8)$$

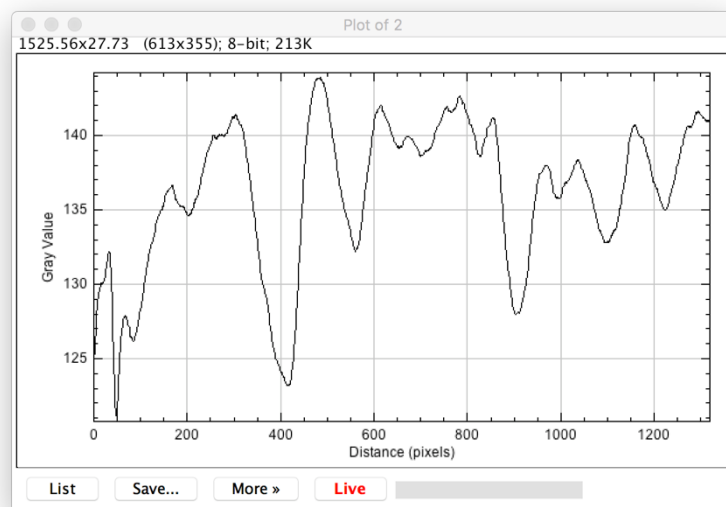
onde: ds representa a distância do pico da mancha até a linha base e dm representa a distância da frente do solvente até a linha base.

Figura 10 – Seleção do perfil cromatográfico



Fonte: Produção do próprio autor.

Figura 11 – Perfil de intensidade - gerado pelo *ImageJ*



Fonte: Produção do próprio autor.

2.8 Distância euclidiana

O modelo RGB já foi mencionado anteriormente na seção 2.3, porém, outro espaço utilizado no processamento de imagens é o Lab, também chamado de CIE $L^*a^*b^*$. No espaço de cores Lab tem-se também três dimensões para cada *pixel*. No modelo RGB cada *pixel* tem valor de vermelho, verde e azul, contudo, no modelo Lab cada *pixel* tem um valor de luminosidade (L - *lightness*), uma coordenada cromática verde-vermelho (a) e uma azul-amarelo (b). Este espaço

é capaz de descrever todas as cores perceptíveis ao olho humano (MOKRZYCKI; TATOL, 2009).

Para saber se duas cores são iguais ou próximas, uma medida geralmente usada é a distância euclidiana, que é utilizada para encontrar similaridade entre dois objetos. No caso de uma imagem no espaço RGB, a distância entre dois *pixels* pode ser calculada de acordo com a Equação (9)

$$D_E = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2} \quad (9)$$

onde:

D_E é a distância euclidiana;

R_1 , G_1 e B_1 são os valores de intensidade de vermelho, verde e azul do *pixel* 1;

R_2 , G_2 e B_2 são os valores de intensidade de vermelho, verde e azul do *pixel* 2.

De maneira similar, no espaço Lab a comparação entre a cor de dois *pixels* pode ser encontrada como mostra a Equação (10) (MOKRZYCKI; TATOL, 2009)

$$D_E = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2} \quad (10)$$

onde:

L_1 , L_2 equivalem aos valores de luminosidade dos *pixels* 1 e 2, respectivamente;

a_1 , a_2 equivalem aos valores das coordenadas verde-vermelho dos *pixels* 1 e 2;

b_1 , b_2 equivalem aos valores das coordenadas azul-amarelo dos *pixels* 1 e 2.

3 METODOLOGIA

A metodologia apresentada se baseia no método de medição em que, através da construção de uma eurística se busca extrair dados que irão compor uma informação pelo especialista em cromatografia. No caso específico, os testes buscam mostrar se é possível uma determinação automática da semelhança entre diferentes amostras de plantas através de algoritmos que implementam técnicas de processamento de imagens digitais para fins de comparação de dados de cor e de distância em uma imagem de placa cromatográfica.

Para a base de dados foram disponibilizadas cinco imagens de duas placas de cromatografia, onde cada uma contém seis perfis cromatográficos. O método proposto não se baseia em validação de resultado por experimentação, mas somente é uma implementação de uma eurística para a obtenção de métricas em cromatografia, logo, a quantidade de amostras não expressa significado quantitativo ou qualitativo para o algoritmo, pois não há necessidade de validar a eurística de medição.

Assim, o processo conta, ordenadamente, com as seguintes etapas:

1. Aquisição das fotos das placas obtidas de uma câmera digital, fornecidas pelo laboratório de análises químicas do Instituto Federal do Espírito Santo (IFES). As placas são resultados da aplicação da técnica de cromatografia em camada delgada em plantas de Aroeira.
2. Redução da imagem para realização de alguns procedimentos. Não há necessidade da resolução total da imagem, visto que é bem alta e, conseqüentemente, algumas etapas de processamento podem se tornar mais complexas gerando o mesmo resultado;
3. Detecção da placa na imagem. A imagem é uma foto da placa com seu entorno;
4. Detecção da região de interesse. Ou seja, a região que apresenta as manchas na placa;
5. Segmentação dos seis perfis cromatográficos presentes na placa;
6. Aplicação do filtro da mediana nos perfis cromatográficos para correção do ruído produzido nas fotos durante sua aquisição;
7. Cálculo da posição das frações detectadas dentro do perfil cromatográfico. Para isso, precisa-se saber a posição em x e y das manchas/substâncias detectadas;
8. A partir das posições das manchas, calcula-se o valor R_f para cada mancha detectada. Este valor é uma métrica importante para a comparação dos perfis;
9. Identificação da cor de cada mancha como uma combinação de vermelho, verde e azul (RGB). A cor é uma métrica importante para comparação de substâncias nos perfis;

10. Comparação de dois perfis cromatográficos por vez utilizando os valores R_f e cor.

Todas as etapas apresentadas na metodologia serão detalhadas no capítulo que segue.

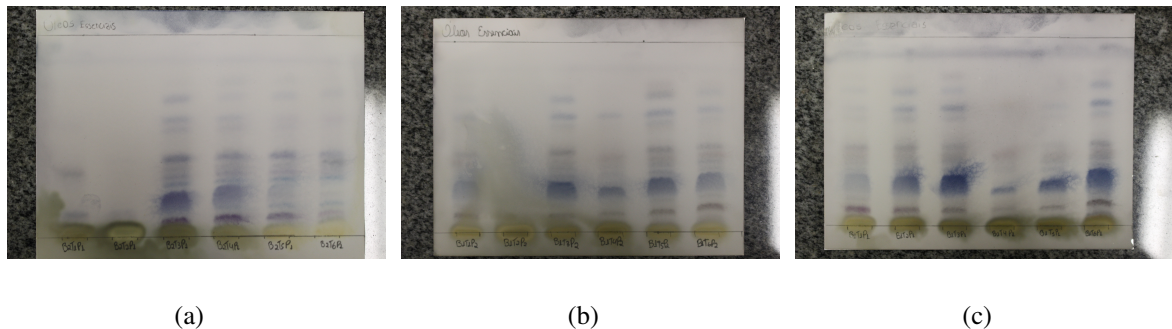
4 EXPERIMENTOS E RESULTADOS

4.1 Base de dados

Ao desenvolver este trabalho foi observado que algumas das placas estavam borradas (Figura 12) e foram desconsideradas. Ao final, restaram cinco fotos (duas placas) para o estudo, todavia, mesmo com uma quantidade baixa de amostras, foi possível criar um padrão para a análise comparativa automática dos perfis cromatográficos. Além disso, as seguintes considerações foram feitas:

- Todas as placas possuem a mesma estrutura como pode ser visto na Figura 13: as bordas internas, que são uma linha inferior de aplicação do solvente e uma superior para identificação do óleo; e as bordas externas, que são as quatro linhas que delimitam a placa;
- As linhas das placas são paralelas às bordas da foto;
- Todas as fotos foram tiradas à mesma distância mantendo os mesmos padrões de foco e de iluminação, com a câmera paralela à placa;
- Todas as placas possuem 6 (seis) perfis cromatográficos que serviram de base para a elaboração do algoritmo.

Figura 12 – Imagens descartadas

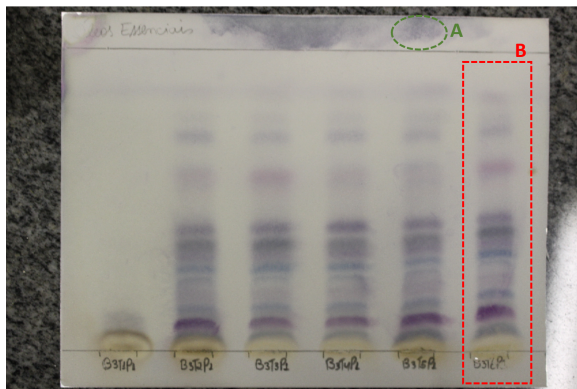


Fonte: Produção do próprio autor.

4.2 Segmentação dos perfis cromatográficos

As placas utilizadas na pesquisa seguem o padrão visto na Figura 13. Nelas, há uma linha na parte superior que identifica o tipo de óleo/amostra e outra linha na parte inferior que determina o ponto de aplicação das amostras. Essas linhas são denominadas bordas internas, enquanto que as bordas das placas são denominadas bordas externas. Cada placa possui seis amostras diferentes que devem ser separadas para comparação de suas substâncias.

Figura 13 – Exemplo de imagem da placa de cromatografia a ser segmentada. A: Mancha do revelador; B: Corrida cromatográfica de uma amostra



Fonte: Produção do próprio autor.

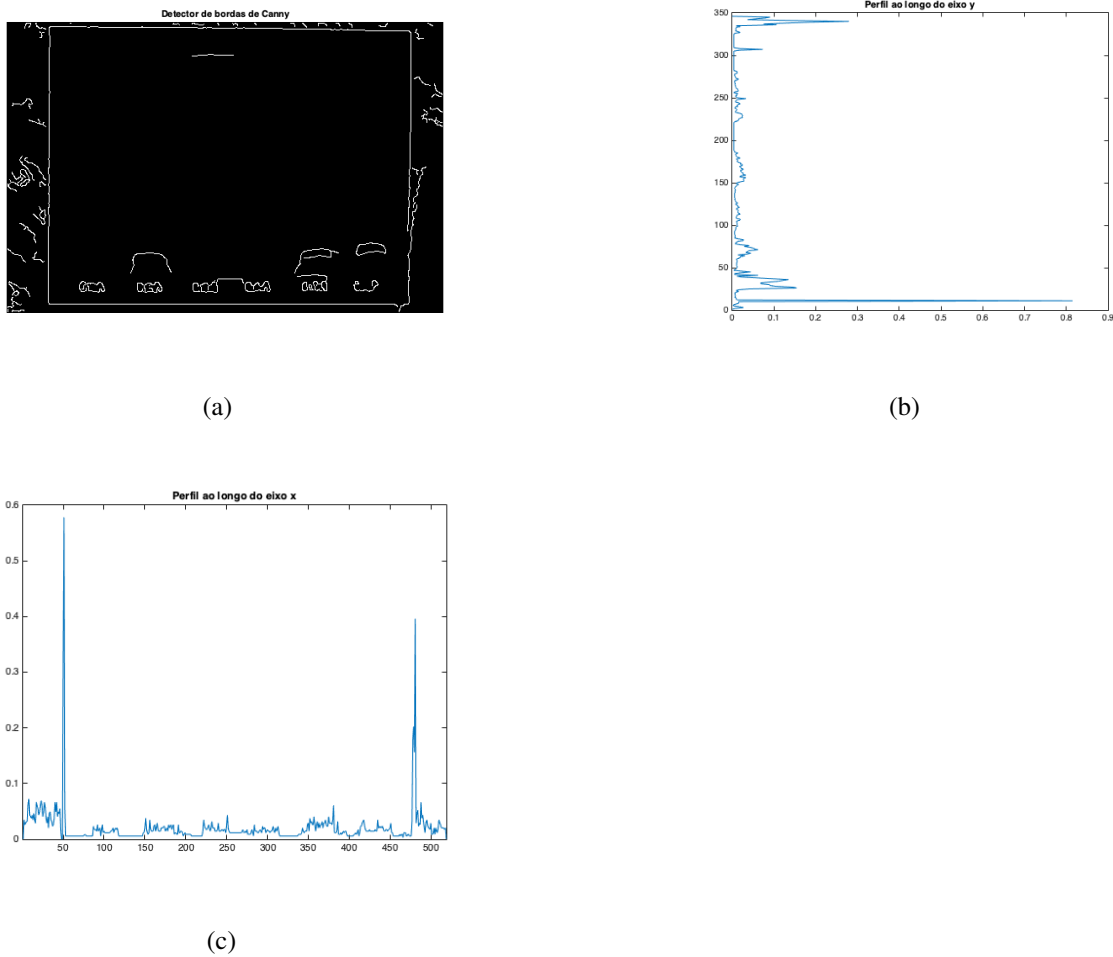
O perfil de intensidade já poderia ser traçado com esta imagem, porém, pode-se observar que há uma variedade de irregularidades que interferem no perfil das faixas, por exemplo, as manchas na parte superior (manchas do revelador). Por isso, uma das primeiras etapas é recortar a imagem de forma que reste apenas a parte que interessa da placa, do ponto de vista do processamento de imagem, ou seja, as seis faixas.

Por se tratar de uma imagem com resolução muito alta ($5184 \times 3456 \text{ pixels}$), foi feita uma redução da imagem para 10% do tamanho original utilizando a interpolação bi-cúbica. O detector de bordas de Canny foi aplicado na nova imagem e as bordas externas ficaram em evidência. O perfil de intensidade dos valores binários da imagem segmentada foi traçado, tanto para extensão em x quanto em y , como pode ser visto na Figura 14(b)(c). Assim, os pontos com os maiores valores (*pixels* mais claros) são identificados e reconhecidos como borda.

O método Canny foi o que apresentou melhores resultados para as imagens adquiridas, conservando as bordas da placa em quase toda sua totalidade (Figura 14(a)). Além disso, o perfil de intensidade das bordas também correspondeu ao esperado, podendo-se observar nas Figuras 14(a) e (b) os picos em destaque correspondentes às bordas da placa. Finalizada a etapa de detecção da placa, a próxima é encontrar a região de interesse contendo unicamente as seis faixas da placa.

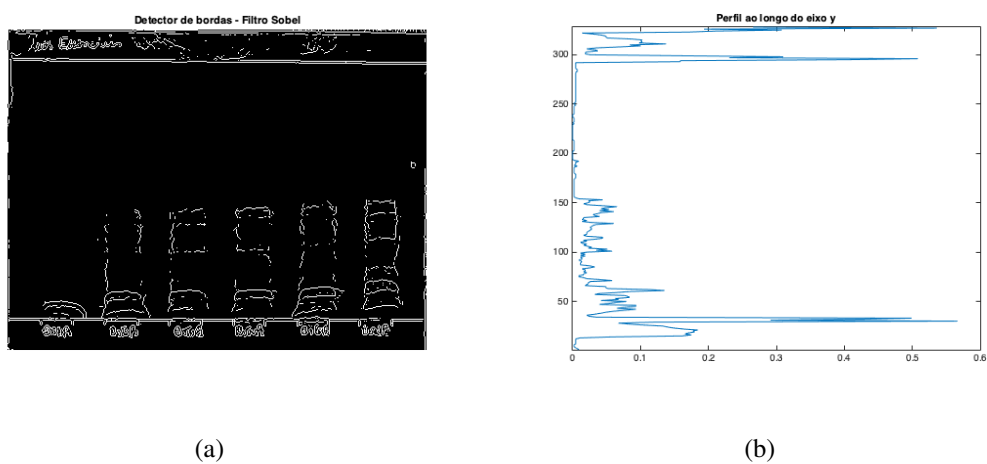
A partir da posição das bordas, a imagem reduzida foi recortada de forma a restar apenas a placa com suas bordas internas. A partir daí outra segmentação foi realizada, neste caso, o filtro utilizado foi o de Sobel. O perfil ao longo de y foi traçado e as bordas internas identificadas (Figura 15). Neste caso, o método que apresentou melhor extração de bordas nas amostras foi o de Sobel. Da mesma forma que na etapa anterior, o perfil de intensidade permitiu a identificação da posição das bordas através de seus picos.

Figura 14 – Aplicação do detector de bordas Canny. (a) Imagem binária da placa com as bordas detectadas através do método de Canny. (b) Perfil de intensidade de (a) na horizontal. (c) Perfil de intensidade de (a) na vertical



Fonte: Produção do próprio autor.

Figura 15 – Aplicação do detector de bordas de Sobel. (a) Imagem binária da placa com as bordas detectadas através do método de Sobel. (b) Perfil de intensidade de (a) na horizontal

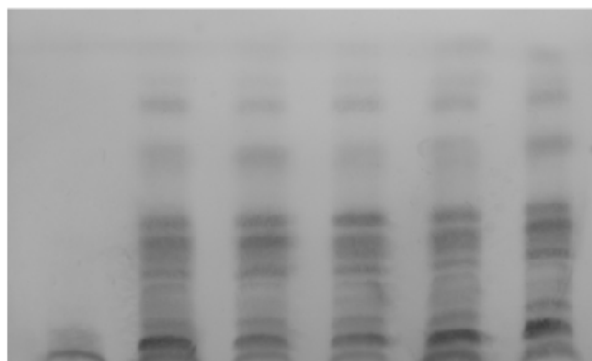


Fonte: Produção do próprio autor.

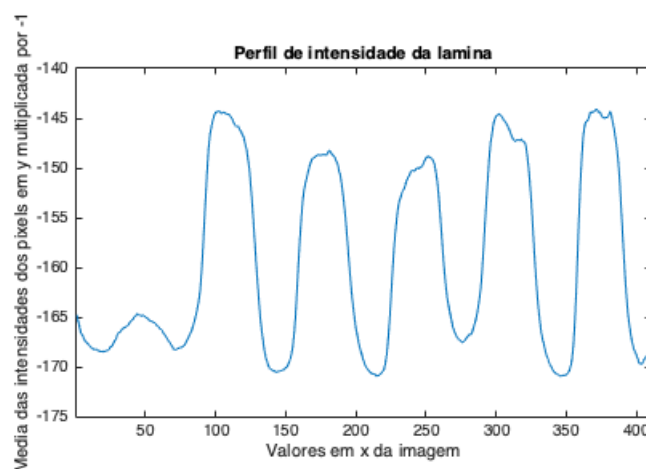
Então, com a posição das bordas externas e internas, a imagem reduzida pôde ser recortada de maneira a restar apenas a região de interesse, como mostra a Figura 16(a). Em posse desta nova imagem, o perfil de intensidade foi traçado (Figura 16(b)). Nele, pode-se identificar cada uma das seis faixas e observar que todas seguem uma distribuição similar à gaussiana.

Dada a curva gerada, realizou-se então a localização dos mínimos que correspondem ao intervalo entre faixas. Logo, na imagem original, cada uma das faixas foi cortada a partir destes pontos intermediários. Neste caso, as posições dos pontos foram transpostas para o tamanho da imagem real e cada faixa foi salva individualmente no diretório em que o código é executado. As faixas recortadas podem ser vistas na Figura 17.

Figura 16 – placa recortada e seu novo perfil de intensidade. (a) Região de interesse encontrada após detecção das bordas. (b) Perfil de intensidade de (a)



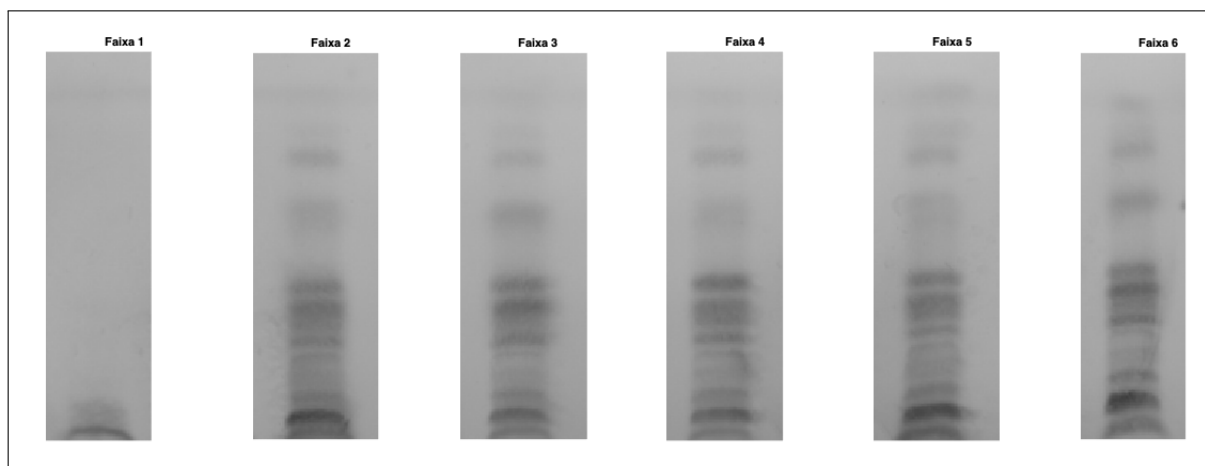
(a)



(b)

Fonte: Produção do próprio autor.

Figura 17 – As seis faixas obtidas a partir do perfil de intensidade da Figura 16



Fonte: Produção do próprio autor.

Como todas as placas presentes nas imagens fornecidas seguiam um mesmo padrão, a detecção das faixas através da identificação das variações na curva representada pela Figura 16(b) foi capaz de identificar as seis faixas das placas sem necessidade de adaptação do código.

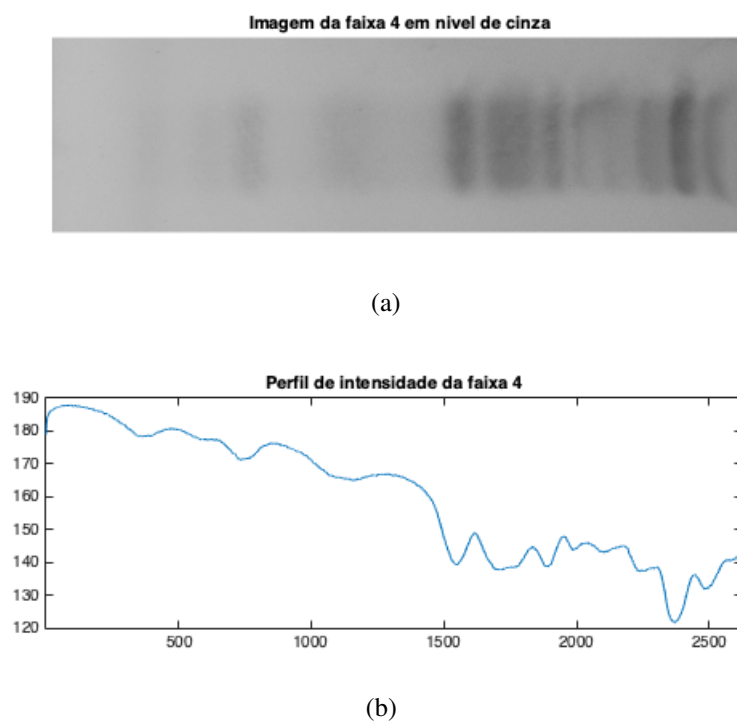
4.3 Caracterização dos perfis cromatográficos

Nesta etapa, cada perfil cromatográfico já foi salvo e está preparado para extração das suas características. Estas são: o valor R_f e; os valores RGB de cada mancha identificada. Além disso, também são salvas a matriz da imagem e a posição x e y do centro de cada mancha a fim de auxiliar na plotagem da faixa. Porém, antes de extrair estas características é necessário processar as informações do perfil cromatográfico.

Primeiramente, cada faixa é convertida em nível de cinza e filtrada utilizando o filtro da mediana, para correção das imperfeições que possam surgir na aquisição fotográfica e, então, seu perfil de intensidade é calculado na sua maior dimensão para extrair as características de cada mancha. Ao visualizar o perfil da Figura 18, percebe-se uma decrescência devido à intensidade luminosa da foto que não é uniforme. Para corrigir tal fenômeno foi feita uma correção da linha de base (Figura 19) através da função **detrend** do MATLAB®. Após essa correção foi possível observar no perfil cromatográfico que cada vale corresponde a uma mancha. Além disso, nota-se que a transição das manchas equivale a um pico no perfil cromatográfico, pois a intensidade de cinza no fundo é superior (branco equivale a 255 e preto a 0) àquela das manchas.

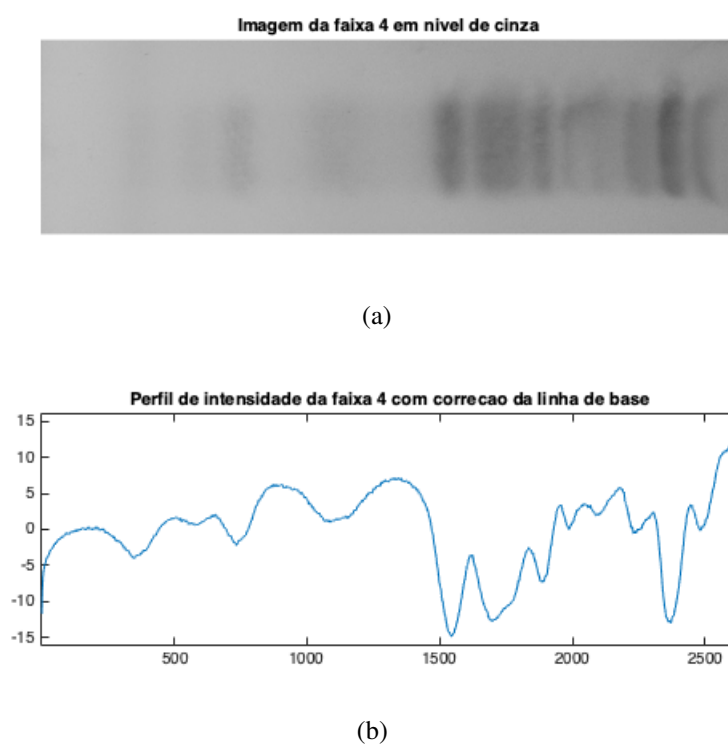
Assim, com o vetor do perfil de intensidade corrigido, ou seja, com a correção da linha de base, pode-se realizar então a identificação dos picos. Para isso, o vetor é multiplicado por menos um, visto que desta maneira o que antes era vale se tornará um pico. Então, o centro de cada mancha identificada corresponderá a um pico. A Figura 20(a) mostra a imagem colorida da faixa e a

Figura 18 – Perfil de intensidade sem correção da linha de base



Fonte: Produção do próprio autor.

Figura 19 – Perfil de intensidade com correção da linha de base

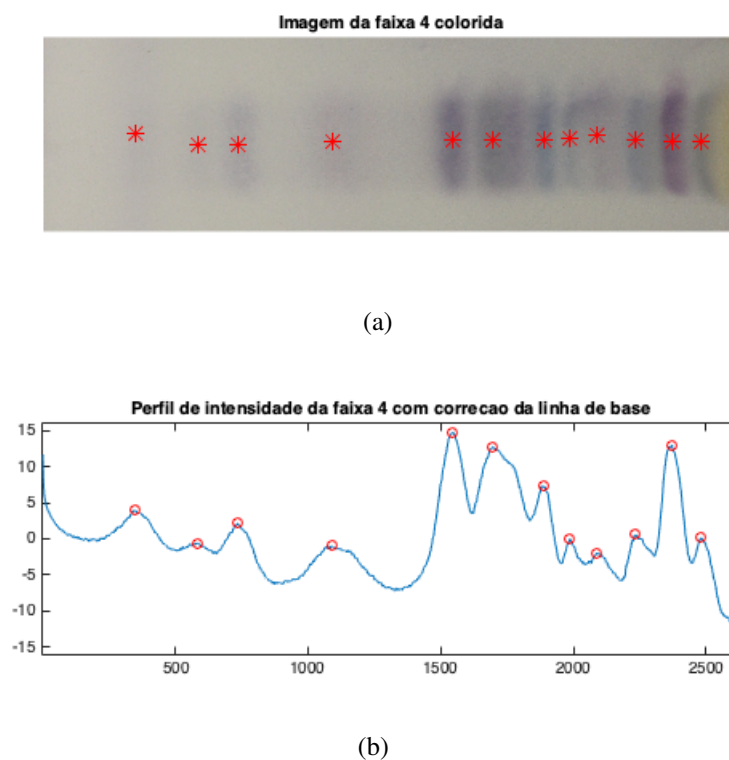


Fonte: Produção do próprio autor.

Figura 20(b) o seu perfil de intensidade com identificação dos seus picos.

Ao analisar a Figura 20 como um todo, observa-se que os picos do perfil de intensidade correspondem de maneira satisfatória aos centros das manchas. Porém, deve-se notar que quando o intervalo entre as manchas não é bem definido, pode acontecer de uma mancha não ser detectada. Isto ocorre porque ao definir o valor mínimo de pico na função *findpeaks* do MATLAB®, responsável por identificar os picos, foi necessário escolher um valor que permitisse encontrar os picos de forma adequada. Se este valor fosse muito pequeno, vários picos que não correspondem à manchas seriam identificados, por isso, no caso em que os picos são realmente pequenos, pode ser que eles não sejam identificados corretamente.

Figura 20 – Faixa colorida seu perfil de intensidade com as manchas detectadas. (a) Faixa colorida e, em vermelho, os centros das manchas identificadas. (b) Perfil de intensidade de (a) e marcação de seus picos



Fonte: Produção do próprio autor.

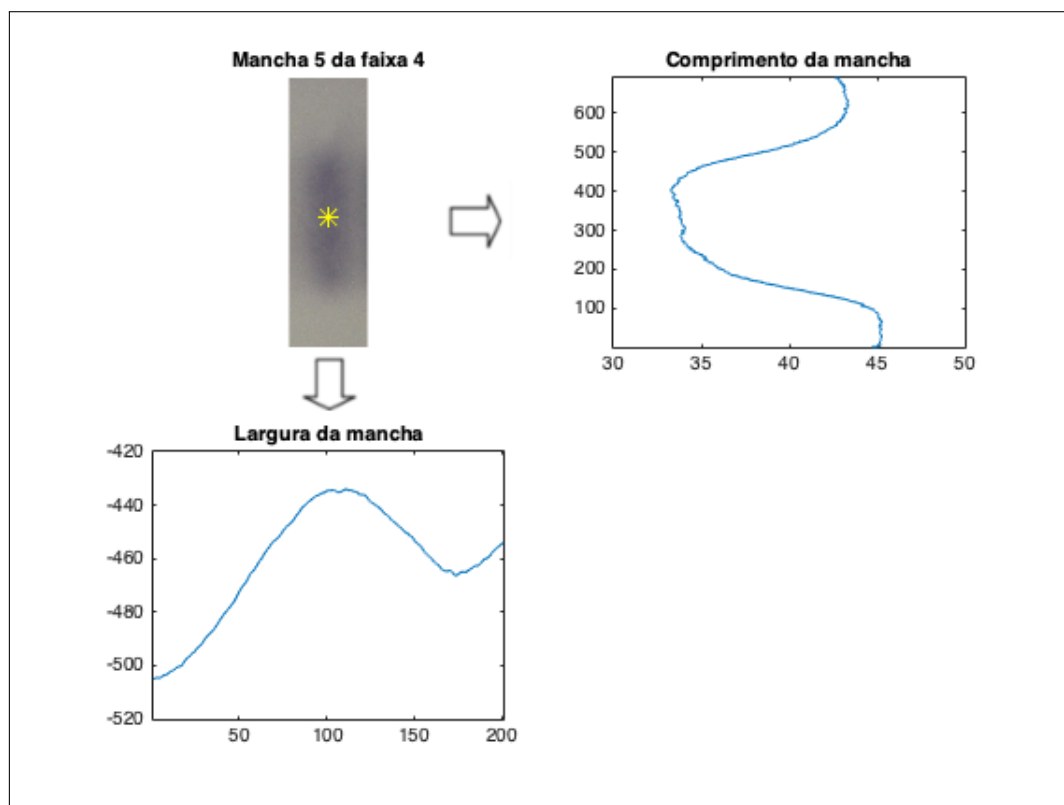
Identificadas as manchas, o próximo passo é identificar seus centros, ou seja, sua posição na outra dimensão. Para isso, cada mancha foi recortada da faixa e teve seu perfil de intensidade traçado nos dois eixos (Figura 21), o valor do pico pode ser corrigido, pois agora tem-se uma referência mais local. O tamanho recortado de cada mancha depende do desvio padrão da distância entre os picos identificados, ou seja, neste caso tem-se o centro da mancha e considera-se a largura do recorte igual a dois desvios padrão devido ao intervalo de confiança permanecer em 95%. No perfil de intensidade relativo ao comprimento da banda, nota-se a tendência a uma distribuição

gaussiana, porém não necessariamente é visto um pico. Para achar o centro, foi considerado então a média dos dois pontos onde a média do vetor intensidade corta o perfil.

A forma que o centro da mancha em y foi encontrado gerou diferentes valores para cada mancha, pois dependia da distribuição da própria mancha na placa. A Figura 20(a) apresenta a posição dos centros de acordo com o método explicado anteriormente. Nela, nota-se que apesar dos centros não estarem numa mesma linha, eles estão bem próximos.

No caso de uma mancha apresentar um centro muito destoante da média dos demais, ela tem sua posição corrigida para o centro da faixa. Isto ocorre no caso em que as manchas são muito claras e o fundo acaba interferindo na sua curva. Na realidade, o centro de todas as manchas poderiam ser considerados como o centro da faixa, porém para se ter uma referência de cor mais individual para cada mancha, optou-se por encontrar uma posição de centro para cada mancha (Figura 21).

Figura 21 – Uma mancha selecionada e seus perfis de intensidade horizontal e vertical



Fonte: Produção do próprio autor.

Então, com o centro de cada mancha calculado, é possível calcular o valor médio da sua cor no seu centro. Este será considerado a referência de cor da mancha. Para isso, faz-se a média dos valores RGB de uma janela de 5×5 pixels em volta do centro de cada mancha. Já o valor de R_f é dado pela divisão da posição do pico de cada mancha pela distância entre o ponto de aplicação da amostra e a linha de chegada da fase móvel, que é dada simplesmente pela largura da imagem.

Sendo assim, as características de cada mancha identificada da faixa são salvas e estão prontas para serem comparadas.

4.4 Comparação dos perfis cromatográficos

Finalizadas as etapas anteriores, agora é possível comparar as substâncias presentes em cada amostra de uma ou várias placas. Para isso, o *script* solicita ao usuário a escolha de dois perfis para comparar, seja da imagem selecionada ou do caminho de dois arquivos *.mat* contendo as características dos perfis. Então, o perfil com mais substâncias identificadas é usado como referência, ou seja, através de seus pontos são procurados correspondentes no outro perfil fornecido. Esta correspondência primeiramente é dada pela proximidade de valores R_f . Se os valores R_f de duas manchas comparadas forem bem próximos, ou seja, se sua diferença for inferior a um limiar, sendo este limiar um valor que o usuário pode escolher, as suas cores são então comparadas. A proximidade das cores também é dada através de um limiar que também pode ser fornecido pelo usuário. Porém, neste caso, a proximidade das cores é dada pela distância euclidiana entre elas. No caso em que mais de uma mancha possua a diferença R_f inferior ao limiar, a mancha correspondente será a que possui a menor distância euclidiana.

Os limiares *default* definidos no código foram encontrados a partir de alguns testes realizados com todas as imagens e faixas. Além disso, o usuário pode escolher qual modelo de cor utilizar, se o RGB ou o Lab (alternativa de sistema de coordenadas de cores). No caso de escolher o modelo Lab, as coordenadas RGB da mancha são convertidas para as coordenadas Lab.

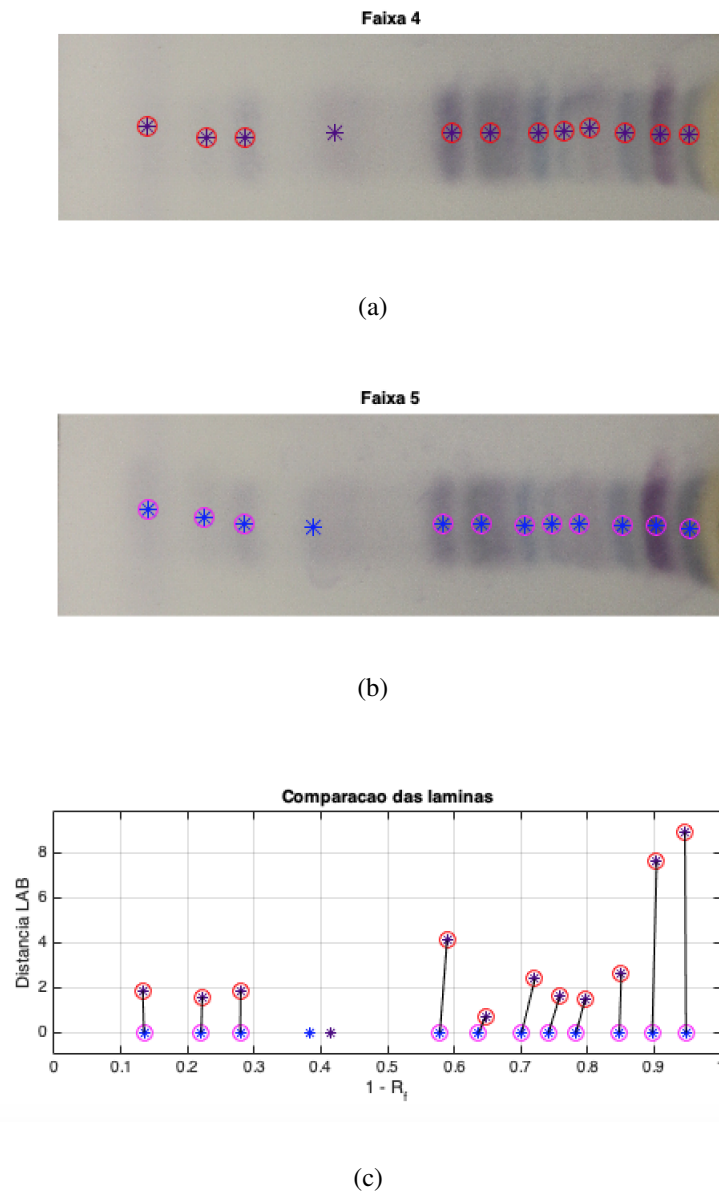
A Figura 22 mostra a imagem resultante da etapa de comparação das faixas. Nela há a imagem das duas faixas fornecidas assim como a marcação das suas manchas identificadas. A Figura 22(a) corresponde a imagem de referência, a que possui mais pontos identificados e quando contornados de um círculo vermelho significa que foi encontrada uma correspondência na outra placa, que está representada pela Figura 22(b). A Figura 22(c) é um gráfico comparativo das duas placas. Em roxo são os pontos de (a) e em azul os de (b) e em contorno vermelho e magenta os pontos de correspondência das placas. A proximidade dos pontos no eixo x significa que os valores R_f são próximos ou não. Enquanto a distância em y corresponde à proximidade das cores dos pontos identificados e correspondidos. Vale notar que quanto mais próximos, nos dois eixos, maior a chance de corresponder à mesma substância presente nas duas placas.

Na Figura 22(c) observa-se que cada faixa possui um ponto que não pôde ser identificado na outra, ou seja, não puderam ser classificados como mesma substância porque os valores de R_f estavam a uma distância superior ao limiar *default*.

Não houve a implementação de um método para analisar a qualidade das imagens adquiridas para aplicação do algoritmo de medição dos parâmetros cromatográficos, mas, em geral, para

todas as imagens fornecidas e faixas presentes nas placas foi possível a comparação através da extração automática das informações de cor e de distância (métricas usadas para a comparação de semelhança entre as plantas).

Figura 22 – As placas selecionadas e a comparação de suas substâncias (ou manchas). Limiares de R_f e de cor: 0,027 e 9



Fonte: Produção do próprio autor.

4.5 Código desenvolvido

O Apêndice A contém um resumo das funções criadas ao longo deste projeto. A *main* (A.1) é onde o código se inicia e onde é feita a chamada para as outras funções. Seguindo o que foi descrito na Metodologia (Capítulo 3), uma das primeiras funções a ser chamada é a responsável

pela detecção de bordas presente em A.2 e descrita na seção 4.2. Já o perfil de intensidade, que é uma das bases deste projeto tem seu cálculo descrito em A.3. A função *salvaFaixas* (A.4) é responsável por salvar as faixas recortadas. Além disso, as funções *salvaDados* (A.5) e *comparaFaixas* (A.6) foram descritas, respectivamente, nas seções 4.3 e 4.4.

4.6 Alguns resultados comparativos

A Tabela 1 apresenta os resultados comparativos de alguns perfis cromatográficos presentes nas imagens fornecidas. A primeira coluna diz respeito a imagem utilizada. Neste caso, cada resultado comparativo é de uma imagem diferente. A segunda coluna contém a figura que mostra o resultado comparativo. A terceira e quarta colunas apresentam o número de manchas identificadas nas faixas, sendo a primeira faixa a faixa de referência. A quinta coluna possui o número de correspondência identificadas nas duas faixas, ou seja, quantas manchas foram identificadas em ambas. Por fim, a última coluna é a relação entre o número de correspondências e o número máximo de manchas identificadas em uma das faixas. Para todos estes, a opção de comparação de cor utilizada foi através do modelo de cores Lab. Os limiares de cor e de *Rf* foram, respectivamente, 9 e 0,027.

Tabela 1 – Alguns resultados obtidos a partir do código desenvolvido

	Figura	Nº Manchas		Correspondências	%
		1ª faixa	2ª faixa		
1	Figura 23	11	6	3	27,27%
2	Figura 24	12	12	12	100,00%
3	Figura 25	12	10	10	83,33%
4	Figura 26	12	12	11	91,67%
5	Figura 27	12	12	11	91,67%

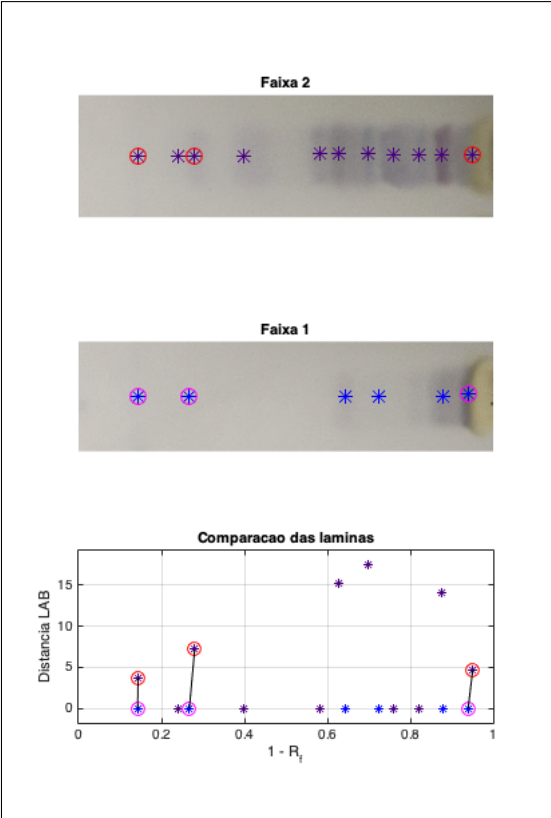
Fonte: Produção do próprio autor.

A Figura 23 mostra um caso em que visualmente já se percebe que a distribuição do perfil cromatográfico é diferente nas duas faixas, porém elas possuem três substâncias em comum que podem ser identificadas na imagem. Lembrando que o tamanho da linha indica a distância em cor das correspondências e o deslocamento em *x* a diferença nos valores de *Rf*.

A Figura 24 mostra um resultado em que todas as substâncias encontradas em uma faixa foi também encontrada na outra, obtendo-se assim uma correspondência de 100% entre ambas as faixas.

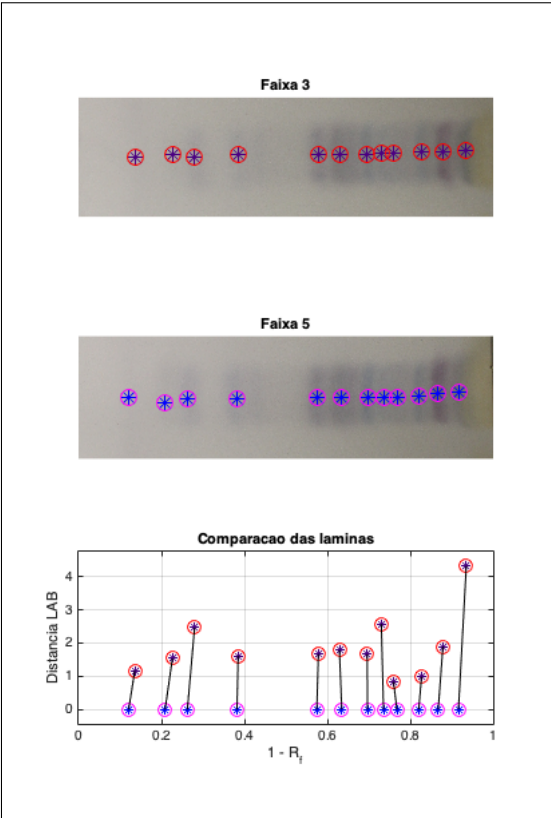
Percebe-se na Figura 25 que a segunda faixa possui menos pontos que a primeira, ou seja, o algoritmo não percebeu a variação entre as bandas e não identificou manchas em posições semelhantes à primeira faixa. Neste caso, nem todas as substâncias identificadas em uma faixa

Figura 23 – Comparação 1



Fonte: Produção do próprio autor.

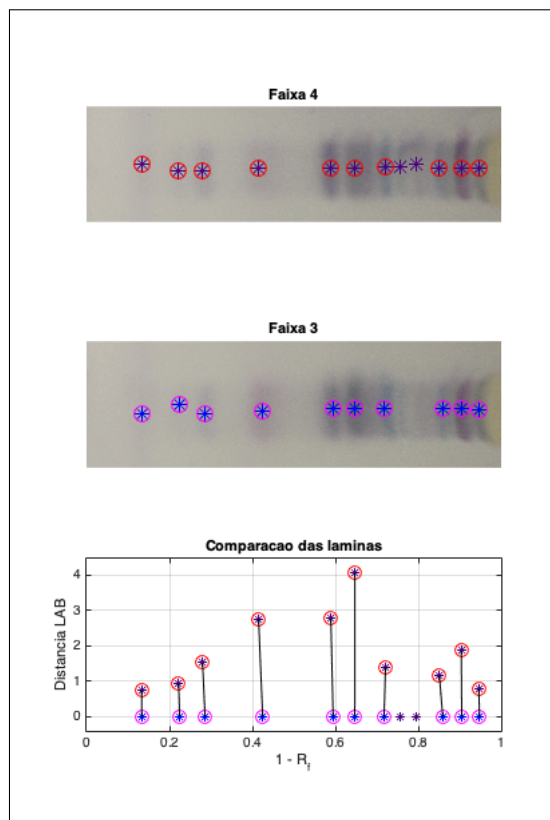
Figura 24 – Comparação 2



Fonte: Produção do próprio autor.

foi encontrada na outra.

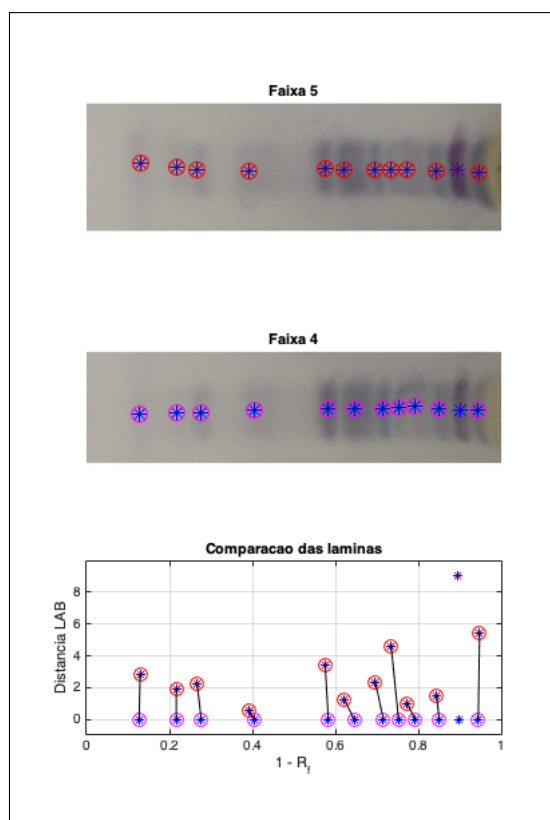
Figura 25 – Comparação 3



Fonte: Produção do próprio autor.

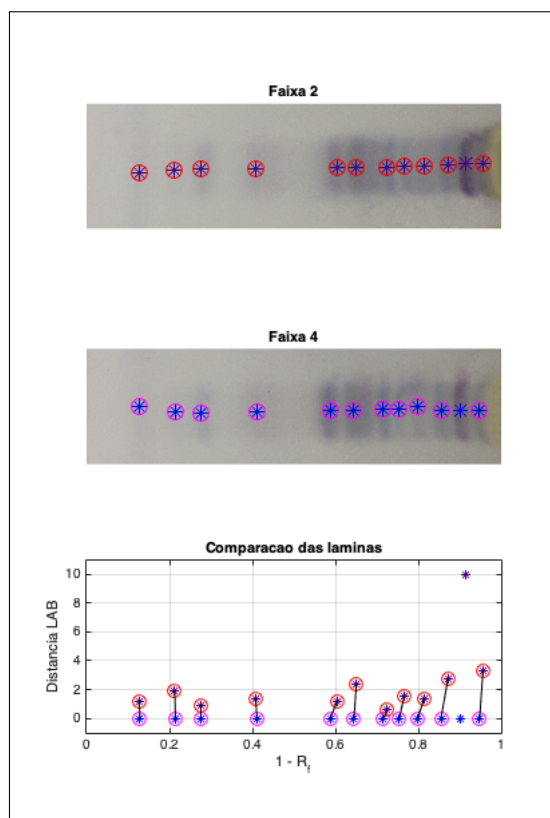
Já as Figuras 26 e 27 possuem resultados semelhantes e apenas uma mancha não foi identificada em ambas as faixas. Ao analisar os gráficos, nota-se que no critério R_f as manchas são correspondentes, porém as cores estão a uma distância euclidiana superior à tolerância fornecida ao algoritmo.

Figura 26 – Comparação 4



Fonte: Produção do próprio autor.

Figura 27 – Comparação 5



Fonte: Produção do próprio autor.

5 CONCLUSÃO

O propósito deste trabalho era fornecer uma alternativa automática para a análise qualitativa de placas cromatográficas. Para isso, foram criadas sub-rotinas para encontrar as linhas de aplicação e de chegada e detectar os perfis presentes na placa e também para caracterizar o perfil identificando os centros das manchas para calcular seu valor R_f e sua cor. A partir dessas informações, o algoritmo pôde comparar os perfis cromatográficos identificando as frações em comum. Ou seja, dados dois perfis cromatográficos de duas amostras diferentes de plantas, a partir do algoritmo desenvolvido, é possível comparar e analisar se as manchas presentes neles são iguais ou não. Isso é avaliado pela proximidade dos valores R_f e das cores. Um outro caso possível é quando um dos perfis equivale ao padrão de uma substância específica que se deseja saber se está presente na amostra.

Um dos obstáculos encontrados no desenvolvimento deste trabalho foi a quantidade limitada de placas em estudo e de fotos fornecidas, o que dificultou a parametrização de algumas funções, por exemplo, a função para encontrar os picos (*findpeaks*) se utilizada sem parâmetros, identificava vários picos. Para resolver este problema, foram colocados alguns limites de distância entre picos, largura de picos e proeminência, porém, para encontrar estes limites foram realizados vários testes que por partes funcionavam para uma faixa e não para a outra.

Em resumo, o algoritmo desenvolvido apresenta ser uma alternativa para a análise da presença de substâncias em plantas que atualmente é feita de forma manual e imprecisa, pois, além de reduzir o tempo gasto na análise, ele encontra os centros das manchas de acordo com sua distribuição e compara as cores de acordo com seus valores exatos enquanto o método visual utiliza da subjetividade e instrumentos do analisador.

5.1 Trabalhos futuros

Uma das características que pode ser melhorada é o aumento da robustez da função capaz de identificar os picos dos perfis de intensidade.

Técnicas de aprendizado de máquina poderiam ser implementadas e comparadas com a proposta utilizada se houvesse a aquisição de mais imagens. Não obstante, também a necessidade de melhorar a captura das imagens de forma a reduzir a quantidade de informações espúrias em função de sujeira, iluminação inadequada e problemas de preparo nas lâminas.

Do ponto de vista de processamento digital de imagens, pode-se estudar a possibilidade de utilizar a Transformada *Bottom Hat* para processar o fundo e facilitar a detecção de cores na análise.

Além disso, poderiam-se catalogar amostras de padrões, criando um banco de dados contendo as informações de R_f e de cor e também os parâmetros utilizados na CCD para que, nas mesmas condições, seja possível classificar as substâncias identificadas nas amostras das plantas.

Para obter uma conclusão mais específica na comparação de perfis, pode-se também desenvolver um método para calcular a probabilidade de uma amostra corresponder a determinada planta nos casos em que as manchas dos cromatogramas não são totalmente correlacionadas.

Finalmente, com o objetivo de encapsular e facilitar o uso do algoritmo de comparação de perfis cromatográficos implementado neste projeto, é possível desenvolver uma interface gráfica para o usuário.

REFERÊNCIAS BIBLIOGRÁFICAS

- BELE, A. A.; KHALE, A. An overview on thin layer chromatography. *International Journal of Pharmaceutical Sciences and Research*, v. 2 (2), p. 256–267, 2010. Citado 3 vezes nas páginas 13, 15 e 16.
- CABRAL, E. L. L. *PMR2560 – Notas de Aula de Visão Computacional Detecção de bordas*. 2018. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/4301995/mod_resource/content/0/V10%20-Deteccao%20de%20bordas.pdf>. Acessado em: 10 de fevereiro de 2019. Citado na página 24.
- COSKUN, O. Separation techniques: Chromatography. *North Clin Istanbul*, v. 3 (2), p. 156–160, 2016. Citado na página 13.
- CÉSAR, I. da C. et al. Determinação de daidzeína, genisteína e gliciteína em cápsulas de isoflavonas por cromatografia em camada delgada (CCD) e cromatografia líquida de alta eficiência (CLAE). *Revista Brasileira de Farmacognosia*, v. 17, p. 616 – 625, 12 2007. Citado na página 13.
- DEGANI, A. L. G.; CASS, Q. B.; VIEIRA, P. C. Cromatografia: um breve ensaio. *Química Nova na Escola*, v. 7, p. 21–25, maio 1998. Citado 4 vezes nas páginas 4, 13, 14 e 16.
- FILHO, O. M.; NETO, H. V. *Processamento Digital de Imagens*. Rio de Janeiro: Brasport, 1999. Citado 3 vezes nas páginas 21, 22 e 23.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 2nd. ed. New Jersey: Prentice Hall, 2002. Citado 2 vezes nas páginas 18 e 19.
- HAN, D. Comparison of commonly used image interpolation methods. *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering*, 03 2013. Citado na página 22.
- HESS, A. V. I. Digitally-enhanced thin-layer chromatography: An inexpensive, new technique for qualitative and quantitative analysis. *Journal of Chemical Education*, v. 84 (5), p. 852, maio 2007. Citado 2 vezes nas páginas 13 e 14.
- INTERNATIONAL UNION OF PURE AND APPLIED CHEMISTRY. *Compendium of Chemical Terminology*. 2nd. ed. [S.l.], 2014. Citado na página 16.
- JULIÃO, L. et al. Cromatografia em camada fina de extratos de três quimiotipos de lippia alba (mill) n.e.br. (erva-cidreira). *Revista Brasileira de Farmacognosia*, v. 13, p. 36–38, 2003. Citado na página 13.
- KLETTE, R. *Concise Computer Vision: An Introduction into Theory and Algorithms*. London: Springer-Verlag, 2014. Citado na página 22.
- MOKRZYCKI, W.; TATOL, M. Perceptual difference in l * a * b * color space as the base for object colour identification. In: . [S.l.: s.n.], 2009. Citado na página 27.
- OLECH, M. et al. Investigation of antiradical activity of plant material by thin-layer chromatography with image processing. *Food Chemistry*, v. 132, n. 1, p. 549–553, 2012. Citado 3 vezes nas páginas 18, 19 e 25.

SHERMA, J.; FRIED, B. (Ed.). *Handbook of Thin-Layer Chromatography*. 3th. ed. New York: Marcel Dekker, 2003. v. 89. Citado na página 13.

SOUSA, A. V.; MENDONÇA, A. M.; CAMPILHO, A. J. C. Chromatographic pattern classification. *IEEE Transactions on Biomedical Engineering*, v. 55, p. 1687–1696, 2008. Citado 2 vezes nas páginas 18 e 20.

VALENTE, A. L. P.; AUGUSTO, F.; RIEDO, C. R. F. *Análise Quantitativa por Cromatografia*. 2003. Disponível em: <<http://chemkeys.com/br/wp-content/themes/chemkeysbr/articleI.php?u=YW5hbGlzZS1xdWFudGl0YXRpdmEtcG9yLWNyb21hdG9ncmFmaWE=>>. Citado 2 vezes nas páginas 16 e 17.

VALVERDE, J.; THIS, H.; VIGNOLLE, M. Quantitative determination of photosynthetic pigments in green beans using thin-layer chromatography and a flatbed scanner as densitometer. *Journal of Chemical Education*, v. 84 (5), p. 1505, maio 2007. Citado 2 vezes nas páginas 13 e 25.

YU, H. et al. Characterization of drug authenticity using thin-layer chromatography imaging with a mobile phone. *Journal of Pharmaceutical and Biomedical Analysis*, v. 125, p. 85–93, jun. 2016. Citado 3 vezes nas páginas 13, 18 e 19.

Apêndices

APÊNDICE A – RESUMO DAS FUNÇÕES DESENVOLVIDAS

A.1 Função principal: *main.m*

A função inicial do projeto de onde tudo se inicia está presente no Apêndice B, seção B.1. Ela é responsável pela chamada de rotinas e funções e também permite a interação com o usuário para o fornecimento de dados e informações. Primeiro o usuário escolhe a imagem da lâmina que quer analisar. Então a imagem é reduzida e convertida para nível de cinza e passada como argumento para a função responsável por detectar as bordas: ***detectaBordas*** (seção A.2).

Então, com a imagem original sem as bordas, o código continua até a localização dos máximos e mínimos para detecção das faixas. Sabe-se que existe uma variação entre as faixas e seus intermédios, portanto foi escolhido um intervalo para calcular esta variação, quando próxima de zero corresponde a uma faixa ou a um intermédio, pois são regiões onde os valores tendem a ser próximos. Enquanto seus pontos de máxima e mínima correspondem a uma mudança, ou seja, a um intermédio entre faixa. Sendo esses vales e picos então encontrados pela função ***findpeaks*** do MATLAB®. A justificativa por não utilizar diretamente esta função baseia-se no fato de que a linha de base pode interferir e que pode existir uma faixa cujo pico não é tão proeminente. Porém, ao calcular a variação, as grandes alterações, por exemplo a mudança de faixa, são levadas em consideração. Além disso, desta forma, os ruídos e a distribuição não uniforme da luminosidade não possuem grande influência nos picos e vales encontrados.

Após a identificação das faixas, suas imagens são salvas através da função ***salvaFaixas*** (seção A.4) e suas características são calculadas e salvas pela função ***salvaDados*** (seção A.4). Por fim, em posse de todos os dados necessários, pode-se então realizar a comparação entre as faixas desejadas. Para isso, o usuário fornece a informação das lâminas e faixas que deseja comparar, e então a função de comparação (***comparaFaixas*** - seção A.6) é chamada e seu resultado apresentado. Importante salientar que ao realizar a chamada desta última é possível alterar os limiares de erro/variação de valores de R_f e cor aceitáveis pelo usuário.

A.2 Função ***detectaBordas.m***

Esta função (Apêndice B, seção B.2) é responsável por realizar a detecção das bordas das imagens de lâminas fornecidas. A imagem em nível de cinza reduzida é passada como argumento e são retornadas as posições para corte da borda exterior e da borda interior. Os pontos de corte do contorno da lâmina: dois pontos com as posições x e y que equivalem a um retângulo são encontrados a partir do detector de bordas *Canny* e do perfil de intensidade, através da função ***perfil*** (seção A.3), da imagem filtrada resultante (Figura X). Então, de maneira similar, são encontradas as posições em y das bordas horizontais da lâmina a partir do detector de bordas de *Sobel* (Figura X).

A.3 Função *perfil.m*

Esta função pode ser vista no Apêndice B, seção B.3, e é a base de todo o projeto desenvolvido pois ela é a responsável por calcular o perfil de intensidade em x e em y da imagem em nível de cinza fornecida I_g . Dada a imagem I_g de resolução 800×600 fornecida, será atribuído a cada dimensão um vetor com a distribuição de suas intensidades. Por exemplo, na dimensão x , este vetor terá 800 valores, sendo cada um deles a média das intensidades de cinza em y , ou seja, será dado pela média de 600 valores. Da mesma maneira ocorre com a dimensão em y , um vetor de 600 valores é retornado. Além disso, são retornados os valores das posições de cada uma das dimensões. Ou seja, no final são retornados quatro vetores, dois com as intensidades e dois com as posições de 1 a 800 e 1 a 600.

A.4 Função *salvaFaixas.m*

Uma simples função, presente no Apêndice B, seção B.4, que apenas salva as imagens no diretório atual. São fornecidos: a imagem original, o número referente a imagem, este serve apenas para identificar a imagem e; um vetor que possui a posição de corte das faixas.

A.5 Função *salvaDados.m*

Para cada uma das seis faixas da imagem esta função executa o que foi descrito na seção 4.3 - **Caracterização dos perfis cromatográficos**. Esta função a princípio salva as informações da imagem fornecida no início do código *main* (seção A.1). Mas é importante salientar que ela pode ser utilizada de maneira independente do resto do código para calcular as características de qualquer imagem de lâmina que o usuário possua. Para isso é possível alterar o diretório para o desejado e, retirar o *for* responsável por percorrer as seis faixas, dentro da própria função e executá-la. O código desta função está no Apêndice B, seção B.5

A.6 Função *comparaFaixas.m*

Assim como a função *salvaDados* (seção A.5), esta função teve sua rotina descrita na seção 4.4 - **Comparação dos perfis cromatográficos**. Seu código pode ser encontrado no Apêndice B, seção B.6.

APÊNDICE B – CÓDIGOS DAS FUNÇÕES DESENVOLVIDAS

B.1 Código *main.m*

```

1 clear all
2 close all
3 warning off
4
5 opcao = input(['Qual imagem deseja abrir? \nDigite um numero de 1 a ...
    5' ...
6     ' para as imagens cadastradas \nou digite 0 para enviar um ...
    diretorio' ...
7     ' \nOpcao: ']);
8 % opcao=5;
9 if opcao==0
10     dirusu=input('Caminho ate a imagem: ','s');
11     dir=strcat(dirusu)
12 elseif opcao>0 & opcao <6
13     % caso ja tenha um diretorio com as imagens, pode colocar aqui ...
    diretamente
14     dir=strcat('images/dadas/',num2str(opcao),'.JPG')
15 else
16     display('Opcao invalida')
17     return
18 end
19
20 Iorig=(imread(dir));      % Carrega a imagem de acordo com a opcao ...
    escolhida
21 [m n o]=size(Iorig);
22 Im=imresize(Iorig, 0.1); % Reduz a imagem para 10%
23 Ig=rgb2gray(Im);         % Converte a imagem para nivel de cinza
24
25 figure(1)
26 imshow(Iorig)
27 title('Imagem escolhida')
28
29 % Retira as bordas da imagem da lamina
30 [X1 X2 y1 y2 x1 x2]=detectaBordas(Ig);
31 Icut=Ig(X1+1:X2-1,y1+1:y2-1); % Corta as bordas externas
32 Icut=Icut(x1:x2+1,:);         % Corta as bordas internas
33
34 %% Perfil das laminas
35 [comp larg]=size(Icut);
36 Icut=Icut(10:comp-10,10:larg-10);
37 % @vec_x vetor com a soma dos pixels em x

```



```

38 % @vec_y vetor com a soma dos pixels em y
39 % @m vetor de valores em y (vetor da comprimento)
40 % @n vetor de valores em x (vetor do largura)
41 [vec_x, m, vec_y, n]=perfil(Icut);
42
43 % figure(2)
44 % subplot(2,1,1)
45 % title('Imagem recortada')
46 % imshow(Icut)
47 % subplot(2,1,2)
48 % plot(n,-(vec_y))
49 % xlim([1 size(vec_y,1)])
50 % title('Perfil de intensidade da lamina')
51 % ylabel('Media das intensidades dos pixels em y multiplicada por -1')
52 % xlabel('Valores em x da imagem');
53
54 %% Localizacao dos minimos/maximos
55 dt=20;
56 f=vec_y;
57 fdev=f;
58 tam=size(f);
59
60 for i=1:tam(1) % fazendo df/dt
61     if i>(tam(1)-dt)
62         fdev(i)=(f(tam(1))-f(i))/(tam(1)-i);
63     else
64         fdev(i)=(f(i+dt)-f(i))/(dt);
65     end
66 end
67
68 % encontra os vales da df/dt do perfil da lamina
69 [peaks,vales]=findpeaks(-fdev,n,'Annotate','extents','MinPeakProminence' ...
    ...
    ,0.05,'MinPeakDistance',5);
71 % figure(3)
72 % plot(n,-fdev,'*',n(vales),peaks,'or');
73 % xlim([1 larg])
74
75 %% Registro de cada faixa
76 n_faixas=6; % numero de faixas (6)
77 faixa=ones(n_faixas,2); % cria um vetor do tamanho do numero de ...
    faixas (6)
78
79 % se vales =7, encontrou um ponto de fim da lamina antes da borda
80 if size(vales,2)==7
81     % calcula a media para que a ultima faixa nao fique tao estreita
82     vales(7)=floor(0.5*(vales(7)+tam(1)));

```

```

83 elseif size(vales,2)==6      % se nao identificou, coloca no final da ...
    lamina
84     vales(7)=tam(1);
85 end
86
87 % delimita as faixas de acordo com os vales
88 for i=1:n_faixas
89     faixa(i,1)=vales(i);
90     faixa(i,2)=vales(i+1);
91 end
92
93 figure(5)    % figura que contem os cortes das faixas
94 set(gcf, 'Position', [100, 100, comp*6, larg*6])
95 for i=1:n_faixas
96     subplot(1,n_faixas,i)
97     imshow(Icut(1:length(m),faixa(i,1)-10:faixa(i,2),:))
98     title(['Faixa ' num2str(i)])
99     xlim([1 100])
100 end
101
102 %% Corta imagem colorida
103 % Para imagem inteira
104 Icolor=Iorig(X1*10:X2*10,y1*10:y2*10,:);
105 Icolor=Icolor(x1*10:x2*10, :, :);
106
107 %% Salvando faixas
108 salvaFaixas(Icolor,num2str(opcao),faixa);
109 display('Imagem de cada faixa salva no diretorio atual!')
110
111 %% Salvando caracteristicas das faixas
112 salvaDados(num2str(opcao));
113 display('Caracteristicas de cada faixa salvas no diretorio atual!')
114
115 %% Escolher laminas para comparar
116 sn=input(['Para retornar digite (1). Caso queira comparar laminas ...
    de' ...
117     ' imagens diferentes (2) \nou comparar apenas as laminas da ...
    imagem' ...
118     ' atual ' num2str(opcao) ' (3)? \n']);
119 modo='lab';
120 limiar_rf=0.027;    % default: limiar_rf=0.027
121 limiar_rgb=0.16;    % default: limiar_rgb=0.16
122 limiar_lab=9;    % default: limiar_lab=9
123 if strcmp(modo,'lab')
124     limiar_cor=limiar_lab;
125 else
126     limiar_cor=limiar_rgb;

```

```

127 end
128 if sn==3
129     lam1=input(['Escolha a partir da Figura 5 uma faixa/lamina para ...
130               ' ...
131               'comparacao: ']);
132     lam2=input(['Escolha a partir da Figura 5 outra faixa/lamina ...
133               para ' ...
134               'comparacao: ']);
135     dirData1=strcat('dado_',num2str(opcao),'_faixa_', ...
136                   num2str(lam1),'.mat');
137     dirData2=strcat('dado_',num2str(opcao),'_faixa_', ...
138                   num2str(lam2),'.mat');
139     comparaFaixas(dirData1,dirData2, limiar_rf, limiar_cor, modo);
140 elseif sn==1
141     return
142 elseif sn==2
143     verifica=input(['Deseja carregar uma imagem nova (1) ou ' ...
144                   'os dados ja estao salvos (2)? '])
145     if verifica==1
146         display(['Rode o codigo novamente e carregue imagem ...
147                 desejada.' ...
148                 ' \nAs imagens(.png) e dados(.m) das laminas serao ...
149                 salvas ' ...
150                 'no diretorio atual.']);
151     else
152         return
153     elseif verifica==2
154         % exemplo: dados/dado_3_faixa_4.mat
155         dirData1=input('Digite o nome/caminho dos dados (arquivo ...
156                       .mat) e uma lamina para comparacao: ', 's');
157         dirData2=input('Digite o nome/caminho dos dados (arquivo ...
158                       .mat) da outra lamina para comparacao: ', 's')
159         comparaFaixas(dirData1,dirData2, limiar_rf, limiar_cor, modo);
160     else
161         display('Opcao invalida');
162         return
163     end
164 else
165     display('Opcao invalida');
166     return
167 end

```

B.2 Código *detectaBordas.m*

```

1 function [X1 X2 y1 y2 x1 x2] = detectaBordas(Ig)
2 % Segmentacao das laminas - deteccao das bordas

```

```
3
4 % Encontra as bordas externas
5 BW2 = edge(Ig);
6 BW2 = edge(Ig, 'canny', 0.25);
7
8 % imshow(BW2)
9
10 % Plota perfil binarizado para localizar as retas
11 [vec, m, vec2, n]=perfil(BW2);
12 [B,I]=sort(vec, 'descend');
13 aux=I(2);
14 for i=1:10
15     if I(i)<30
16         aux=I(i);
17         break;
18     end
19 end
20
21 for i=1:10
22     if I(i)>200
23         aux2=I(i);
24         break;
25     end
26 end
27
28 X1=aux;
29 X2=aux2;
30
31 %%
32 % Encontra as 4 retas mais relevantes
33 [B2,I2]=sort(vec2, 'descend');
34
35 aux=I2(2);
36 for i=1:10
37     if I2(i)<200
38         aux=I2(i);
39         break;
40     end
41 end
42
43 for i=1:10
44     if I2(i)>200
45         aux2=I2(i);
46         break;
47     end
48 end
49
```

```
50 y1=aux;
51 y2=aux2;
52
53 Icortada=Ig(X1+1:X2-1,y1+1:y2-1);
54
55 % figure
56 % %subplot(2,2,1)
57 % imshow(BW2)
58 % title('Detector de bordas de Canny')
59 % %subplot(2,2,3)
60 % figure
61 % plot(n,vec2)
62 % xlim([1 size(BW2,2)])
63 % title('Perfil ao longo do eixo x')
64 % %subplot(2,2,2)
65 % figure
66 % plot(wrev(vec),m')
67 % title('Perfil ao longo do eixo y')
68 % %
69 % figure
70 % imshow(Icortada)
71 %%
72 % Encontra as bordas internas da placa
73 BW = edge(Icortada,'sobel',0.019);
74
75 % figure(1)
76 % imshow(BW)
77
78 [vec, m, vec2, n]=perfil(BW);
79 [B,I]=sort(vec, 'descend');
80
81 % figure
82 % % subplot(1,2,1)
83 % imshow(BW)
84 % title('Detector de bordas - Filtro Sobel')
85 % figure
86 % % subplot(1,2,2)
87 % plot(wrev(vec),m')
88 % title('Perfil ao longo do eixo y')
89 % ylim([1 size(BW,1)])
90 %%
91 [t1 t2]=size(I);
92
93 superior=1;
94 inferior=t1;
95 % limites - superior 5% t1:20% t1, inferior 80% t1: 95 % t1
96 for i=1:20
```

```
97     if I(i)>floor(0.05*t1) & I(i)<floor(0.2*t1)
98         if I(i)>superior
99             superior=I(i);
100        end
101    end
102 end
103
104 for i=1:20
105     if I(i)>floor(0.8*t1) & I(i)<floor(0.95*t1)
106         if I(i)<inferior
107             inferior=I(i);
108         end
109     end
110 end
111
112 x1=superior;
113 x2=inferior;
114
115 % Icortada=Icortada(x1:x2+1,:);
116 % figure
117 % imshow(Icortada)
118
119 end
```

B.3 Código *perfil.m*

```
1 function [Vetor, M, Vetor2, N] = perfil(Ig)
2 % Funcao que calcula o vetor de intensidade da Imagem Ig: [Vetor, M, ...
3     Vetor2, N] = perfil(Ig)
4 % @Vetor vetor com a soma dos pixels em x
5 % @Vetor2 vetor com a soma dos pixels em y
6 % @M vetor de valores em y (vetor do comprimento)
7 % @N vetor de valores em x (vetor da largura)
8
9 [m n]=size(Ig);
10 Vetor=ones(m,1);
11 Vetor2=ones(n,1);
12
13 % perfil de uma lamina
14 for i=1:m
15     Vetor(i)=sum(Ig(i,:));
16 end
17 M=1:1:m;
18 Vetor=Vetor/n;
```

```
19 for j=1:n
20     Vetor2(j)=sum(Ig(:,j));
21 end
22 N=1:1:n;
23 Vetor2=Vetor2/m;
```

B.4 Código *salvaFaixas.m*

```
1 function [ output_args ] = salvaFaixas(I, n_imagem, d)
2 % Funcao que salva as 6 faixas de uma dada placa cromatografica no ...
3   diretorio
4 % atual.
5 % saveSlices(I, n_imagem, d)
6 % I -> corresponde a imagem tamanho original
7 % n_imagem -> ao numero da imagem
8 % d -> posicao de corte das faixas
9
10 for i=1:6
11     Icut=I(:,10*(d(i,1)+0):10*(d(i,2)+10),:);
12     imwrite(Icut, strcat('images/geradas/nova_', n_imagem, '_faixa_', ...
13       num2str(i), '.png'));
14 end
15 end
```

B.5 Código *salvaDados.m*

```
1 function [ output_args ] = salvaDados(n_imagem)
2 % Funcao que salva as caracteristicas das 6 faixas: Imagem, centro e cor
3 % das manchas e valores Rf.
4
5 for i=1:6 % percorre as 6 faixas da imagem
6     Dados.n=i;
7
8     n_faixa=num2str(i);
9
10    diretorio=strcat('nova_', n_imagem, '_faixa_', n_faixa, '.png');
11    % diretorio=strcat('images/dadas/teste.png');
12
13    I_2=(imread(diretorio)); % Abre imagem da faixa
14    ori=imrotate(I_2,90); % Rotaciona
15    im_faixa_cinza=rgb2gray(ori); % Converte para nivel de cinza
```

```

16 im_faixa_cinza=medfilt2(im_faixa_cinza);    % Aplica filtro mediana
17 %%
18 % Calcula seu perfil de intensidade
19 [vec_x, m_x, vec2_x, n_x]=perfil(im_faixa_cinza(:,:));
20 y=detrend(vec2_x); % Corrige a linha de base
21
22 % figure
23 % subplot(2,1,1)
24 % imshow(im_faixa_cinza);
25 % title(['Imagem da faixa ' n_faixa ' em nivel de cinza'])
26 % subplot(2,1,2)
27 % plot(n_x,vec2_x);
28 % title(['Perfil de intensidade da faixa ' n_faixa ])
29 % xlim([1 size(im_faixa_cinza,2)])
30 %
31 % figure
32 % subplot(2,1,1)
33 % imshow(im_faixa_cinza);
34 % title(['Imagem da faixa ' n_faixa ' em nivel de cinza'])
35 % subplot(2,1,2)
36 % plot(n_x,y);
37 % xlim([1 size(im_faixa_cinza,2)])
38 % title(['Perfil de intensidade da faixa ' n_faixa ...
39 %       ' com correcao da linha de base'])
40
41 % valores encontrados apos testes realizados - podem ser alterados para
42 % conveniencia do usuario
43 minPico = 60/(3000/size(n_x,2));
44 minLarg = minPico/2;
45
46 % Encontra os picos que correspondem as manchas da faixa
47 [peaks, loc]=findpeaks(-y,n_x,'Annotate','extents',...
48     'MinPeakProminence',0.005,'MinPeakDistance',minPico,...
49     'MinPeakWidth',minLarg/1.25);
50 picos_identificados=loc;
51 %%
52 qtd=size(picos_identificados,2)+1;
53 Δ=ones(1,qtd);
54 Δ(1:qtd-1)=picos_identificados;
55 ultimo=size(im_faixa_cinza,2);
56 Δ(qtd)=ultimo-picos_identificados(qtd-1);
57 Δ2=Δ; % vetor com as diferencas
58
59 % @Δ2 contera diferenca entre os picos consecutivos
60 for t=2:qtd-1
61     Δ2(t)=Δ(t)-Δ(t-1);
62 end

```



```
63
64 % calculo do desvio padrao das distancias
65 desvio=min(100,floor(std( $\Delta$ 2)));
66
67 % @ponto_centro contera os valores dos centros em x das manchas - apenas
68 % inicializacao para ter o mesmo tamanho
69 ponto_centro=picos_identificados;
70
71 % @picos_identificados2 contera os valores dos picos corrigidos mais ...
    local
72 picos_identificados2=picos_identificados;
73
74 % para cada pico identificado, ou seja, faixa, sua posicao sera ...
    recalculada
75 for k=1:size(picos_identificados,2)
76
77     % calcula pico apenas da mancha
78     [vec_comp, pontos_comp, vec_comparg, pontos_comparg]= ...
79         perfil(im_faixa_cinza(:,max(picos_identificados(k)-desvio,1):...
80             min(picos_identificados(k)+desvio,ultimo)));
81
82     % media do perfil da mancha
83     media=mean(vec_comp);
84
85     clear loc_comp
86     % @ponto1 - o ponto onde a media corta o grafico do perfil pela ...
        1a vez
87     ponto1=1;
88     for i=1:floor(0.4*size(vec_comp,1))
89         if vec_comp(i)<media
90             ponto1=i;
91             break
92         end
93     end
94
95     % @ponto2 - o ponto onde a media corta o grafico do perfil por ...
        ultimo
96     ponto2=size(vec_comp,1);
97     for i=floor(0.6*size(vec_comp,1)):size(vec_comp,1)
98         if vec_comp(i)>media
99             ponto2=i;
100             break
101         end
102     end
103
104     % ponto do centro - a media de ponto1 e ponto2
105     ponto_centro(k)=floor((ponto1+ponto2)/2);
```

```

106
107 % perfil de intensidade para corrigir a posicao do pico identificado
108 [peaks_comp, loc_comp]=findpeaks(-vec_comparg,pontos_comparg,...
109     'Annotate','extents','MinPeakWidth', minLarg,...
110     'MinPeakProminence',0.005,'MinPeakDistance',minPico);
111
112 if loc_comp ≠ 0
113     Dados.n;
114     if size(loc_comp)>1
115         if (abs(loc_comp(2)-loc_comp(1)>100)) % mais de um pico ...
116             identificado
117             if abs(-desvio+loc_comp(1))<80 % atualiza a posicao ...
118                 do pico
119                 picos_identificados2(k)=picos_identificados(k) ...
120                 -desvio+loc_comp(1);
121             end
122             if abs(-desvio+loc_comp(2))<80 % cria um novo pico
123             picos_identificados2(size(picos_identificados2)+1)=...
124             picos_identificados(k)-desvio+loc_comp(2);
125             ponto_centro(size(picos_identificados2))=ponto_centro(k);
126             end
127         else
128             if abs(-desvio+loc_comp(1))<80 % atualiza a posicao do pico
129             picos_identificados2(k)=picos_identificados(k) ...
130             -desvio+loc_comp(1);
131             else % cria um novo pico
132             display('oi')
133             picos_identificados2(size(picos_identificados2)+1)=...
134             picos_identificados(k)-desvio+loc_comp(1);
135             ponto_centro(size(picos_identificados2))=ponto_centro(k);
136             end
137         end
138     end
139 end
140
141 % figure
142 % subplot(2,2,2)
143 % plot(wrev(vec_comp),pontos_comp) % precisa inverter o vetor
144 % title('Comprimento da mancha')
145 % ylim([1 size(vec_comp,1)])
146
147 % subplot(2,2,3)
148 % plot(pontos_comparg, -vec_comparg)
149 % title('Largura da mancha')
150 %
151 % xlim([1 size(vec_comparg,1)])
152 % subplot(2,2,1)

```

```
151 %     imshow(ori(:,picos_identificados2(k)-desvio: ...
152 %           min(picos_identificados2(k)+desvio,ultimo),:)); % opcao ...
153 %     colorida
154 %     % %
155 %     title(['Mancha ' num2str(k) ' da faixa ' n_faixa])
156 %     %     imshow(ori(:,picos_identificados(k)-desvio:...
157 %           min(picos_identificados(k)+desvio,ultimo))); % opcao ...
158 %     preto e branco
159 %     hold on
160 %     plot(desvio,ponto_centro(k),'y*','MarkerSize',10)
161 end
162 %%
163 % limites dos desvios do centro da lamina no sentido da largura da ...
164 %     faixa
165 limite1=3*size(vec_comp,1)/7;
166 limite2=4*size(vec_comp,1)/7;
167 % se os centros das faixas estiverem fora dos limites, entao o ...
168 %     centro sera
169 % atribuido ao centro da faixa
170 for i=1:size(ponto_centro,2)
171     if (ponto_centro(i) < limite1) | (ponto_centro(i) > limite2)
172         ponto_centro(i) = size(vec_comp,1)/2;
173     end
174 end
175 % figure
176 % subplot(2,1,1)
177 % imshow(ori)
178 % title(['Imagem da faixa ' n_faixa ' colorida'])
179 % subplot(2,1,2)
180 % plot(n_x,-y,n_x(picos_identificados2),peaks,'or');
181 % xlim([1 ultimo])
182 % title(['Perfil de intensidade da faixa ' n_faixa ...
183 %     ' com correcao da linha de base'])
184 % subplot(2,1,1)
185 % hold on
186 % plot(picos_identificados2,ponto_centro,'*','...
187 %     'MarkerEdgeColor','r','MarkerSize',10)
188 %
189
190 %% calculo rf
191 linha_inicio=1;
192 linha_fim=ultimo;
```

```
193 Rf=ultimo-picos_identificados2; % equivale a distancia do ponto de ...  
    aplicacao ate o centro da mancha  
194  
195 % Rf normalizado  
196 Rf=Rf/(linha_fim-linha_inicio);  
197 Dados.rf=Rf';  
198  
199 %% encontrar cor - rgb  
200 Dados.picos=picos_identificados2';  
201 Dados.centro=ponto_centro';  
202  
203 janela=5; % 5x5  
204 for i=1:size(Dados.picos,1)  
205     window_color=(ori((Dados.centro(i)-janela):...  
206         (Dados.centro(i)+janela), (Dados.picos(i)-janela):...  
207         (Dados.picos(i)+janela),1:3));  
208     window_color_r=window_color(:,1);  
209     window_color_g=window_color(:,2);  
210     window_color_b=window_color(:,3);  
211  
212     cores(i,1)=floor(mean(window_color_r(:)));  
213     cores(i,2)=floor(mean(window_color_g(:)));  
214     cores(i,3)=floor(mean(window_color_b(:)));  
215  
216 end  
217 Dados.cores_rgb=cores;  
218 Dados.imagem=ori;  
219  
220 %%  
221 save_dir=strcat('dados/dado_',n_imagem,'_faixa_',n_faixa,'.mat');  
222 % save_dir=strcat('teste.mat');  
223  
224 save(save_dir,'Dados')  
225  
226 clear Dados  
227 clear cores  
228 clear window_color  
229 clear picos_identificados  
230 clear picos_identificados2  
231 clear perfil_todo  
232 clear media  
233 clear ponto_centro  
234 clear desvio  
235 clear vec_comp  
236 clear peaks_comp  
237 end  
238 end
```

B.6 Código *comparaFaixas.m*

```
1 function [output_args] = comparaFaixas(dir1,dir2,rfDiff,colorDiff,modo)
2 % Primeiro e segundo sao os diretorios para as laminas a comparar.
3 % Terceiro e quarto parametro equivalem aos erros aceitaveis de Rf e de
4 % distancia quadratica das cores RGB.
5
6 % dir1='dados/dado_1_faixa_5.mat'
7 % dir2='dados/dado_2_faixa_5.mat'
8 % dir1='dados/dado_5_faixa_6.mat'
9 % dir2='dados/dado_4_faixa_6.mat'
10
11 first_lamina = load(dir1);
12 second_lamina = load(dir2);
13
14 % primeira lamina eh a que possui mais picos - sera de referencia
15 if size(second_lamina.Dados.rf,1)>size(first_lamina.Dados.rf,1)
16     aux=second_lamina;
17     second_lamina=first_lamina;
18     first_lamina=aux;
19     clear aux
20 end
21
22 % inicializa vetor com validacao dos Rf
23 valida_rf=ones(size(first_lamina.Dados.rf,1),1);
24 posicoes=zeros(size(first_lamina.Dados.rf,1),1);
25
26 % limite de variacao dos Rf
27 epsilon=rfDiff;
28
29 % limite da variacao em cor
30 epsilon2=colorDiff;
31
32 % inicializa vetor da distancia
33 dist_qua=zeros(size(posicoes,1),1);
34
35 novoJ=1;
36
37 % verifica valores rf
38 for i=1:size(first_lamina.Dados.rf,1)
39     for j=novoJ:size(second_lamina.Dados.rf,1)
40         aux = abs(first_lamina.Dados.rf(i)-second_lamina.Dados.rf(j));
41         % verifica limiar Rf
42         if aux < epsilon
43             if strcmp(modo, 'lab')
44                 lab1=rgb2lab(first_lamina.Dados.cores_rgb(i,:)./255);
```

```

45         lab2=rgb2lab(second_lamina.Dados.cores_rgb(j,:)/255);
46         aux_dist = sqrt(sum((lab1(:) - lab2(:)).^ 2))/1; %/5
47     else
48         rgb1=first_lamina.Dados.cores_rgb(i,:)/255;
49         rgb2=second_lamina.Dados.cores_rgb(j,:)/255;
50         aux_dist = sqrt(sum((rgb1(:) - rgb2(:)).^ 2))/1; %/5
51     end
52
53     % verifica limiar cor e no caso de mais de um Rf
54     % correspondente, valida o com menor distancia
55     if aux_dist < epsilon2 & dist_qua(i)==0 | ...
56         aux_dist < epsilon2 & aux_dist < dist_qua(i)
57         posicoes(i)=j;
58         valida_rf(i)=0;
59
60         if (novoJ < size(second_lamina.Dados.rf,1))
61             novoJ=j+1; % como correspondencia foi ...
62                         encontrada
63                         % pula para o proximo pico
64         end
65     end
66     dist_qua(i)=aux_dist;
67 end
68 end
69
70 % validacao final - 0 significa validado, se em algum dos dois casos nao
71 % foi validado, nao sera identificado como mesma substancia no final
72 valida_rf;
73
74 % inicializa variaveis
75 cont=1;
76 vec_valida=0;
77 picos_validos_2=0;
78 centros_validos_2=0;
79 distancia=0;
80 vec_valida_2=0;
81 % variaveis auxiliares para plotagem
82 for i=1:size(valida_rf)
83     if (valida_rf(i)*first_lamina.Dados.rf(i)==0)
84         vec_valida(cont)=first_lamina.Dados.rf(i);
85         distancia(cont)=dist_qua(i);
86         vec_valida_2(cont)=second_lamina.Dados.rf(posicoes(i));
87         picos_validos_2(cont)=second_lamina.Dados.picos(posicoes(i));
88         centros_validos_2(cont)=second_lamina.Dados.centro(posicoes(i));
89         cont=cont+1;
90     end

```

```
91 end
92
93 pontos1=[1-vec_valida;1-vec_valida_2];
94 pontos2=[distancia; zeros(1,size(vec_valida_2,2))];
95
96 figure(6)
97 set(gcf, 'Position', [0, 0, 400, 600])
98 subplot(3,1,1)
99 imshow(first_lamina.Dados.imagem)
100 xlim([1 size(first_lamina.Dados.imagem,2)])
101 hold on
102 plot(first_lamina.Dados.picos,first_lamina.Dados.centro,'*',...
103      'MarkerEdgeColor',[75 0 130]/255,'MarkerSize',10)
104 plot(first_lamina.Dados.picos,not(valida_rf).*first_lamina.Dados.centro,...
105      'ro','MarkerSize',12)
106 hold off
107 title(['Faixa ' num2str(first_lamina.Dados.n)])
108 subplot(3,1,2)
109 imshow(second_lamina.Dados.imagem)
110 xlim([1 size(second_lamina.Dados.imagem,2)])
111 hold on
112 plot(second_lamina.Dados.picos,second_lamina.Dados.centro,'*',...
113      'MarkerEdgeColor','b','MarkerSize',10) % [128 0 0]/255
114 plot(picos_validos_2,centros_validos_2,'mo','MarkerSize',12)
115 hold off
116 title(['Faixa ' num2str(second_lamina.Dados.n)])
117 subplot(3,1,3)
118 plot(1-(first_lamina.Dados.rf),dist_qua,'*',...
119      'MarkerEdgeColor',[75 0 130]/255,'MarkerSize',5)
120 xlim([0 1])
121 hold on
122 plot(1-second_lamina.Dados.rf,0,'*', 'MarkerEdgeColor','b','MarkerSize',5)
123 plot(1-vec_valida,distancia,'ro','MarkerSize',10)
124 plot(1-vec_valida_2,0,'mo','MarkerSize',10)
125 ylim([-0.1*max(dist_qua(:)),1.1*max(dist_qua(:))])
126 plot(pontos1,pontos2,'k')
127 title('Comparacao das laminas')
128 xlabel('1 - R_f')
129 ylabel(['Distancia ' upper(modos)])
130 grid on
131 end
```