

Universidade Federal de Mato Grosso do
Sul
Faculdade de Computação



**Análise de Imagens: Método
Alternativo de Mensuração do
Comprimento de Plântulas para
Testes de Vigor**

Marcos Pinheiro Vilhanueva

Dissertação de Mestrado

Orientação: Prof. Dr. Amaury Antônio de Castro Jr. (orientador)
Profa. Dra. Elisângela Silva da Cunha Rodrigues (coorientadora)

Área de Concentração: Sistemas Embarcados e Robótica

Análise de Imagens: Método Alternativo
de Mensuração do Comprimento de
Plântulas para Testes de Vigor

Campo Grande, 20 de janeiro de 2016.

Resumo

Para verificar se uma semente é de boa qualidade fisiológica, destacam-se os testes de vigor. Um dos testes de vigor mais simples e utilizados por pesquisadores da área é o de comprimento de plântulas, geralmente realizado manualmente. As sementes são mantidas em um germinador, durante 7 dias e, depois, as plântulas são medidas com uma régua. Dessa forma, o processo torna-se trabalhoso e custoso para grandes lotes de sementes. Nesse contexto, este trabalho apresenta um método alternativo e automatizado, que faz uso de técnicas de processamento e suavização de imagens, bem como de um método de extração de contornos. Além das técnicas de processamento de imagens utilizadas, foi desenvolvido o protótipo de um dispositivo para captura de imagens das sementes germinadas. As imagens obtidas são tratadas e analisadas pelo software, através das técnicas implementadas de processamento de imagens como análise de histograma em tons de cinza, limiarização, filtro de média e comprimento da imagem através da contagem dos pixels, para obtenção do comprimento da plântula. Além da aplicação de técnicas de processamento de imagens em problemas da agricultura, o método proposto é diferenciado, pois torna o processo de aferição mais rápido, preciso e menos trabalhoso, quando comparado ao método manual.

Palavras-chaves: sementes, limiarização, processamento digital de imagens.

Abstract

Vigor tests are used to verification of seed physiological quality. One of the most common and simple vigor tests is the seedling length test, usually performed manually. The seeds are kept in a germination chamber for 7 days and then the seedlings are measured with a ruler. Thus, the process becomes laborious for large batches of seeds. This work presents as an alternative, an automated method, which makes use of processing techniques and smoothing images, and a contour extraction method. In addition to image processing techniques has been developed a prototype device to capture images of germinated seeds. The images are processed and analyzed by the software, through the techniques of image processing such as histogram analysis grayscale, thresholding, median filter and length of the image by counting pixels to obtain the length of the seedling. Besides the application of agricultural problems in image processing techniques, the proposed method and differential because it makes the fastest measurement process, precise and less labor compared to the manual method.

Keywords: seeds, thresholding, digital image processing.

Sumário

Lista de Figuras	vi
Lista de Tabelas	viii
1 Introdução e Motivação	2
1.1 Considerações Iniciais	2
1.2 Definição do Problema e Justificativa	3
1.3 Objetivos e Organização do Texto	4
2 Testes de Vigor de Sementes	5
2.1 Germinação e Vigor	5
2.2 Testes de Vigor Diretos e Indiretos	6
2.3 Tipos de Testes de Vigor	6
2.3.1 Envelhecimento Acelerado	8
2.3.2 Condutividade Elétrica	9
2.3.3 Frio	10
2.3.4 Comprimento de Plântulas	11
2.3.5 Massa Seca	12
2.4 Descrição, vantagens e desvantagens do teste de Comprimento de Plântulas .	13
3 Processamento de Imagens	18
3.1 Fundamentos	18
3.1.1 Processamento de imagens	19
3.1.2 Conceito de imagem digital	21
3.1.3 Aquisição/digitalização e Restauração/Realce	22
3.1.4 Segmentação	28

3.1.5	Extração de atributos	29
3.2	Aplicações de Processamento de Imagens na Agricultura	29
4	Experimentos	32
4.1	Fase 1 - A ideia inicial e o programa desenvolvido na linguagem C	33
4.2	Fase 2 e 3 - Programa desenvolvido em Java	38
4.2.1	Fase 2 - Usando os <i>frameworks ImageJ</i> e <i>Swing</i>	39
4.2.2	Fase 3 - A opção calibrar e como foi realizada a medição em centímetros.	43
5	Conclusão	46
5.1	Dificuldades Encontradas	46
5.2	Resultados e Perspectivas	47
A	Código fonte do programa de análise por limiarização alterado	51
B	Arquivo de informações das plântulas geradas pelo sistema	64
C	Principais métodos da classe Processar Imagem	66

Lista de Figuras

2.1	Câmara de Envelhecimento	8
2.2	Condutividade Elétrica	9
2.3	Teste de Frio	10
2.4	Comprimento de Plântulas	11
2.5	Massa Seca	12
2.6	Germinador de Sementes.	13
2.7	a) Folhas de papel sobrepostas na bancada b) Dobra feita no papel para ajudar na fixação das folhas.	14
2.8	a) A primeira folha aberta. b) As folhas 2 e 3 ensopadas para receber as sementes.	14
2.9	a) Sementes colocadas no terço superior da folha b)Sementes colocadas em ziguezague em todo o papel. c) Folha dobrada sobre as sementes.	15
2.10	a) A primeira folha molhada até duas vezes e meia o seu peso. b) Enrolando as 3 folhas junto com as sementes.	15
2.11	a) O rolo dos papeis com as sementes. b) Rolos de papel colocados no germinador.	15
2.12	Sementes depois de 7 dias no germinador.	16
3.1	Etapas de um sistema de VC genérico. Fonte: CONCI et all, 2008 [1]	20
3.2	Pixel da imagem. Fonte:12G turma, 2014 [2]	22
3.3	Resolução da imagem. Fonte:Falcão, 2014 [3]	22
3.4	Profundidade da imagem	23
3.5	Imagem em tons de cinza e binária respectivamente. Fonte: CONCI et all, 2008 [1]	24
3.6	Histograma de uma imagem digital. Fonte: CONCI et all, 2008 [1]	25
3.7	Efeito de compressão de histograma. Fonte: CONCI et all, 2008 [1]	26

3.8	Efeito de expansão de histograma. Fonte: CONCI et all, 2008 [1]	26
3.9	Kernels	28
3.10	Planta atacada pela lagarta do cartucho. Fonte: [4]	30
3.11	Planta atacada pela lagarta elasmio. Fonte:VARELLA, 2006 [4]	31
3.12	Planta com traços de planta invasora. Fonte:VARELLA, 2006 [4]	31
4.1	Fases do desenvolvimento.	32
4.2	Caixa de Isopor.	34
4.3	Webcam na parte superior da caixa de isopor.	34
4.4	Programa original de segmentação por limiarização, Pistori 2015 [5]	35
4.5	Programa alterado e processando uma imagem de soja.	35
4.6	Programa alterado com imagens processadas.	36
4.7	Imagem original com fundo azul.	37
4.8	Programa com todas as imagens: original, com fundo azul e tentando separar as plântulas.	37
4.9	Imagem original das plântulas.	41
4.10	Imagem em tons de cinza.	41
4.11	Imagem que foi retirada o fundo através da análise do histograma.	42
4.12	Imagem suavizada e transformada para preto e branco.	42
4.13	Primeira plântula separada.	43
4.14	Segunda e terceira plântula separada.	43
4.15	Quarta e quinta plântula separada.	44
4.16	Opção do sistema para calibrar o histograma.	44
4.17	Opção do sistema para calibrar a cor vermelha da legenda.	45

Lista de Tabelas

2.1	Tabela descritiva dos testes de vigor.	7
2.2	Tabela com os Resultados da Germinação Realizada por 7 Dias.	16
4.1	Tabela descritiva das fases do programa.	33
4.2	Tabela dos métodos criados para o processamento da imagem.	40

Agradecimentos

Ao meu orientador Professor Dr. Amaury Antônio de Castro Junior, que não somente orientou, mas ajudou-me em todas as etapas deste trabalho, sendo mestre compreensivo e competente.

À minha co-orientadora Professora Dra. Elisângela Silva da Cunha Rodrigues pela sua contribuição no texto e nas ideias sobre o desenvolvimento do sistema.

À UFMS, que muito me acrescentou através do corpo docente e pela oportunidade de cursar o mestrado.

À minha esposa Olga Barczyszczyn Vilhanueva que me apoiou em todo o processo do mestrado, nunca deixando que eu desistisse do término desta árdua tarefa.

Ao professor Dr. Matheus Bornelli de Castro que foi um dos inspiradores desta ideia e me auxiliou na parte de agricultura e também ao professor Dr. Wesley Nunes Gonçalves pelo conhecimento na área de processamento de imagens.

Aos professores, mestre Eli Castanho e doutora Fabrícia Carla Viviane, pela ajuda na finalização do texto.

Aos meus pais Heitor Vilhanueva e Neucy Pinheiro pelo apoio e a ajuda durante a realização deste trabalho.

Dedicatória

*“ Ao criador do universo,
a quem tudo pertence e
nada foge ao seu domínio.
Em cujas mãos estou apoiado,
dedico esta obra. ”*

Capítulo 1

Introdução e Motivação

Esta capítulo faz uma explanação da história da semente, explicando o que é germinação e vigor. Ele termina com a proposta deste trabalho e seus objetivos.

1.1 Considerações Iniciais

Há cerca de 10 mil anos o homem verificou que a semente, quando plantada em condições adequadas, dá origem a uma planta igual àquela que a formou e que esta se multiplica dezenas ou até centenas de vezes em relação à semente original. Neste cenário, as sementes passaram a ser material de grande importância para a tranquilidade e a prosperidade dos povos. Durante muito tempo, a produção de sementes seguiu trajetória bastante irregular, alternando fases de avanços e recuos, coincidentes com maior e menor progresso da humanidade. Com a revolução industrial, a produção de sementes evoluiu, pois a crescente mecanização dos meios de produção e de transportes resultava em maior concentração urbana, no aparecimento de novas cidades, numa melhor e mais rápida colonização dos novos continentes, numa ampliação do período médio da vida humana, e tudo isso, consequentemente, na necessidade de maior produção de alimentos e de matéria-prima para as indústrias (CARVALHO e NAKAGAWA, 2012) [6].

A produção de sementes também teve seus problemas, como as fraudes que começaram a existir. Para tentar evitar este tipo de abuso, em 1816, em Berna, na Suíça, surgiu o primeiro decreto proibindo a venda de sementes de trevo adulteradas. Mas isto não foi suficiente para resolver os problemas. Percebeu-se que era necessário um órgão ou instituição que verificasse, mediante critérios padronizados, se a lei em questão estava sendo observada. Esse órgão era um laboratório no qual sementes eram analisadas para verificar se elas se enquadravam nos padrões de qualidade impostas pela lei (CARVALHO e NAKAGAWA, 2012) [6].

Com essa finalidade, surgiu em 1869, em Tharandt, na Saxônia, Alemanha, o primeiro laboratório de sementes do mundo. Depois deste surgiram outros, não só na Alemanha como em outros países da Europa. Nobbe criou regras para analisar sementes. Mas somente em 1897 surgiu um manual sobre o assunto, do norte-americano chamado Jenkins, que estabeleceu regras que são observadas até hoje (CARVALHO e NAKAGAWA, 2012) [6].

A qualidade da semente pode ser conceituada como o somatório dos atributos genéticos, físicos, fisiológicos e sanitários que influenciam a capacidade de originar plantas com maior produtividade. A alta qualidade da semente reflete diretamente no resultado final da cultura, em termos de ausência de moléstias transmitidas pela semente, do alto vigor das plantas, da maior produtividade e de uniformidade da população (CASTRO, 2011) [7].

A avaliação da qualidade fisiológica das sementes é realizada pelo teste de germinação, que consiste em colocar as sementes em um substrato com boas condições de umidade e temperatura para a ocorrência da emergência e desenvolvimento das estruturas do embrião. Porém, esse teste tem limitações por fornecer resultados que superestimam o potencial fisiológico das sementes, devido ao fato de ser conduzido sob condições ideais e artificiais. As condições adversas como umidade do solo, clima, competição, entre outras, podem impor uma desuniformidade entre o teste de germinação e os resultados de campo. Portanto, embora seja o único obrigatório por lei, esse teste aplicado isoladamente, muitas vezes não é eficiente para prever o comportamento das sementes no campo.

Para complementar as informações do teste de germinação criou-se o conceito de vigor. A *Association of Official Seed Analysts* (AOSA, 1983) [8], definiu o vigor de sementes como aquelas propriedades que determinam o potencial para uma emergência rápida e uniforme e para o desenvolvimento de plântulas ¹ normais, sob uma ampla faixa de condições ambientais. Vários testes de vigor foram desenvolvidos procurando precisar o comportamento de lotes de sementes, em campo, com dados obtidos em laboratório.

Esses testes são classificados em métodos diretos e métodos indiretos. Os diretos são os métodos que procuram simular as condições (às vezes adversas) que ocorrem no campo e o outro procura avaliar atributos que indiretamente se relacionam com vigor (físicos, biológicos, fisiológicos). NAKAGAWA, 2000). Os testes indiretos se dividem em fisiológicos e bioquímicos. Dentre estes, o que importa para este trabalho são os fisiológicos, que avaliam aspectos da germinação ou crescimento da plântula.

1.2 Definição do Problema e Justificativa

O milho e a soja estão entre os cereais mais cultivados do mundo. Seus grãos de elevado valor nutritivo são muito empregados na alimentação humana e animal. No Brasil, a produção de milho na safra 2015 foi de 78 milhões de toneladas (Notícias Agrícolas, 2016 [9]). A soja, por sua vez, apresentou produção de 94,5 milhões de toneladas, na safra 2015 (Notícias Agrícolas, 2016 [9]). Ambas as culturas estão em plena expansão, sendo que a soja cresceu cerca de 10% em área plantada em relação ao ano de 2012 e o milho aproximadamente 18%. Estes resultados atraem o interesse de grandes empresas do setor agroindustrial, motivadas a investir cada vez mais em sementes de qualidade, o que aumenta a produtividade. A cadeia de produção de sementes de milho e soja são complexas e a coordenação eficaz de todas as atividades envolvidas garante a competitividade da empresa e o fornecimento de sementes no momento certo, com qualidade satisfatória e mínimo custo JUNQUEIRA e MORABITO, 2006 [10].

¹Plântula é o embrião vegetal já desenvolvido e ainda encerrado na semente

Os métodos de avaliação da qualidade de sementes precisam ser rápidos, eficientes e facilmente aplicáveis. Neste contexto, os testes de vigor têm grande importância, pois são muito úteis nas etapas de um programa de produção, como avaliação do potencial fisiológico de lotes com germinação semelhante, seleção de lotes para a semeadura com base no potencial de emergência das plântulas em campo, avaliação do potencial de armazenamento, avaliação do grau de deterioração, controle de qualidade pós-maturidade, avaliação da qualidade fisiológica e auxílio em métodos de seleção durante o melhoramento de plantas e avaliação de efeitos de injúrias mecânicas e térmicas, tratamento fungicida e de outros fatores adversos pré e pós-colheita.

Existem vários testes de vigor que podem ser feitos com a semente como: velocidade de germinação, primeira contagem do teste de germinação, comprimento da plântula, peso da massa seca da plântula e classificação do vigor das plântulas, que são realizados em condições laboratoriais e percentagem de emergência de plântulas, velocidade de emergência de plântulas, altura de plântula e peso da massa fresca da plântula, que são realizados em campo. Um dos testes indiretos que tem grande vantagem em relação aos demais, por ser relativamente rápido, com boa precisão e com custo muito reduzido, é a medida do crescimento de plântulas, que é realizado com uma régua. Esse teste torna-se trabalhoso quando realizado com uma grande quantidade de sementes, pois exige trabalho manual extenso.

Uma alternativa para o teste de vigor é a análise de imagem da semente com a ajuda de um software que separa as plântulas e mede seu tamanho analisando a imagem.

1.3 Objetivos e Organização do Texto

Neste trabalho é proposto o desenvolvimento de um produto composto por *software* e *hardware* (*middleware*), para a análise do vigor das sementes através do comprimento da plântula. Na pesquisa sobre o crescimento de plântulas, as sementes de milho e soja foram escolhidas por terem um crescimento mais uniforme e retilíneo, além da sua importância econômica e agrônômica. A análise da imagem destas sementes, por este método, torna-se mais rápida e passível de implementação através de um *software*, com o tratamento da imagem e extração do comprimento da plântula. O *hardware* é composto de uma estrutura para acomodar uma folha de papel tamanho A4 com as sementes e uma webcam para capturar a imagem.

A organização do trabalho será descrita à seguir. No Capítulo 2, serão apresentados os conceitos de teste de vigor direto e indireto mais a fundo, apresentados os conceitos de vigor de sementes, descritos os tipos de testes de vigor que existem e o teste de vigor baseado no comprimento de plântulas, que será o foco desse trabalho. Também é mostrado como é realizado o teste de vigor através do comprimento das plântulas, descrito como foram realizados os primeiros testes com o comprimento de plântulas e explicados o *hardware* e o *software* que foram criados.

No Capítulo 3 é realizado um embasamento teórico sobre o processamento de imagens e suas aplicações na agricultura. Finalmente no Capítulo 4, são descritos os experimentos e os resultados práticos.

Capítulo 2

Testes de Vigor de Sementes

Este capítulo trata especificamente do vigor de sementes, os testes diretos e indiretos e os tipos de testes de vigor que podem ser realizados e os mais usados. Os testes mais usados são explicados detalhadamente.

2.1 Germinação e Vigor

A agricultura emprega diversas formas de tecnologia, pode-se perceber isto, por exemplo nas máquinas modernas utilizadas em diferentes etapas da produção, nos esforços para aumento de produtividade, no desenvolvimento de culturas com resistência a doenças e de plantas adaptadas ao clima local através da engenharia genética, entre outros.

Neste contexto, surge a necessidade de sementes que devem apresentar elevada qualidade genética, física, fisiológica e sanitária que determinam o seu valor para semeadura. Destaca-se o aspecto fisiológico da qualidade das sementes, diretamente responsável pelo seu desempenho em campo e armazenamento dos grãos. Segundo MENEZES, 2014 [11], o método rotineiro para determinar a qualidade das sementes (muitas vezes o único) é o teste de germinação, que, embora muito útil, não informa sobre o vigor, longevidade e emergência em campo. Além disso, necessita um prazo de 7 a 28 dias para fornecer os resultados, período considerado longo para atender aos interesses comerciais dos produtores de sementes.

A germinação das sementes no campo sofre a influência de fatores ambientais variados, como seca, erosão, frio entre outros. Nos testes de germinação as sementes ficam em condições ideais o que pode distanciar os resultados dos obtidos no campo. Segundo COSTA, 2013 [12], uma vez que o teste de germinação não tem apresentado resultados consistentes que se correlacionem positivamente com o desempenho das sementes no campo, principalmente sob amplas condições ambientais, surgiu o conceito de vigor entre aqueles envolvidos com a ciência e tecnologia de sementes. A avaliação do vigor representa, atualmente, uma ferramenta fundamental em programas de produção de sementes, tanto do ponto de vista dos produtores, permitindo o monitoramento da qualidade da produção, quanto do ponto de vista dos consumidores (agricultores), como elemento norteador na tomada de decisões.

Segundo PERES, 2010 [13], historicamente os testes de vigor tiveram início com o desenvolvimento do teste padrão de germinação, e em 1869, Nobbe estabeleceu o primeiro

laboratório de análise de sementes em Tharandt/Alemanha; em 1876, ocorreu o mesmo em Connecticut/EUA. Durante vários anos, a avaliação da qualidade fisiológica foi efetuada somente através do teste padrão de germinação; apenas a partir de 1940 o teste de tetrazólio¹ foi desenvolvido por Lakon, na Alemanha. De acordo com o relato de vários pesquisadores, a introdução do termo vigor foi, primeiramente, atribuída a Nobbe, em 1876, que utilizou a palavra “*triekraft*” que significa força motriz ao discorrer sobre o processo de germinação. Em 1925, Fick e Hibbard fizeram os testes de condutividade elétrica. Em 1931 o teste de velocidade de germinação por Stahl, 1930 e 1935 o teste de frio por Reddy e aprimorado por Tatum e Zuber entre 1942 e 1943. Em 1950, Franck propôs que os testes relacionados à percentagem de emergência das plântulas em solo deveriam ser chamados testes de vigor. Segundo o ISTA o vigor é a soma de todas as propriedades da semente as quais determinam o nível de atividade e o seu desempenho, ou do lote de sementes durante a germinação e a emergência de plântulas. Sementes que tenham um bom desempenho são classificadas como vigorosas e as de baixo desempenho são chamadas de sementes de baixo vigor.

2.2 Testes de Vigor Diretos e Indiretos

Segundo CARVALHO e NAKAGAWA, 2012 [6], os testes de vigor são divididos em diretos e indiretos. Como diretos classificam-se os testes que, de certo modo, imitam as condições, as vezes adversas do campo, onde as sementes mostram sua habilidade de emergir sob condições simuladas de estresse. Os indiretos procuram avaliar atributos que se relacionam com o vigor (físicos, biológicos, fisiológicos) das sementes, isto é, aqueles que medem um componente fisiológico específico da semente indiretamente relacionado com seu desempenho germinativo, por exemplo, o teste de condutibilidade elétrica. Segundo OLIVEIRA, 2009 [14], os testes diretos, por serem realizados tanto no laboratório como no campo, possuem a dificuldade de se padronizar, devido à variação das condições climáticas; Por isso, deve-se utilizar cautela quando houver comparação entre lotes de sementes. Outra dificuldade encontrada é a variação de interpretação dos resultados dos testes, por estes serem realizados por laboratórios e/ou analistas diferentes e sua avaliação ser muitas vezes subjetiva. Por isso, atualmente há trabalhos que utilizam análise computadorizada de imagens para ter-se uma maior consistência dos resultados, como por exemplo, TEIXEIRA, 2004 [15] e LIMA e JORGE, 2014 [16].

2.3 Tipos de Testes de Vigor

Segundo VIEIRA, 2011 [17], as características de um teste de vigor devem ser: simplicidade, rapidez, baixo custo, objetivo, reproduzível, e com resultados relacionados com a emergência das plântulas em campo. Segundo CARVALHO e NAKAGAWA, 2012 [6], apesar de diversos estudos que buscam a padronização dos testes de vigor, são encontradas certas dificuldades em função de que o vigor pode ser refletido através de várias características, como velocidade de germinação, uniformidade de emergência, resistência ao frio, temperatura

¹O teste de tetrazólio baseia-se na atividade das enzimas desidrogenases, as quais catalizam as reações respiratórias nas mitocôndrias durante a glicólise e o ciclo de Krebs. Estas enzimas, particularmente a desidrogenase do ácido málico, reduzem o sal de tetrazólio nos tecidos vivos.

e umidade elevadas, substâncias tóxicas, entre outros. Diante disto, deve-se ressaltar a importância da realização de um conjunto de testes que responda a estas características.

MCDONALD, 1975 [18] classificou e incluiu novos métodos de análise de sementes, como testes físicos, fisiológicos, bioquímicos e de resistência. Os testes físicos avaliam aspectos morfológicos ou características físicas das sementes possivelmente associadas ao vigor. Exemplos destes são: tamanho das sementes, peso unitário, densidade, coloração e testes de raios x.

Os testes fisiológicos procuram determinar atividades fisiológicas específicas, cuja manifestação depende do vigor. Exemplos destes são: classificação do vigor das plântulas, primeira contagem de germinação, velocidade de germinação ou de emergência de plântulas, transferência de matéria seca, teste de exaustão e crescimento das plântulas.

Os bioquímicos avaliam alterações bioquímicas associadas ao vigor das sementes. Exemplos destes são: testes de respiração, teste da atividade da descarboxilase do ácido glutâmico (ADAG), teste de tetrazólio, teste de condutividade elétrica, teste de lixiviação de potássio, teste dos aldeídos voláteis e teste dos ácidos graxos livres.

Testes de resistência avaliam o desempenho das sementes expostas a estresse. Exemplos destes são: germinação a baixas temperaturas, imersão em água quente, teste de submersão, imersão em solução osmótica, imersão em soluções tóxicas à semente, teste do tijolo moído (teste de Hiltner), envelhecimento acelerado e teste de frio.

Tabela 2.1: Tabela descritiva dos testes de vigor.

Teste de vigor	Objetivo	Tipo
Envelhecimento acelerado	Colocar sementes sob estresse de temperatura e umidade	Resistência
Condutividade elétrica	Avaliar a quantidade de eletrólitos que é liberada pelas sementes embebidas em água	Bioquímico
Frio	Testar as sementes em condições adversas de excesso de água e baixas temperaturas	Resistência
Comprimento de plântulas	Testar o crescimento das plântulas da semente	Fisiológico
Massa seca	Extraír toda a água da plântula através de um aquecimento controlado em laboratório e depois pesar a massa seca	Fisiológico

De todos os testes acima, os que são padronizados pela ISTA são: taxa de crescimento de plântulas, classificação do vigor de plântulas, envelhecimento acelerado, teste a frio, teste do tijolo moído, teste da deterioração controlada tetrazólio, condutividade elétrica e tetrazólio da camada de aleurona. Os padronizados pela AOSA (1983) são: classificação do vigor de plântulas, taxa de crescimento de plântulas, envelhecimento acelerado, teste a frio, germinação à temperatura subótima tetrazólio e condutividade elétrica.

Dessa forma, vários testes de vigor têm sido desenvolvidos, aprimorados e utilizados com o objetivo de estimar a qualidade fisiológica das sementes. Na Tabela 2.1 são apresentados os testes mais utilizados.

2.3.1 Envelhecimento Acelerado

O teste de envelhecimento acelerado tem como base colocar as sementes sobre um estresse que consiste basicamente de temperaturas e umidade elevadas. Segundo VIEIRA, 2011 [17], os estudos de Crocker e Groves mostraram que a morte das sementes durante o armazenamento, era causada pela coagulação de proteínas e que o aquecimento “acelerava” este processo; estes pesquisadores sugeriram que testes de germinação conduzidos após exposição, relativamente rápida, de sementes secas a temperaturas elevadas (50-100°C), poderiam ser úteis para prever sua longevidade.

O teste de envelhecimento simula um estresse que pode ser reproduzido em câmaras que são geralmente de aço inoxidável com prateleiras perfuradas e móveis (Figura 2.1). Em seu interior uma resistência elétrica ligada a um termostato, com uma câmara interna que pode receber 3 litros de água. As amostras de sementes podem conter de 200 a 250 sementes que permanecem em um ambiente úmido com uma temperatura entre 40 e 45 °C. A umidade inicial da semente influencia no resultado. Depois disto a semente é colocada para germinar e é verificado o seu vigor.



Figura 2.1: Câmara de Envelhecimento

Segundo DUTRA e VIEIRA, 2004 [19], o teste de envelhecimento acelerado é um dos mais utilizados no Brasil e no mundo, particularmente para sementes de milho e soja. Vários estudos têm sido conduzidos visando o aperfeiçoamento deste procedimento como teste de vigor. Sabe-se, também, que estes estudos avaliam, principalmente, o efeito de fatores que afetam os resultados, como teor de água da semente, tamanho da amostra e tipo de câmara usada no envelhecimento.

De acordo com BERTOLIN, 2011 [20], o teste de envelhecimento acelerado integra muitas das importantes características desejadas em um teste de vigor, é rápido, econômico, simples e útil para todas as espécies. Inicialmente proposto como um método para avaliar o potencial de armazenamento de sementes, este teste é realizado em condições de alta temperatura e umidade relativa do ar (cerca de 100%) por períodos curtos (3 a 4 dias), seguido por um teste de germinação. As diferenças de umidade inicial das sementes devem ser consideradas na interpretação deste teste. Logo, sementes mais úmidas sofrerão maiores efeitos deletérios do estresse, devido à intensificação da sua atividade metabólica.

2.3.2 Condutividade Elétrica

De acordo com CUSTODIO, 2005 [21], pesquisas com diferentes espécies têm mostrado que o decréscimo na germinação e no vigor é diretamente proporcional ao aumento da lixiviação de solutos, indicando que a condutividade elétrica é um método eficiente para a avaliação do vigor. Alguns fatores também podem afetar os resultados destes testes, como idade da semente, genótipo, qualidade da água, temperatura e duração do período de embebição, grau de umidade e número de sementes.

O teste de condutividade elétrica consiste em avaliar a quantidade de eletrólitos que é liberada pelas sementes embebidas em água (Figura 2.2). Este processo está relacionado à perda da integridade dos sistemas de membranas celulares, que causam a deterioração das sementes, fazendo com que percam seu vigor. O tempo de embebição muda de acordo com o tamanho das sementes e pode variar de 2 a 24 horas em sementes maiores, como as de soja. Por exemplo BROUWER e MULDER, 1982 [22], trabalhando com feijão encontraram resultados consistentes com 4 horas de embebição que se assemelharam aos resultados obtidos com 24 horas, concluindo que este seria um método mais rápido para detectar a baixa qualidade de lotes de sementes. FILHO, 1990 [23] concluiu que períodos mais curtos (4 e 8 horas) podem ser utilizados para a identificação de diferenças mais acentuadas de vigor em sementes de soja, enquanto a embebição mais prolongada torna esse teste mais sensível às diferenças de vigor.



Figura 2.2: Condutividade Elétrica

2.3.3 Frio

O teste de frio deve ser realizado sob condições adversas de excesso de água e baixas temperaturas, porque este tipo de ambiente retarda a germinação e favorece o desenvolvimento de microrganismos patogênicos. Este teste consiste em colocar porções de areia e terra em uma caixa e acrescentar água até que ela chegue a 60% de umidade, depois colocar a caixa a uma determinada temperatura por alguns dias e ir alternando a temperatura entre 5 e 7 dias que pode ser de 10 a 17 °C (Figura 2.3). Os testes podem ser feitos com terra e sem terra.

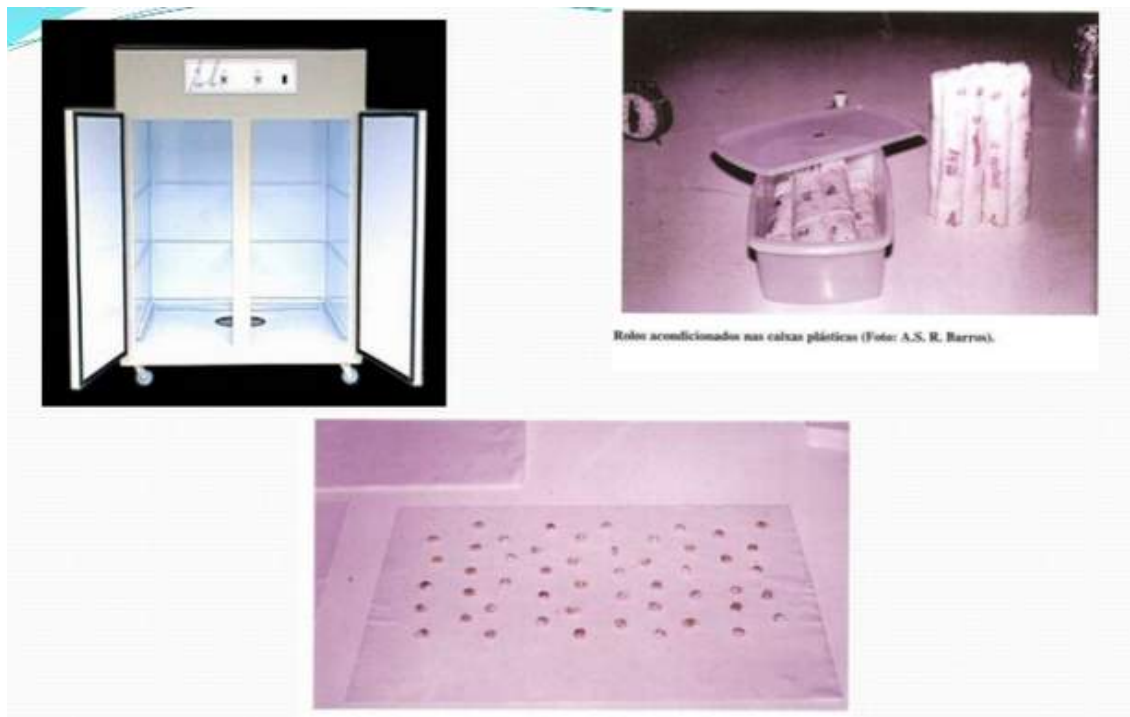


Figura 2.3: Teste de Frio

A eficiência do teste de frio foi comprovada experimentalmente por diversos pesquisadores e foi sugerido seu uso na seleção de linhagens de milho antes do período de semeadura, devido a sua sensibilidade em prognosticar o desempenho das sementes em campo. Muitas empresas produtoras de sementes de milho utilizam esse teste nos programas internos de controle de qualidade. Através do conhecimento das condições desfavoráveis de campo, é possível às empresas a manutenção de um padrão rígido de comercialização das sementes através do uso do teste de frio, para predizer a emergência de plântulas sob condições de frio e alta umidade. Esse teste tem sido utilizado com sucesso, também, para diferenciar níveis de vigor relacionados com a forma, peso, tamanho e tratamento de sementes. No entanto, é importante ressaltar que seria necessária a padronização da metodologia do teste de frio, de acordo com as características de cada espécie a ser testada.

2.3.4 Comprimento de Plântulas

Este teste tem como base o crescimento das semente, assim as que crescem mais, são consideradas mais vigorosas, isto porque as mais vigorosas originam plântulas com maior taxa de crescimento, em função da maior translocação das reservas dos tecidos de armazenamento para o crescimento do eixo embrionário. As plântulas depois de germinadas são medidas com uma régua e suas medidas expressas em centímetros (Figura 2.4).

A determinação do comprimento médio das plântulas normais é realizada, tendo em vista que as amostras que expressam os maiores valores são mais vigorosas.

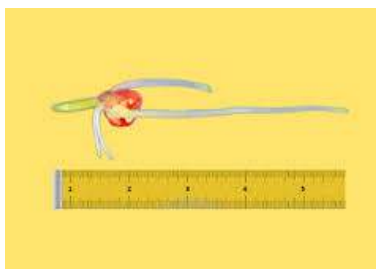


Figura 2.4: Comprimento de Plântulas

Os testes que avaliam o comprimento de plântulas são recomendados pela AOSA e ISTA. Atualmente, os referidos testes estão sendo bastante empregados nos laboratórios de análise de sementes, por apresentarem as vantagens de um baixo custo, de serem relativamente rápidos, não necessitarem de equipamentos especiais nem demandar treinamento específico sobre a técnica empregada. A metodologia básica para a obtenção do comprimento de plântula é a utilização de dez a vinte sementes colocadas em uma ou duas fileiras, no substrato por repetição, sendo utilizadas, geralmente, quatro repetições. O procedimento pode ser realizado em rolos de papel ou em caixas próprias para a germinação. Se for feito nestas caixas, ela deve ser mantida a um ângulo de 45° ou mais com a bandeja do germinador.

Depois de preparadas, as sementes são colocadas na câmara de germinação, na temperatura recomendada para cada espécie a ser testada. Nas gramíneas, mede-se o comprimento total da plântula ou apenas o comprimento da raiz principal. Nas leguminosas, toma-se a medida da extremidade da raiz até a inserção dos cotilédones, ou parte da plântula (raiz primária, hipocótilo, epicótilo). A escolha da estrutura adequada para avaliação é importante para ter-se resultados consistentes e comparáveis.

Segundo NAKAGAWA, 1999 [24], o comprimento médio da plântula ou da sua(s) parte(s) eleita(s) é obtido somando as medidas tomadas de cada plântula normal, em cada repetição ou subamostra, e dividindo, a seguir, pelo número de plântulas normais mensuradas. Os resultados são expressos em mm ou em cm, com uma casa decimal. A temperatura deste teste deve ser bem controlada, já que é um teste minucioso e pequenas diferenças podem ser consideráveis no momento da avaliação.

O escalonamento das amostras colocadas no germinador deve ser de intervalo de tempo pequeno entre a primeira e a última amostra. Outro cuidado a ser tomado é o da fixação da semente no lugar adequado, quando é utilizada a caixa para germinação a 45°. Para sementes

mais arredondadas e menores e sementes peletizadas ²pode-se fazer uso de um papel tipo plissado, para que assim permaneçam no lugar correto.

2.3.5 Massa Seca

Este teste de vigor consiste em extrair toda a água da plântula através de um aquecimento controlado em laboratório e depois pesar a massa que restou. Primeiro é pesado o material antes de secar e depois que este for submetido à secagem é pesado novamente, fazendo uma média de todas as plântulas usadas (Figura 2.5).



Figura 2.5: Massa Seca

Para esta determinação, as amostras que apresentam maiores pesos médios de matéria seca de plântulas normais são consideradas mais vigorosas. As sementes vigorosas proporcionam maior transferência de massa seca de seus tecidos de reserva para o eixo embrionário, na fase de germinação, originando plântulas com maior peso, em função do maior acúmulo de matéria. A instalação deste teste pode tanto ser realizada da mesma forma que a do teste anterior, como no campo, sulcos ou em bandejas, seguindo a metodologia utilizada para o teste de percentagem de emergência de plântulas. Além disso, o peso da matéria seca da plântula pode ser obtido após a determinação do peso da massa fresca da plântula. Para obtenção do peso da matéria seca da plântula, são retiradas as plântulas normais do substrato e removidos os cotilédones, restos de sementes e/ou qualquer tipo de reserva, obtendo apenas as plântulas, que são colocadas em estufa a 80°C por 24 horas. Logo após, as plântulas são colocadas para resfriar em dessecador e pesadas em uma balança de precisão 0,001g ou 0,0001g. Depois de determinado o peso da matéria seca total, divide-se pelo número de plântulas para se obter o peso médio da matéria seca mg/plântula.

²Peletização é a aplicação de um revestimento rígido e seco que visa modificar individualmente o formato e o tamanho das sementes, tornando-as esféricas. Este processo tem a finalidade de facilitar o manuseio de sementes ou permitir a utilização de semeadoras mecânicas de precisão, reduzindo grandemente o consumo de semente e o gasto com mão-de-obra para o desbaste das plântulas excedentes.

2.4 Descrição, vantagens e desvantagens do teste de Comprimento de Plântulas

As vantagens deste teste são: baixo custo; não necessita de equipamentos especiais; não demanda treinamento adicional específico sobre a técnica empregada e relativamente rápido. As desvantagens ou a inviabilização deste teste ocorre quando há um grande número de lotes, que pode viciar os resultados pela repetição dos mesmos movimentos quando são realizadas as medições através de uma régua. Para entender e detalhar esta técnica, foram realizados experimentos com um pesquisador da área.

A realização deste teste consiste em 8 repetições de 10 a 12 sementes que são colocadas em rolos de papel para germinação. Para o preparo do rolo são usadas três folhas, duas sob as sementes e uma cobrindo-as. Elas deverão ser umedecidas com água destilada com a quantidade de duas vezes e meia o seu peso. Por exemplo, se o papel pesa 10 gramas, deverão ser colocados 25 ml de água.

O local onde serão colocados os papeis durante a montagem, que pode ser uma bancada, deve ser limpo com álcool 70%, para evitar a contaminação por fungos. As sementes são dispostas em linhas no terço superior da folha em ziguezague para que na hora do crescimento elas não causem interferência umas nas outras. A micrópila da semente (pequenos orifícios no tegumento dos óvulos dos vegetais fanerógamos ³, cuja finalidade é permitir a fecundação ou a alimentação) deve ser posicionada para a parte inferior do papel. Estes papeis devem ser enrolados e posicionados verticalmente no germinador (Figura 2.6) a uma temperatura de 25°C e alta umidade, onde permanecerão por 7 dias (no caso de sementes de milho e soja).



Figura 2.6: Germinador de Sementes.

³Fanerógamas são as plantas que produzem sementes, flores e frutos, onde os óvulos e o pólen são os gametas feminino e masculino, ou seja, as estruturas para reprodução são claramente visíveis.

2.4. Descrição, vantagens e desvantagens do teste de Comprimento de Plântulas FACOM-UFMS

Para fazer o teste de vigor, podem ser usadas sementes que foram mantidas a 10°C. Se as sementes não estiverem na temperatura ambiente devem ser deixadas fora da refrigeração até que fique com esta temperatura. São separadas 3 folhas de papel de germinação e pesadas, depois elas são colocadas uma sobre a outra em cima de uma bancada esterilizada (Figura 2.7 a). Logo após elas são dobradas (Figura 2.7 b) para que fique mais fácil virar a primeira folha (Figura 3.3 a). O próximo passo é molhar até duas vezes e meia o seu peso (Figura 2.8 b) e a partir disto as sementes são colocadas no terço superior da segunda folha em ziguezague (Figuras 2.9 a e 2.9 b). Em seguida, a folha é virada (Figura 2.9 c) em cima das sementes e é derramada água até que fique até duas vezes e meia o seu peso (Figura 2.10 a), depois elas são enroladas juntamente com a semente no lado oposto da dobra feita no início. Quando as folhas são enroladas deve se tomar o cuidado de não apertar (Figura 2.10), mas somente enrolar até formar um tubo. Logo após isto, o rolo deve ser colocado em pé (Figura 3.6 a e b) no germinador a 25°C, com as sementes para cima, onde permanecerá por 7 dias.



Figura 2.7: a) Folhas de papel sobrepostas na bancada b) Dobra feita no papel para ajudar na fixação das folhas.

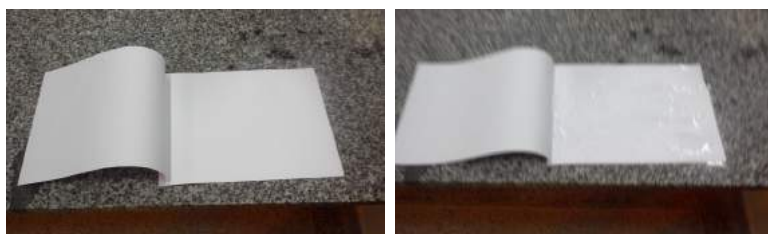


Figura 2.8: a) A primeira folha aberta. b) As folhas 2 e 3 ensopadas para receber as sementes.

No final deste período o papel é aberto e verifica-se como as sementes germinaram. Conforme a Figura 2.12, no teste realizado, podemos ver que a germinação não é igual para todas as sementes. Em alguns casos as raízes saíram do papel e tiveram várias radículas ao longo da raiz principal, indicando um bom vigor. Outras tiveram a germinação bem simples. Pela parte aérea também é possível verificar o vigor das sementes, porque as que tiveram a raiz pouco desenvolvida, provavelmente terão o mesmo resultado na parte aérea.

Uma das características que dificultará a análise das imagens são as raízes secundárias, que tem papel importante na definição do vigor das plântulas, mas não foi dada a importância devida pela falta de um experimento real, onde constata-se que estas raízes podem mudar o resultado de uma análise. Um exemplo é uma semente que tem a raiz principal de 15 cm e nenhuma raiz secundária de tamanho considerável, e outra semente que tem 12 cm de raiz principal, mas com raízes secundárias de 3 ou 4 cm cada uma. Analisada somente a raiz



Figura 2.9: a) Sementes colocadas no terço superior da folha b) Sementes colocadas em ziguezague em todo o papel. c) Folha dobrada sobre as sementes.



Figura 2.10: a) A primeira folha molhada até duas vezes e meia o seu peso. b) Enrolando as 3 folhas junto com as sementes.

principal, a primeira será considerada com maior vigor, mas levando em conta as secundárias, a segunda passa a ser a de maior vigor.



Figura 2.11: a) O rolo dos papeis com as sementes. b) Rolos de papel colocados no germinador.

A medida é feita com uma régua e se dá da seguinte maneira: a raiz e a parte aérea das plântulas são medidas separadamente, que serão utilizadas posteriormente para o cálculo da média. Este será realizado somando-se todas as medidas da parte aérea e dividindo pelo



Figura 2.12: Sementes depois de 7 dias no germinador.

número de sementes normais, anormais e mortas. O mesmo procedimento é realizado com as medidas da raiz. O resultado fornecerá a média do comprimento da raiz e da parte aérea.

A Tabela 3.1 mostra os resultados obtidos em uma medida realizada pelos alunos do curso técnico de agricultura do Instituto Federal do Mato Grosso do Sul (IFMS), realizada em uma disciplina ministrada pelo prof. Matheus Bornelli de Castro.

Tabela 2.2: Tabela com os Resultados da Germinação Realizada por 7 Dias.

Semente	Tamanho em cm	
	Raiz	Parte aérea
Planta 1	16,00	9,00
Planta 2	16,00	8,00
Planta 3	20,50	2,00
Planta 4	13,50	4,50
Planta 5	16,00	8,00
Planta 6	23,00	7,00
Planta 7	21,00	9,00
Planta 8	20,50	6,50
Planta 9	20,00	4,00
Planta 10	15,00	8,50
Média	18,15	6,65

Nestes resultados, é notório observar que, na maioria das vezes, o crescimento da raiz está relacionado com o crescimento da parte aérea, isto é, a que tem maior raiz tem a maior parte aérea.

A planta 6 tem a raiz mais desenvolvida, mas não a parte aérea. Por outro lado, a planta 7 tem a parte aérea maior e a segunda maior raiz. É interessante observar isto, porque nos primeiros testes realizados, eles conduziam a um análise errada, indicando que a maior raiz tem a maior parte aérea. Mas de posse destes dados, pode-se observar que as informações do crescimento da parte aérea não indicam que a raiz é tem um desenvolvimento parecido. Os

resultados do comprimento de plântulas podem ser acompanhados do teste de massa seca, para que através de um comparativo um teste reforce o outro.

A base deste teste é que uma semente vigorosa, ao ser embebida em água, começa a reorganizar todo o sistema de membrana, volta a respirar e mobilizar reserva para o crescimento do eixo embrionário. Quanto mais vigorosa, maior será o comprimento da raiz e da parte aérea. Dessa forma, é possível comparar diferentes lotes e classificá-los de acordo com a qualidade fisiológica.

Os testes feitos de forma manual são simples de realizar e fáceis de verificar o comprimento das plântulas, mas o processo é muito repetitivo e cansativo. Nas primeiras medidas feitas, percebe-se que não há grandes dificuldades para realizar o teste, mas depois de medir uma certa quantidade de plântulas já começa a ficar cansativo, chegando a até fazer medidas não muito precisas, o que pode introduzir erros. Isto foi observado em aula prática com estudantes do IFMS.

Capítulo 3

Processamento de Imagens

Este capítulo traz um resumo de processamento e análise de imagens, suas técnicas e alguns exemplos de aplicações que existem na agricultura.

3.1 Fundamentos

De acordo com CONCI et al., 2008 [1], a computação gráfica (CG) engloba pelo menos 3 grandes áreas: a síntese de imagens (SI), o processamento de imagens (PI) e a análise de imagens (AI). A imagem digital é uma representação de uma imagem em regiões discretas, limitada através de um conjunto finito de valores inteiros que representam cada um dos seus pontos. Até a década de 1980 o maior emprego das imagens digitais eram provenientes da pesquisa espacial. Segundo GONZALEZ e WOODS, 2008 [25], uma das primeiras aplicações de técnicas para melhoria de informação visual para interpretação humana foi o melhoramento de imagens digitalizadas para jornais, enviadas por meio de cabo submarino de Londres para *New York*. A introdução do sistema de transmissão de imagens via cabo no início dos anos 20 reduziu o tempo requerido de mais de uma semana para menos de três horas no transporte de imagens através do oceano Atlântico. Atualmente, as imagens digitais encontram-se difundidas em muitas aplicações. A expansão do seu emprego se deve muito ao fato de que, grande parte das informações que o ser humano obtém do mundo, que o cerca, ocorre através de imagens. Assim foram se caracterizando as diversas áreas, subáreas e especificidades da computação gráfica. A diferença entre as áreas de CG se concentra na relação entre o uso de técnicas de inteligência artificial, dados e imagens. Se os dados são usados para a geração de imagens, a área em consideração será a síntese de imagens. Se esses dados forem resultados de informações adquiridas das imagens, a área considerada será a análise de imagens. O processamento de imagens é o ramo da computação gráfica que transforma as imagens, assim como o processamento de dados é o ramo da computação que transforma os dados. Pode-se dizer que, um sistema de processamento de imagens, elas referem-se tanto aos dados de entrada como de saída. Esse processamento pode envolver o rearranjo dos pontos ou pixels (*picture element*) da imagem alterando os valores dos seus tons.

A área de processamento de imagens inclui tópicos como diminuição de ruídos, realce de imagem, restauração de imagens e recuperação de imagens armazenadas em banco de

dados, baseada no conteúdo visual. Se houver alguma interpretação da informação, esta é deixada para o usuário. Mas a tarefa de interpretar informações da imagem também pode ser realizada através de algoritmos computacionais. Por outro lado, os algoritmos de análise de imagens tomam imagens como entradas, mas produzem outros tipos de saída. A análise de imagens (AI) consiste em encontrar parâmetros descritivos (em geral numéricos) e que representem de modo sucinto informações importantes da imagem. A AI se dedica a desenvolver teorias e métodos voltados a extração de informações úteis contidas em imagens.

Relacionada a análise de imagens, a visão computacional (VC) vem se desenvolvendo muito nos últimos anos. Esta área trata da extração de informações das imagens e da identificação e classificação de objetos presentes nas imagens. Os sistemas de VC vem sendo usados em reconhecimento de pessoas, assinaturas e objetos; inspeção de peças em linhas de montagem; orientação de movimentos de robô em indústria automatizadas etc. Os sistemas de visão computacional envolvem AI e técnicas de inteligência artificial (IA) ou de tomada de decisão, que permitem a identificação e classificação de objetos ou imagens. Com o desenvolvimento da inteligência artificial, área da ciência em que os programas de computador são desenvolvidos para simular a cognição, passou a ser possível elaborar programas de computador que possam perceber visualmente o ambiente, (CONCI et al, 2008 [1]).

O reconhecimento da importância da visão computacional se deu com a idéia de que os computadores poderiam ser usados para simular processamento complexos de percepção visual. Até então, a visão e outras formas de percepção e cognição eram entendidas como exclusivas de seres vivos, não desenvolvíveis através de máquinas. Em algumas referências, o termo visão de máquina é usado com uma alusão a esse processo, assim com o termo visão robótica é usado quando o processo é realizado por um robô.

3.1.1 Processamento de imagens

O processamento de imagens digitais envolve procedimentos que são geralmente expressos em forma de algoritmos (implementadas por software), excetuando a aquisição e exibição de imagens. A única razão para se ter um *hardware* especializado para processamento de imagens é a necessidade de velocidade ou para vencer algumas limitações fundamentais de computação (GONZALEZ e WOODS, 2008 [25]). As técnicas de processamento de imagens digitais podem ser classificadas quanto ao seu escopo ou ao seu resultado. O escopo refere-se à abrangência das técnicas de processamento. O resultado está relacionado com o que se deseja obter de cada técnica. A Figura 3.1 mostra esquematicamente um sistema de visão computacional que é composto das seguintes etapas: Aquisição/Digitalização, Restauração/-Realce, Segmentação, Extração de atributos/características, Classificação/Reconhecimento e Decisão. A primeira etapa que é a aquisição de imagens consiste na conversão de uma cena real tridimensional em para uma imagem em duas dimensões.

Para CONCI et al, 2008 [1], as imagens reais possuem um número ilimitado de cores ou tons. No processamento de imagens computacional, é necessário limitar os níveis de cores ou tons possíveis de serem atribuídas a cada pixel da imagem (graduação tonal). Isso se deve à necessidade das imagens serem representadas por um número finito de tons, em função de *hardware* e *software*. Dependendo de quantos tons serão utilizados, pode-se produzir uma imagem com mais ou menos graduação tonal. Outro fator de grande relevância é a

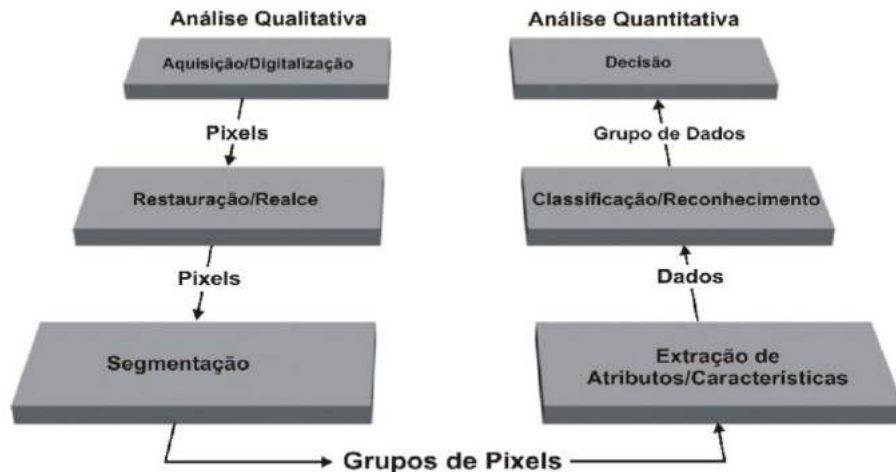


Figura 3.1: Etapas de um sistema de VC genérico. Fonte: CONCI et al, 2008 [1]

resolução ou quantidade de pontos x,y (amostragem). Quanto maior, mais detalhes podem ser armazenadas e, conseqüentemente, maior será o espaço gasto em armazenamento, isto é detalhado mais à frente.

Segundo o autor anteriormente citado, as operações de restauração e realce são consideradas básicas no processamento de imagens, que tem por objetivo destacar detalhes que são de interesse para análise ou que tenham sofrido alguma deteriorização, buscando compensar deficiências específicas, geradas no momento de aquisição, na transmissão ou em alguma etapa do processamento. Quando se pode identificar experimentalmente a função que representa a deficiência ou construir um modelo matemático adequado, é possível buscar a função inversa e aplicá-la sobre a imagem deformada.

No contexto de análise de imagens científicas e industriais, é comum serem necessários dados relacionados aos objetos presentes na imagem. As operações de segmentação procuram isolar regiões de pontos da imagem pertencentes a objetos para posterior extração de atributos e cálculo de parâmetros descritivos.

A operação de segmentação mais simples é a limiarização ou separação por "tom de corte". A partir da separação e caracterização dos objetos, as operações de extração de características dos objetos, as operações de extração de características podem ser consideradas. Essa abordagem é interessante quando a imagem apresenta duas classes - o fundo e o objeto. Se a intensidade dos valores dos pixels do objeto encontra-se num intervalo e a intensidade do fundo encontra-se fora desse intervalo, uma imagem binária pode ser obtida usando-se uma operação de limiarização (*thresholding*) que agrupa os pontos do primeiro intervalo com o valor 1 e o do segundo com o valor 0. Ou seja, tudo o que está acima desse tom é considerado objeto; tudo que está abaixo é considerado fundo da imagem.

Utilizando-se imagens já segmentadas (em objeto e fundo) ou binárias, torna-se possível obter dados relevantes, atributos das regiões ou objetos destacados. Os tipos de atributos ou características mais comuns são: número total de objetos, dimensões, geometria, propriedade de luminosidade e texturas. As características dimensionais são, por exemplo: área, perímetro, centro de gravidade, larguras máxima e mínimas, comprimento etc. As propri-

edades geométricas descrevem as formas dos objetos, tais como circularidade, retilinearidade, concavidade, eixos principais etc.

Uma vez que os descritores da imagem e dos objetos segmentados encontram-se disponíveis, passa-se à etapa seguinte, que consiste em distinguir objetos na imagem agrupando esses parâmetros de acordo com sua semelhança para cada região de pixels encontrada. Essa é a função do processo de classificação e reconhecimento.

De acordo com CONCI et al, 2008 [1], o objetivo de um sistema de visão computacional é tomar decisões a partir da extração de informações do mundo real através das imagens. A tomada de decisão pode ser feita a partir de indagações simples a respeito de parâmetros extraídos dos objetos ou de algoritmos mais complexos de inteligência artificial. A informação não está disponível diretamente e precisa ser obtida. Em algumas situações, é necessário confrontar os dados obtidos com os disponíveis em banco de dados de imagens.

3.1.2 Conceito de imagem digital

Toda imagem digital é representada por uma matriz de pixels, que é a cor de um ponto na imagem (Figura 3.2). Em relação ao tamanho ele é representado pela resolução da imagem e consiste no número máximo de pixels que a imagem pode representar na largura e altura. A definição de resolução de uma imagem é entendida como a quantidade de informação que a imagem contém por unidade de comprimento, isto é, o número de pixels por polegada, ppi (pixels per inch). A resolução da imagem pode também ser definida, de forma imprópria, pelo seu tamanho, ou seja, pelo número de pixels por linha e por coluna. Podemos citar o exemplo de uma mesma imagem que tem 128 x 128 pixels de resolução ou 512 x 512 pixels, e o que diferencia as duas é que a que tem maior resolução tem mais detalhes da imagem gerando com isto uma imagem mais definida, como pode ser observado pela Figura 3.3. Nesta figura temos o item a) uma imagem com um maior número de pixels que os itens b) e c). Mas só este atributo não é o responsável pela nitidez da imagem, aqui pode ser acrescentado a profundidade que é a quantidade de cores ou tons que a imagem possui. Na Figura 3.4 temos uma imagem com a mesma resolução mas com uma profundidade de cores diferente, sendo que o item a) possui uma quantidade de tons de cinza maior que o item b) e c). Isto também é válido para imagens coloridas.

Outras características das imagens são o brilho e o contraste que tem basicamente a ver com a variação da cor dos pixels e a variação que eles apresentam. Usando como exemplo uma imagem em tons de cinza, ela será escura ou com baixo brilho se os pixels estiverem próximos do valor 0 (que é o preto) e clara ou com alto brilho se os pixels estiverem próximos do valor 255 (que é o branco). Já o contraste tem a ver com a variação dos tons de cinza, quanto maior a variação maior é o contraste.

Um outro conceito da imagem digital é o histograma que faz uma representação gráfica da função de distribuição dos tons de cinza dos pixels da imagem e fornece uma descrição global da aparência da imagem, mas nada diz sobre seu conteúdo, apesar disto constitui-se em informação crucial para o processamento digital de imagens. Um exemplo do uso é a retirada automática do fundo da imagem e que foi usado neste trabalho.

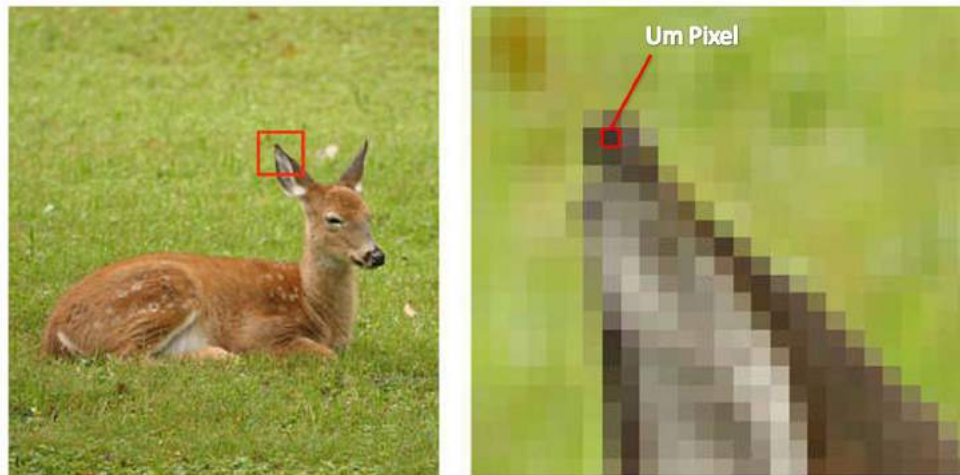


Figura 3.2: Pixel da imagem. Fonte:12G turma, 2014 [2]



Figura 3.3: Resolução da imagem. Fonte:Falcão, 2014 [3]

3.1.3 Aquisição/digitalização e Restauração/Realce

De acordo com GONZALEZ e WOODS,2008 [25], dois elementos são necessários para a aquisição de imagens digitais, sendo que o primeiro é um dispositivo físico e o outro é chamado de digitalizador, que faz a conversão da saída elétrica de um dispositivo de sensoramento físico para a forma digital. O primeiro elemento deve ser sensível a uma banda do espectro de energia eletromagnética e que produza um sinal elétrico de saída elétrico de saída proporcional a um nível de energia percebida.

De acordo com CONCI et al, 2008 [1], quando se realiza alguma medida sobre os objetos contidos na imagem surge um outro parâmetro, a escala. Pode-se defini-la matematicamente idêntica a resolução, só que as unidades serão mais variadas possíveis. Caso não seja conhecida, calculando-se a média das razões entre distância e o número de pixels correspondente. A partir dessa escala, medições podem ser feitas na imagem com qualidade aceitável. É claro que, se os objetos não estiverem em um mesmo plano ou na mesma distância do ponto de captura, as medidas passam a ser mais complexas e envolvem transformações projetivas.

Imagens monocromáticas são imagens digitais nas quais cada pixel possui apenas uma banda espectral. As imagens monocromática podem ser binárias ou em escalas de cinza.

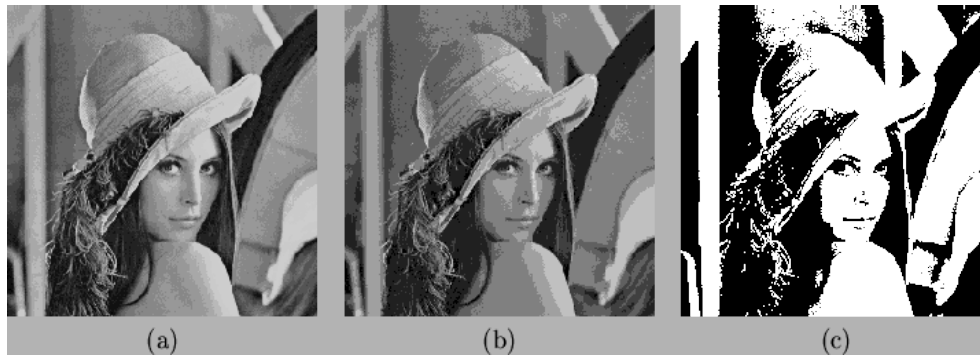


Figura 3.4: Profundidade da imagem

Uma imagem binária, é uma imagem monocromática na qual os pixels assumem apenas dois valores, geralmente 0 ou 1. Uma imagem em escala de cinza, é uma imagem monocromática na qual os pixels podem assumir uma faixa de valores intermediários dependendo do número de bits usado para descrever o tom de cada pixel. Se for usado um byte pode haver 256 tons diferentes, variando do preto ao branco(0 a 255).

Uma imagem binária contém apenas dois tons de cores. Embora os computadores tenham cada vez mais capacidade quanto a representação de cores, sistemas binários ainda são bastante úteis. Isso porque os algoritmos para obtenção de propriedades das imagens binárias são facilmente entendidos e dependem menos tempo e memória, sendo mais rápidos do que aqueles que envolvem imagens em tons de cinza ou coloridas.

CONCI et al, 2008 [1] explica que, geralmente, sistemas binários são úteis em casos em que a silhueta contém informação suficiente para permitir o reconhecimento do objeto e quando o ambiente para aquisição da imagem é bem controlado. Existem muitas aplicações em que esses tipos de sistemas obtêm bons resultados, por exemplo, em reconhecimento de caracteres, análise de cromossomos, reconhecimento de partes de peças industriais, entre outras. Nesses casos, se a forma de aquisição for em tons de cinza, pode-se utilizar uma técnica de limiarização (thresholding) para separar o objeto do fundo. O número de tons entre os valores-limites, branco e preto, que se pode representar em tons depende de quantos bits são alocados na matriz da imagem para armazenamento o tom de cada pixel.

Uma imagem possui uma série de informações armazenadas, e uma das mais relevantes refere-se à cor ou ao tom de seus pixels. Para iniciar o processo de análise de uma imagem, pode-se observá-la a partir de uma ferramenta usual em estatística: o histograma. O histograma de uma imagem é simplesmente um conjunto de números indicando o percentual de pixels naquela imagem, que apresenta determinado nível de cinza ou cor. Esses valores são normalmente representados por um gráfico de barras e fornece, para cada nível de cinza, o número(ou o percentual) de pixels correspondentes na imagem.

Para o mesmo autor, o histograma de uma imagem fornece uma indicação de sua qualidade quanto ao nível de contraste e quanto a sua luminosidade média (ou seja, se a imagem é predominantemente clara ou escura). Um histograma é, portanto, uma representação gráfica da frequência de ocorrência de cada intensidade (nível de cinza ou cor) de uma imagem. Ele é construído da seguinte forma:

- analisa-se o tom de cada pixel
- faz-se a contagem do número de pixels de cada valor de intensidade
- representam-se esses valores na forma de uma tabela ou faz-se o gráfico da frequência correspondente a cada tom de cinza.
- finalmente normalizam-se esses valores dividindo-se pelo número total de pixels da imagem, obtendo-se uma idéia da frequência de cada tom.

Quando as imagens digitais passaram a ser utilizadas em diversas aplicações, a memória e o poder dos computadores disponíveis eram muito limitados e caros. Essas limitações encorajaram projetistas de aplicações a focar seus esforços em sistemas de visão binária. As imagens binárias ainda são úteis em várias aplicações, principalmente porque os algoritmos para computar propriedade de imagens binárias são mais simples. Eles também tendem a ser menos caros e mais rápidos que sistemas que operam em níveis de cinza ou como imagens coloridas. Isso é devido ao fato de os requerimentos de memória e processamento dessas imagens serem significativamente menores. A complexidade computacional de um sistema trabalhando com 256 tons de níveis de cinza será bem maior que um sistema trabalhando com uma imagem binária. O tempo de processamento é mais baixo porque muitas operações em imagens binárias podem ser executadas como operações lógicas em vez de operações aritméticas (CONCI et al, 2008 [1]).



Figura 3.5: Imagem em tons de cinza e binária respectivamente. Fonte: CONCI et al, 2008 [1]

Imagens binárias são úteis para identificar objetos representados na imagem; analisar a forma, nos casos em que as intensidades dos pixels não sejam significativas para uma informação; apresentar a imagem em um dispositivo de saída que tenha baixo nível de quantização; converter uma imagem com formas suaves para outra com arestas acentuadas. Em geral, sistemas que usam imagens binárias são úteis nos casos em que a silhueta contém informação suficiente para permitir o reconhecimento de um objeto e a iluminação do ambiente pode ser contralada adequadamente. Para gerar uma boa silhueta, os objetos devem ser separados facilmente do fundo. Isto é possível se houver apenas alguns objetos no cenário. O sistema de imagem binária geralmente usa um limiar ou vários limiares para separar objetos do fundo. O valor apropriado desse limiar depende da iluminação e das características refletoras dos objetos. A iluminação desempenha um papel significativo no processo de limiarização, já que provoca alterações nos tons da imagem. Uma técnica comum, utilizada para compensar a não-uniformidade da iluminação, baseia-se em projetar o padrão de iluminação em uma

superfície refletora branca. O valor ótimo de limiar pode ser determinado através do uso de técnicas com base nas propriedades estatística da imagem.

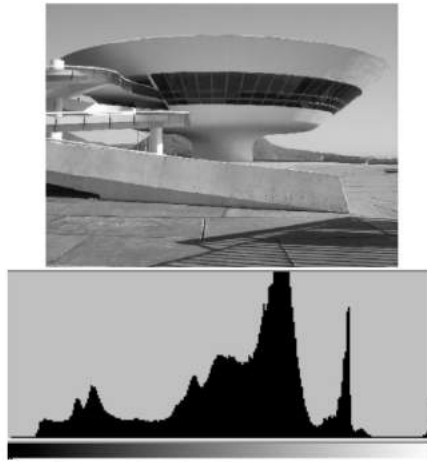


Figura 3.6: Histograma de uma imagem digital. Fonte: CONCI et all, 2008 [1]

A limiarização muitas vezes baseia-se na utilização do histograma, que tem uma função que pode assumir diversas formas: unimodal, bimodal, trimodal, multimodal, etc. O caso mais simples consiste em imagens em que o objeto a ser segmentado apresenta uma tonalidade bem diferente do fundo da imagem. Se a aparência desse histograma for bimodal, será possível estabelecer um limiar entre duas tonalidades, que segmentará as regiões do objeto e do fundo. É fundamental, no uso desse método, a escolha do ponto de corte. Uma ferramenta auxiliar usual é o histograma da imagem. A observação desse gráfico permite a localização do melhor valor de limiar para a imagem. Essa localização é tão mais fácil quanto mais bimodal for o histograma. O agrupamento por limiar é utilizado em imagens monocromáticas que apresentam regiões de picos e vales bem definidas no histograma. Em sua forma mais simples, a limiarização realiza a bipartição do histograma, convertendo os pixels cujo tom de cinza é maior a um certo valor de limiar (T) em brancos (ou pretos) e os demais em pretos (brancos), gerando assim uma imagem binária com duas classes. A binarização pode ser um método para separar os objetos do fundo da imagem e pode ser usado em algumas condições ideais, bem pouco comuns (CONCI et all, 2008 [1]).

O principal problema da limiarização é selecionar um valor de corte, que é onde no histograma é separado as cores que representam o objeto e o fundo. Devido a iluminação irregular ao longo da cena (sombra), à não-uniformidade nos tons do fundo e dos objetos e à presença de vários objetos diferentes, raramente o histograma de imagens reais apresenta dois picos distintos e disjuntos associados ao fundo e aos objetos.

Uma operação pontual em uma imagem digital cada tom da imagem é alterado de acordo com uma função definida, que pode ser linear (alteração de contraste, imagem negativa, etc) ou não-linear (equalização de histograma, operações logarítmicas, etc). Elas podem ser classificadas em dois grupos, baseados na observação da curva de tons ou do histograma.

O contraste é dado pela distribuição dos níveis de cinza da imagem e pode ser observado pela análise do histograma. Se a distribuição não é concentrada em torno de alguns níveis, mas abrange toda a faixa dos tons possíveis, pode-se dizer que a imagem possui bom contraste. Os

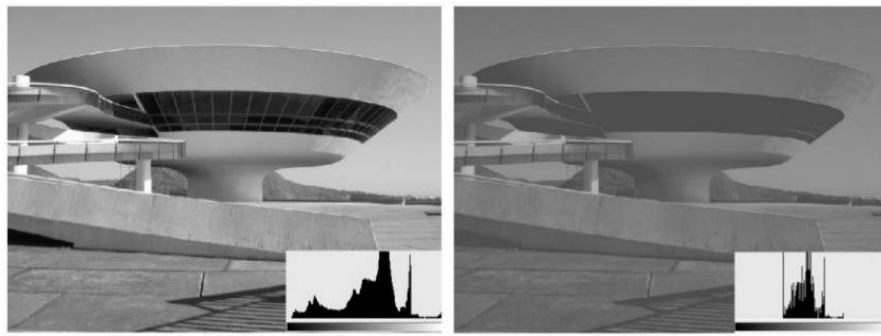


Figura 3.7: Efeito de compressão de histograma. Fonte: CONCI et al, 2008 [1]



Figura 3.8: Efeito de expansão de histograma. Fonte: CONCI et al, 2008 [1]

tons de cinza podem ser modificados através da curva de tons que age no histograma, gerando uma imagem mais clara ou com função inversa mais escura. Também pode-se modificar a imagem através da compressão e expansão do histograma, sendo que o primeiro faz com que a imagem perca o contraste e a nitidez. A contração da faixa de variação do histograma sempre tem o efeito de diminuir o contraste da imagem. O oposto, que seria a expansão, demonstram imagens com uma variação pequena da tonalidade, que não é detectada pelo sistema visual humano. Uma transformação que gera um efeito interessante é a geração da imagem negativa da imagem que pode ser gerada invertendo o histograma.

Segundo, CONCI et al, 2008 [1], existem três grandes classes distintas de operações em imagens: aquelas que são realizadas pontualmente nos pixels, as que são feitas em partes da imagem e aquelas realizadas em toda a imagem. Operações pontuais são operações em que um pixel em uma determinada posição, da imagem resultante, depende apenas do pixel, na imagem original, que se encontra nas mesmas coordenadas. Muitas das operações pontuais são operações que alteram apenas características de cor. Algumas dessas alterações podem ser realizadas de modo a gerar a percepção de mudanças da forma dos objetos na imagem. Qualquer operação pontual pode ser visualizada como um mapeamento de pixels da imagem original para a imagem processada.

No caso de operações que alteram apenas características de cor ou luminância, tais como brilho (intensidade luminosa), contraste, saturação (ou grau de pureza da cor), limiarização, pasteurização etc, pode-se usar a representação em um gráfico.

Embora as funções pontuais estejam restritas somente ao pixel a ser transformado, o processamento em si pode levar em consideração dados globais da imagem, por exemplo, o histograma. Muitas operações pontuais usam o histograma como parâmetro para fornecer resultados diferentes para o pixel da imagem processada.

Para CONCI et al., 2008 [1], operações locais são operações em que um pixel da imagem resultante depende de uma vizinhança de mesmo pixel na imagem original. As operações locais mais comuns são as operações de filtragem que usam a convolução com um *kernel* ou máscara de convolução, de dimensões $n \times n$, onde n é geralmente um valor ímpar 3, 5, 7 e 9 e o pixel a ser alterado é o central da região (mas pode assumir qualquer posição, bem como ser alterado por qualquer valor convencional).

Por exemplo, no caso de convolução com $n=3$, cada pixel da vizinhança N_8 ou dentro de uma região 3×3 vizinha do pixel na imagem original é multiplicada pelo valor correspondente do *kernel* e todos esses valores são somados, resultando no pixel novo. Os valores que ponderam a vizinhança são armazenadas em uma matriz (máscara convolução). Operações de suavização (*bluer*), realce (*sharpen*), detecção de bordas e muitas outras podem ser realizadas assim.

Há operações locais que mudam o tom do pixel para novas posições (x_i, y_i) ou usam o seu tom, $f(x_i, y_i)$, para criar, por exemplo, pixels adicionais a partir de interpolações. A técnica de interpolação não deixa de ser uma operação local que depende apenas de uma vizinhança, embora algumas técnicas de interpolação sejam muito sofisticadas e em geral muitas custosas utilizem na vizinhança bem extensa.

As operações locais, segundo GOMES, 2001 [26], também conhecidas como operações de vizinhança ou filtros espaciais, são operações onde o tom de cinza de determinado pixel na imagem de saída é função não apenas de seu tom de cinza de entrada, mas também dos tons de cinza de seus pixels vizinhos, que é calculado pela média ponderada dos pixels da vizinhança. Estas operações quando escolhidas o *kernel* adequado podem implementar diversos procedimentos, como a redução de ruído (mediana, gaussiana), aguçamento de detalhes (básico, alto reforço) e detecção de bordas (filtros espaço). Estes *kernels* são matrizes quadradas normalmente de 3×3 que como exemplo temos o da mediana, gaussiana, que são mostrados na Figura 3.9. E por último temos as operações geométricas que tem a relação entre os pixels alterados e a mais comum é a reamostragem.



Figura 3.9: Kernels

De acordo com CONCI et al., 2008 [1], operações globais são operações em que um pixel da imagem resultante depende de um processamento realizado em todos os pixels da imagem

original. Nesse grupo de operações estão as que mudam o domínio de descrição, tais como as transformações de Fourier, Wavelet, Hough, co-senos(usada para codificação) e por funções iterativas ou fractal.

Todas essas transformadas geram informações sobre a imagem original. Muitas delas transformam imagens em dados, sendo portanto, técnicas também de análise de imagens, e não apenas de processamento. A transformação de Fourier, por exemplo, é base fundamental para toda a teoria de processamento de sinais e com ela pode-se realizar uma série de operações muito importantes com imagens, definir filtros, identificar texturas, etc. A transformada de Wavelet decompõe a imagem em uma estrutura multi-resolução e a descreve como coeficientes das diversas bases usadas, o que é muito utilizado para compressão de imagens, reconhecimento e busca em banco de imagens. A transformada de Hough procura identificar formas geométricas na imagem, tais como retas e círculos, descrevendo a imagem no domínio dos parâmetros dessas formas.

As transformações geométricas são operações que levam o tom dos pixels na posição (x_0, y_0) da imagem origem para outra posição (x_d, y_d) do espaço em uma imagem destino, ou seja, "modificam a posição" do pixel no espaço da imagem. As principais operações são: translação, mudança de escala, rotação e espelhamento.

3.1.4 Segmentação

A segmentação tem como objetivo reconhecer regiões de uma imagem como objetos, para isto ele divide uma imagem em regiões e distingue essas regiões como objetos independentes deles mesmo e do fundo da imagem. No caso de diferenciar objetos de um fundo podemos ter uma imagem binária onde o fundo é representado pela cor preta e os objetos pela cor branca. De acordo com GOMES, 2001 [26], a segmentação costuma ser a etapa crítica da sequência padrão de PADI, pois é através dela que se reconhece e se identifica os objetos de interesse, sobre os quais será feita a análise.

Dentre as várias técnicas de segmentação podemos citar a limiarização e a segmentação por contornos. A primeira é a mais simples e a mais utilizada e a outra simula o funcionamento da visão humana, detectando as bordas dos objetos. A limiarização usa tons de cinza da imagem para distinguir os objetos e para distingui-los usa o conceito de limiar ou tom de corte que é feito através da análise do histograma da imagem. No caso de diferenciar o fundo dos objetos da imagem usa-se a limiarização bi-modal que separa a imagem em duas partes. Esta técnica foi empregada para separar plântulas do fundo neste trabalho.

Existem casos em que é preciso diferenciar os objetos além do fundo, que também pode ser feito pela limiarização e é chamada de limiarização multi-modal onde pode-se pela análise do histograma realizar várias cores na imagem, separando assim em quantas partes forem necessárias a imagem. Detectar esses limiares de corte não é tarefa simples, mas deve ser feita de uma forma automática, e existem algumas soluções já desenvolvidas como o métodos dos mínimos, a segmentação de Otsu e a limiarização adaptativa. A segmentação por contornos só deve ser usada se a limiarização não produzir resultados satisfatório pelo custo computacional desta.

3.1.5 Extração de atributos

Segundo GOMES, 2001 [26], muitas vezes o resultado da segmentação não é adequado, sendo necessária, para se corrigir as imagens binárias resultantes da segmentação, a aplicação de procedimentos de pós-processamento, como a separação de objetos que se tocam ou o agrupamento de objetos para formar objetos mais complexos. Tais procedimentos são geralmente implementados através de operações lógicas e operações morfológicas. As lógicas são realizadas entre imagens usando operadores lógicos /textit not, and e or (que são a base para outras operações que derivam destas três), passando pixel a pixel da imagem, gerando uma nova imagem. As morfológicas tem como base as operações de erosão e dilatação e são semelhantes às operações locais, porque são orientadas pelas suas vizinhanças que é definida por uma pequena imagem binária. Estas duas operações são opostas, enquanto a erosão tende a encolher os objetos a outra faz com que eles cresçam. Podem ser definidas novas operações morfológicas juntando com as operações lógicas, como o preenchimento de buracos (*hile filling*), a eliminação de objetos que tocam na borda da imagem (*border object killing*) e a extração de objetos marcados (*object marking*). Um dos problemas de segmentação mais corriqueiros é quando os objetos que se tocam, neste caso é utilizada a técnica dos métodos divisores de águas, que usa operações morfológicas juntamente com a limiarização.

Extração de atributos segundo GOMES, 2001 [26], é a etapa da sequência padrão onde se inicia a análise da imagem. Nela são realizadas medidas sobre a imagem segmentada e/ou sobre a imagem em tons de cinza, extraindo-se atributos característicos. Existem dois tipos de medidas, medidas de campo que são realizadas na imagem toda e medidas de região que são realizadas sobre os objetos segmentados na imagem. A primeira pode ser dividida em contagem de Objetos, área, interceptos e interseções, intensidade e textura; enquanto a outra pode ser em tamanho, forma, posição, intensidade e textura.

3.2 Aplicações de Processamento de Imagens na Agricultura

A agricultura de precisão abrange a utilização de tecnologias de sistemas de informação, como imagens de satélite, sistema de posicionamento geográfico (GPS - *Global Position System*), uso de VANT (Veículo Aéreo Não Tripulado), análise de imagens e outros que fornecem subsídios para a tomada de decisões mais precisas. A utilização destas tecnologias proporcionam uma economia de tempo e de recursos que serão de grande valor para o agricultor.

De acordo com HONDA e JORGE, 2011 [27], a aplicação de processamentos computacionais na área agrícola permite o desenvolvimento de softwares, onde cada qual possa suprir as necessidades de informação para o agricultor, resultando em grandes benefícios ao setor agropecuário. Técnicas computacionais, sensoriamento remoto e sistemas de posicionamento global são utilizados em conjunto para o desenvolvimento de *softwares* especialistas que ajudam na gestão agrícola, podendo obter dados através desses sistemas, que auxiliam fortemente na agricultura de precisão.

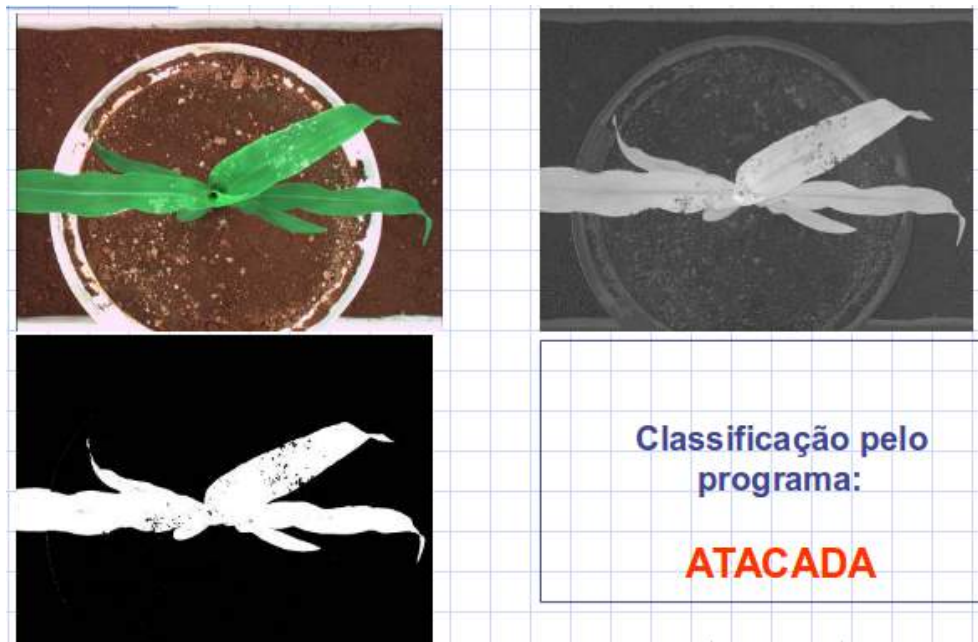


Figura 3.10: Planta atacada pela lagarta do cartucho. Fonte: [4]

O sensoriamento remoto pode ser entendido como a utilização de sensores que coletam dados dos alvos, sem que haja um contato físico com ele. Os sensores capturam os dados através das ondas eletromagnéticas, também denominadas radiação eletromagnética (REM), que são emitidas ou refletidas pelos objetos. A REM é ordenada no espectro eletromagnético de acordo com seu comprimento e frequência. Desta forma, cada sensor é responsável pela captação da intensidade de radiação do objeto, resposta espectral, compondo imagens e dados a respeito do alvo.

Para a agricultura de precisão, esta tecnologia é uma importante fonte de aquisição de dados, podendo periodicamente averiguar o estado nutricional das plantas, variabilidade e os parâmetros do solo, condições vegetais para manejo, mapeamento de produtividade de cada colheita, doenças e infestações por pragas conforme (Figura 3.10) entre outros.

Algumas formas de avaliar a imagem de uma vegetação é através do seu comportamento espectral, índices de vegetação e imagens multiespectrais. O comportamento espectral é a forma como o objeto absorve ou reflete as ondas eletromagnéticas.

Atualmente, as técnicas de processamento digital de imagens e inteligência artificial ajudam nas análises de imagens, podendo detectar problemas em plantas como estresse nutricional, doenças, pragas e plantas invasoras (Figura 3.12).

A análise de imagens pode ter aplicações na identificação do estresse nutricional, de doenças, pragas, plantas invasoras na lavoura, nos testes de vigor de sementes, no posicionamento de implementos agrícolas, automação da colheita, etc. Estas análises aumentam a precisão dos resultados, fornecendo rapidez em contraste com o processamento manual, fazendo assim com que as tomadas de decisões sejam mais rápidas e precisas.

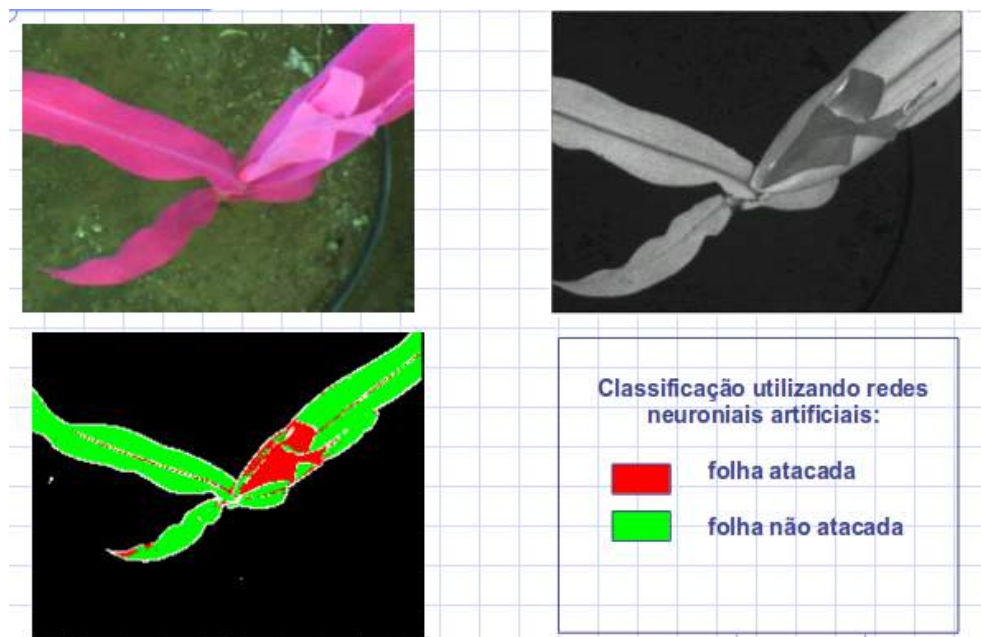


Figura 3.11: Planta atacada pela lagarta elasmobranch. Fonte: VARELLA, 2006 [4]

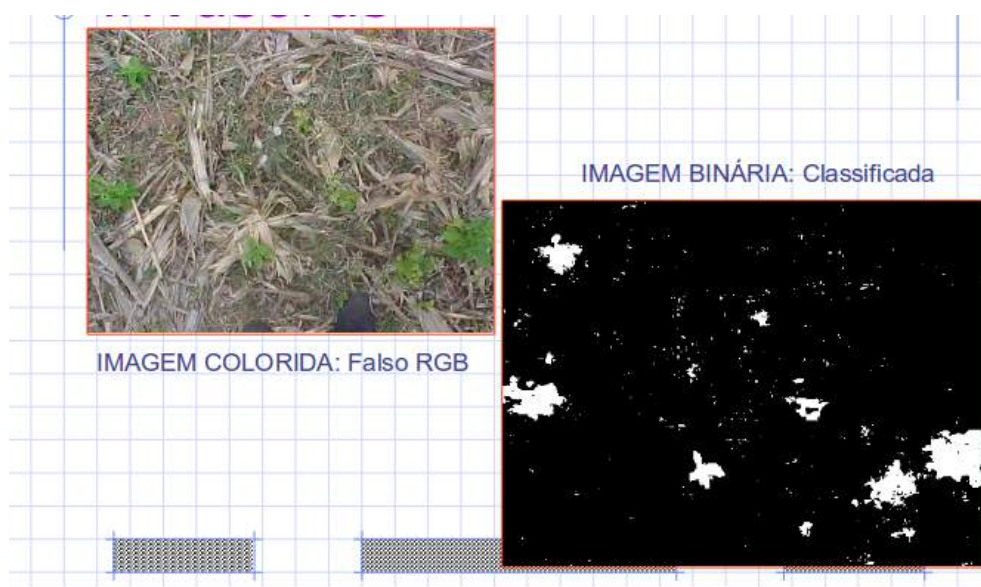


Figura 3.12: Planta com traços de planta invasora. Fonte: VARELLA, 2006 [4]

Capítulo 4

Experimentos

Neste capítulo é relatado o desenvolvimento do programa proposto e todas as alterações pelas quais passou até chegar a versão final. São relatados, também, as características e o problema de cada versão do desenvolvimento, bem como as soluções encontradas. Tais etapas são representadas na Figura 4.1. Ela fornece uma visão geral das fases que o programa passou, suas características, linguagem usada e seus recursos.

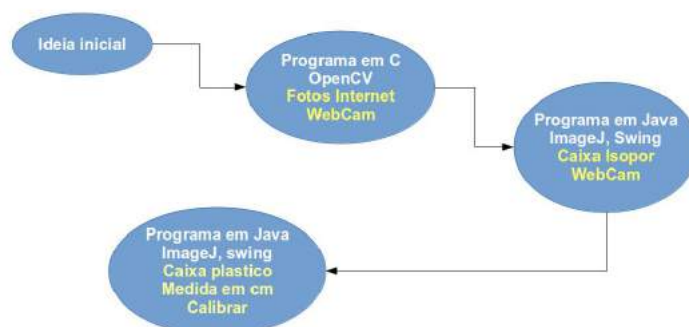


Figura 4.1: Fases do desenvolvimento.

Como mencionado em capítulos anteriores, o teste de comprimento de plântulas é simples. Além disso fornece bons resultados, pois o equipamento usado é uma régua e o treinamento é muito simples. Mas o processo é repetitivo e trabalhoso se for realizado para vários lotes.

Motivado por essas dificuldades e com o objetivo de automatizar o processo, por meio desta pesquisa, foi proposto um sistema computadorizado. Para essa automação, foi necessário desenvolver um programa que capturasse a imagem da plântula, fizesse o seu processamento, separasse o fundo da imagem das plântulas para finalmente, verificar o tamanho de cada uma delas em centímetros. As fases do programa estão descritos na Tabela 5.1.

Tabela 4.1: Tabela descritiva das fases do programa.

Fase	Linguagem e frameworks	Hardware	Características
1	C com OpenCv	Sem	- Uso <i>webcam</i>, retirada do fundo da imagem. - Separação da imagem original em várias. - Interface simples com usuário, só processa imagem passada como parâmetro
2	Java com <i>ImageJ</i> e <i>Swing</i>	Caixa de isopor	- Uso da <i>webcam</i> em java - Retirada do fundo da imagem automaticamente pelo seu histograma - Interface simples
3	Java com <i>ImageJ</i> e <i>Swing</i>	Caixa de plástico	- Colocada na caixa um quadrado vermelho de 1 cm para guia no processamento do tamanho da plântula. - Criada a opção de calibrar.

O sistema é composto de um hardware que possui uma estrutura de isopor (Figura 4.2), onde na sua parte superior são colocadas uma *webcam* (Figura 4.3) e uma fonte de luz (a luz deverá ser branca) para que seja capturada a imagem das plântulas que serão colocadas no fundo da caixa de isopor.

4.1 Fase 1 - A ideia inicial e o programa desenvolvido na linguagem C

Na concepção da ideia a linguagem de programação escolhida foi Java por ser muito usada no meio acadêmico e porque tem muitas bibliotecas disponíveis para automatizar o processamento e resolução de várias partes do desenvolvimento de software. A proposta era transformar a imagem em tons de cinza (para facilitar a sua análise) e converter em uma matriz de pontos, para ser processada por um autômato. A partir da análise dos contornos das plântulas elas poderiam ser separadas e depois medidas. A ideia não foi levada adiante porque durante a pesquisa e o desenvolvimento, foi encontrado um programa desenvolvido pelo prof. Hemerson Pistori [5] de segmentação por limiarização, que foi modificado para atender aos requisitos do sistema proposto. Este era desenvolvido na linguagem C e usava a biblioteca *opencv*, conforme Figura 4.4. O programa isola um objeto pela composição de sua cor do resto da imagem, da seguinte forma: no início do processo ativa a *webcam* do computador e inicia a captura de quadros da imagem, que são processados de acordo com valores máximos e mínimos das cores RGB (textit Red, Green, Blue) que podem ser configurados em um painel do programa (Figura 4.4 - parte esquerda). É gerada uma imagem

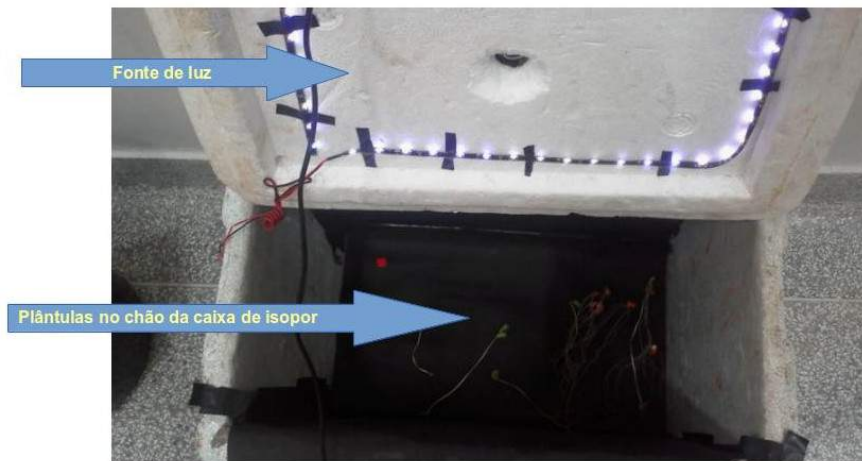


Figura 4.2: Caixa de Isopor.



Figura 4.3: Webcam na parte superior da caixa de isopor.

nova, em que todos os pixels da imagem original que não satisfazem a condição são setados para a cor preta, e o restante não é alterado, ficando com a cor original (Figura 4.4 - canto inferior direito). Além do painel já citado para configurar os valores das cores, isso pode ser feito usando o mouse na imagem original, desenhando um retângulo, onde estão os pixels que serão preservados na nova imagem. Na Figura 4.4 pode-se perceber, que os valores RGB usados, isolam a bolinha de tênis que está na imagem, porque as cores dela foram selecionadas.

O desenvolvimento do sistema com esse software de limiarização trouxe bons resultados, pois respondeu algumas questões que existiam no início do projeto, que são: é possível fazer a captura de imagens pela *webcam* de maneira simples e o seu processamento é possível de ser realizado em tempo hábil. Como separar as plântulas do restante da imagem? Com base nessa nova trajetória tomada pelos testes realizados, o tratamento do fundo da imagem ficou resolvido, e depois de separada as plântulas do fundo da imagem, o próximo passo é a separação das plântulas umas das outras, em várias imagens. Nesse ponto constatou-se

que o processo não é tão simples, não tendo uma solução tão trivial como se esperava, como mostra a Figura 4.5.

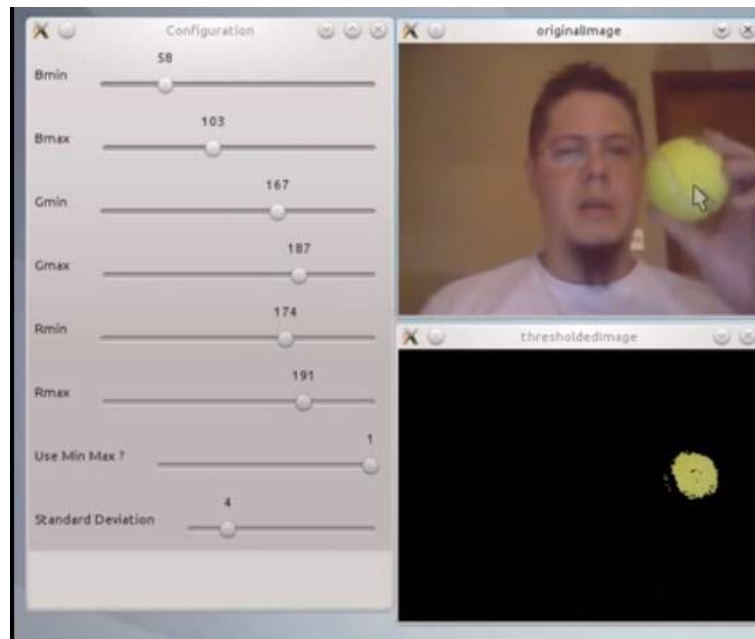


Figura 4.4: Programa original de segmentação por limiarização, Pistori 2015 [5] .

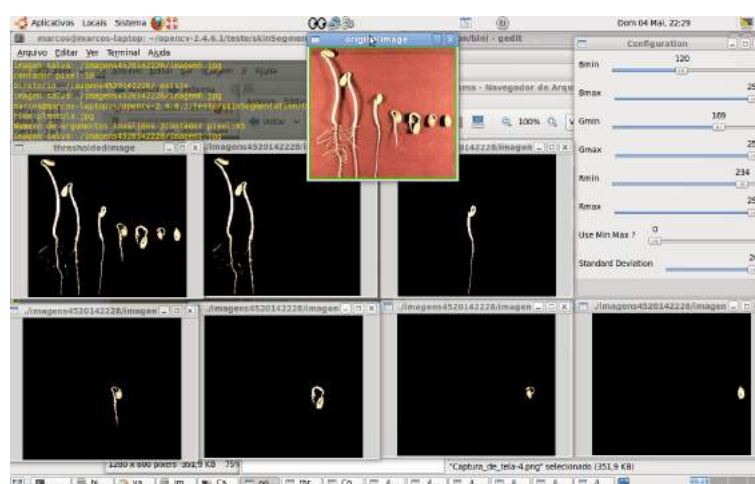


Figura 4.5: Programa alterado e processando uma imagem de soja.

Continuando o desenvolvimento, foram realizadas alterações para atender as necessidades do sistema, mas tomando o cuidado de não perder a característica de separação do fundo da imagem. A primeira alteração realizada foi na origem da imagem que antes era obtida da *webcam*, e agora utiliza o nome da imagem que é informada com o nome do programa (porque os testes com a mesma imagem deveriam ser repetidos). O programa de posse do nome da imagem gera uma nova imagem em que o fundo é retirado, deixando somente as plântulas. Em seguida essa imagem é processada novamente e as plântulas são separadas. Testes preliminares foram realizados com uma imagem de plântulas retirada da Internet.

Os resultados obtidos foram os melhores possíveis, o fundo foi removido e trocado pela cor preta. Todas as plântulas foram separadas e salvas em arquivos de imagem individuais com extensão jpg.

Para melhorar a organização das imagens que seriam criadas para cada plântula foi criado um diretório que tem o seu nome composto pela cadeia “imagens” e data e hora atuais, onde são salvas as imagens processadas. Por exemplo se a data é 01/01/2015 e a hora 15h o diretório criado é: imagens010120151500. Foram criadas funções para organizar melhor o programa, citadas abaixo:

`dataHoraAtual()` - extrai a data e hora atual do sistema para compor o nome do diretório,

`criarDiretorio(std::string nome)` - cria o diretório com o parâmetro nome,

`void separateImage(int primPixel, int ultPixel, int imagemCriada)` - recebe a imagem sem o fundo e separa as plântulas em arquivo jpg distintos no diretório criado

`saveImage(char* outFileName, IplImage* Imagem)` é a função que salva as imagens que foram separadas pela função anterior. O código fonte dessas rotinas são mostrados no apêndice A.

O processamento da primeira imagem produziu a total separação do fundo. Como pode ser visto na Figura 4.5, a imagem colorida e a imagem original, depois a esquerda logo abaixo a imagem é igual a original, mas já retirado o fundo. Em seguida, duas plântulas e posteriormente um arquivo para cada plântula. Tal resultado foi excelente dando a impressão de que a separação das plântulas na imagem seria algo trivial, mas com imagens de um experimento real de milho e soja, o programa não conseguiu separar todas as plântulas (Figura 4.6).

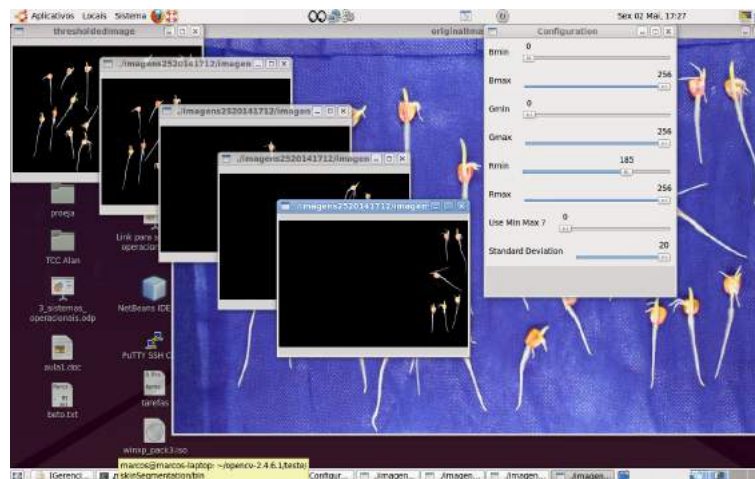


Figura 4.6: Programa alterado com imagens processadas.

As imagens usadas para o teste possuíam um fundo azul ou branco, que são as cores disponíveis para as folhas de germinação (Figura 4.7). Depois de alguns testes com o fundo azul, foi possível retirá-lo, deixando somente as plântulas. Como pode ser notado na Figura 4.8, as plântulas não ficam totalmente separadas umas das outras, podendo uma invadir a região da outra no sentido vertical, dificultando a separação delas com o algoritmo desenvolvido no primeiro teste.



Figura 4.7: Imagem original com fundo azul.

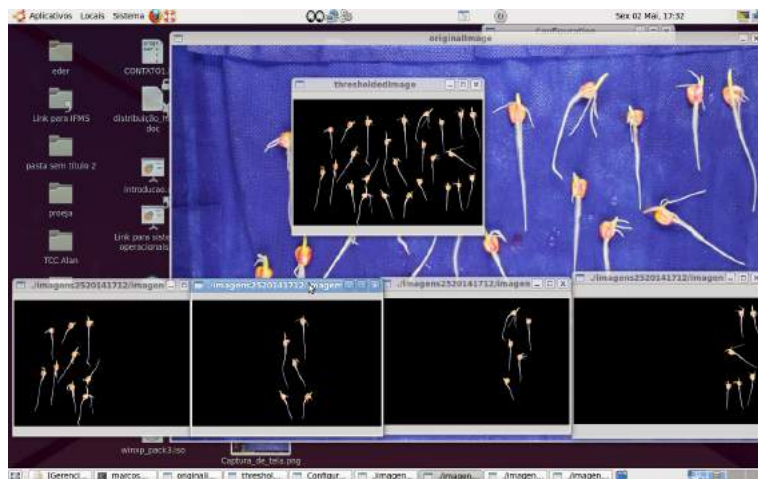


Figura 4.8: Programa com todas as imagens: original, com fundo azul e tentando separar as plântulas.

Para corrigir isto foi desenvolvido o algoritmo mostrado abaixo, que analisa a imagem verificando se todos os pixels na vertical são pretos. As linhas 6 a 14 mostra esta busca na matriz da imagem através das repetições. As linhas 15 a 18 verificam se depois de percorrer a linha inteira da imagem não for encontrado nenhuma cor preta ela guarda a coluna na variável `primPixel`, marcando assim o número da coluna da matriz da imagem. Depois disto volta ao laço da linha 6 para procurar uma coluna totalmente preta novamente, indicando o fim da plântula que é guardado na variável `ultPixel`. De posse do início e fim das coordenadas da plântula, é recortada a imagem (linha 25). Mas quando uma plântula termina e outra já começou não é possível separar somente uma e nos testes feitos são agrupadas várias plântulas.

Código fonte que separa as plântulas.

```
1 int primPixel = -1;
2 int ultPixel = -1;
3 int contCor = 0;
```

```

4  bool tudoPreto = true;
5  //varre a imagem procura colunas todas pretas
6  for (int col=0; col<thresholdedImage->width; col++) {
7      for (int row=0; row<thresholdedImage->height; row++) {
8          int B = data[row*step+col*channels+0];
9          int G = data[row*step+col*channels+1];
10         int R = data[row*step+col*channels+2];
11         if ( B != 0 && G != 0 && R !=0 ) {
12             tudoPreto = false;
13         }
14     }
15     if (!tudoPreto) {
16         if (primPixel == -1) {
17             primPixel = col;
18         }
19         contCor++;
20     } else {
21         if (primPixel != -1 && ultPixel == -1) {
22             ultPixel = col;
23             printf("contador pixel:%d \n", contCor);
24             if (contCor > 5) {
25                 separateImage(primPixel, ultPixel, imagemCriada);
26                 imagemCriada++;
27             }
28             primPixel = -1;
29             ultPixel = -1;
30             contCor = 0;
31         }
32     }
33     tudoPreto = true;
34 }

```

Quando foram usados os recursos da linguagem C para a separação das plântulas usando a matriz, onde se encontravam os pixels da imagem, não foi um trabalho fácil, surgiram dificuldades relacionadas à forma como esta linguagem trabalha com as cores da imagem. A biblioteca opencv também não apresentou recursos conhecidos para resolução deste impasse. Estes fatores começaram a trazer uma demora no desenvolvimento do projeto. Por outro lado a criação de formulários (para mostrar as imagens gerados no diretório citado), de menus e um programa mais intuitivo para o usuário nesta linguagem, trariam um custo muito alto de esforço e tempo, que poderia comprometer o projeto.

4.2 Fase 2 e 3 - Programa desenvolvido em Java

A solução encontrada foi mudar para a linguagem Java com o uso do *framework swing* e *ImageJ* para processamento de imagens. Todo o sistema foi refeito usando as novas características e funções do *framework*, que, em poucos dias, já estava implementado o processamento da imagem, de uma forma diferente do programa da fase 1 (Tabela 5.1). O processamento da imagem ficou mais simples e com um tratamento mais detalhado dos

seus pixels, além de sua manipulação ter se tornado mais fácil. A interface gráfica criada pelo *framework swing* com a construção dos menus e da visualização das imagens por meio de frames, foi criada rapidamente.

4.2.1 Fase 2 - Usando os *frameworks ImageJ* e *Swing*

Nesta fase, a ênfase maior é dada no processamento da imagem e o término do projeto com o uso da caixa de isopor e a imagem adquirida pela *webcam*. A imagem agora é transformada em tons de cinza com um comando (`imageConverter.convertToGray8()`) do *framework* e, com isso foi criado o histograma da imagem que gerou a possibilidade de retirar o fundo através da sua análise de uma forma automática, que no programa da primeira fase era feito de forma manual. Nesta nova fase são gerados dois novos arquivos: um com informações da imagem e outro com o conteúdo dos pixels da imagem.

O primeiro contém as informações do diretório e o nome da imagem, a quantidade de plântulas detectadas, o número de pixels que ela possui e qual o início da coordenada na imagem relativo ao seu posicionamento e finalmente sua largura e altura. Um exemplo deste arquivo é mostrado no apêndice B. O outro arquivo possui as informações dos pixels em tons de cinza, mas representado de uma forma diferente do que está na imagem. Ele foi criado para analisar os pixels das plântulas e achar erros de processamento que com o debugador não eram detectados e nem com a análise da imagem em um programa próprio para isso. Ele é gerado com a extensão `.csv` que é lido por uma planilha eletrônica e cujo objetivo é ajudar na visualização das alterações geradas nas imagens e acompanhar as transformações nos pixels que são criadas. Um exemplo do auxílio desse arquivo no desenvolvimento pode ser citado quando a imagem era transformada em tons de cinza para branco e preto; a transformação era feita de forma errada porque achava que os pixels tinham sido gerado com uns valores de cor e, na análise do arquivo `csv`, percebeu-se que os valores eram diferentes. Isso não seria possível somente com o arquivo de imagem, já a informação não estava clara e seria preciso uma análise de pixel a pixel. Já com o arquivo, ficou fácil de ver na planilha eletrônica porque mostrava os valores dos pixels em forma numérica e não a imagem.

A classe criada se chama `ProcessarImagem` e os seus métodos e um pequeno resumo são mostrados na Tabela 4.2.

Os problemas relatados com a criação de formulários na fase 1 e menus para o usuário foram superados com o uso do *framework swing*, que permite usar estes recursos de maneira mais simples. As telas do sistema criado consistem em um formulário principal que se chama processamento de imagens e possui um menu chamado Arquivo com 3 opções mostradas abaixo e cada opção gera abas mostrando todas as imagens processadas no sistema.

1 - Processar WebCam - ativa a *webcam* do computador para que seja capturada a imagem das plântulas e processada.

2 - Processar Imagem - permite a escolha de uma imagem que contenha plântulas para ser processada

3 - Sair - Finaliza o programa.

Tabela 4.2: Tabela dos métodos criados para o processamento da imagem.

Processamento	Métodos
Ler o arquivo e processar para tons de cinza	public ProcessarImagem(String nomeArquivo); public void processarImagemCinza();
Gerar o histograma e retirar o fundo	private void gerarHistograma();
Retirar ruídos da imagem para terminar de separar as plântulas	public void retirarRuidoImagem();
Troca o cor dos pixels para branco se não forem preto	public void pintarBranco();
Separa as plantulas e salva em arquivos distintos	public void separarPlantulas(); public void recortarParteImagem(Rectangle rect, int numPlantula);
Arquivos que auxiliam e contem informações do sistema	public void gerarArquivoInformacoes(); private void gerarArquivoCsv(String nomeArquivo, ImageProcessor ip);
Métodos para funções diversas	public int larguraImagem(); public int alturaImagem(); public void capturaWebCan(); public void capturaContinuaWebCan(); public void mostrarImagensFrame(); public void mostrarImagem(); public void duplicarMostrarImagem(String titulo); private void gerarArquivoImagem(ImagePlus image, String arquivoImg, String nomeDir);

O processamento da imagem consiste em transformá-la em tons de cinza, depois retirar o fundo dela automaticamente pela análise do histograma. O próximo passo é a separação das plântulas em imagens separadas, como se fez no sistema anterior, com a diferença que agora as plântulas podem interceptar umas as outras, porque o sistema consegue separá-las.

É criado um diretório (com o mesmo nome do arquivo que foi selecionada para processamento) para armazenar todas as imagens geradas pelo sistema, desde a imagem original até as novas imagens geradas durante o processamento realizado pelo programa. Os arquivos csv e de informações já citadas são gravados nesse diretório. Por exemplo, se a imagem a ser processada tem o nome de soja1.jpg, o diretório criado terão nome de soja1, o arquivo de informações soja.jpg.info e, finalmente, são criados os arquivos de imagem:

- 01-imgOriginal - Imagem sem nenhuma alteração, preservando a imagem original
- 02-imgCinza - Imagem transformada em tons de cinza
- 03-imgCinzaProc
- 04-imgBranco

- 05-plantula1
- 0x-plantulax

onde x é o total das plântulas que foram detectadas. Na Figura 4.9 é mostrado a `imgOriginal` e pode-se verificar que nas abas estão os nomes dos arquivos de imagem, no caso, foram detectadas 5 plântulas na imagem. A Figura 4.10 mostra a mesma imagem já transformada em tons de cinza e, com ela, é gerado o histograma, mas ainda não é retirado o fundo da imagem.

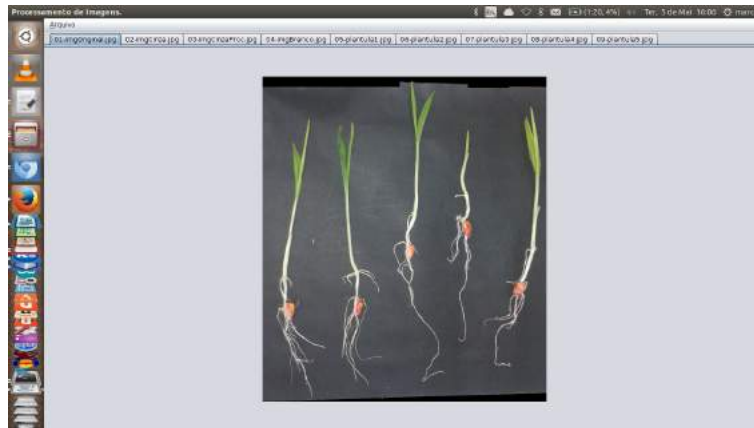


Figura 4.9: Imagem original das plântulas.



Figura 4.10: Imagem em tons de cinza.

Na Figura 4.11, é mostrada a imagem com o fundo retirado pelo histograma. Aqui são chamados métodos `gerarHistograma()` e `processarImagemCinza()`.

Após o término da separação, é contado o número de pixels na vertical de cada plântula e feita uma conversão dos pixels para centímetros.

Na Figura 4.12 pode-se ver a imagem de tons de cinza agora somente nas cores preto e branco. Aqui são chamados métodos `pintarBranco()` e `retirarRuidoImagem()`, sendo que o primeiro converte todos os pixels, que não são pretos, para a cor branca, deixando mais fácil o processamento da medida em cm, porque o sistema tem de analisar somente 2 cores. O

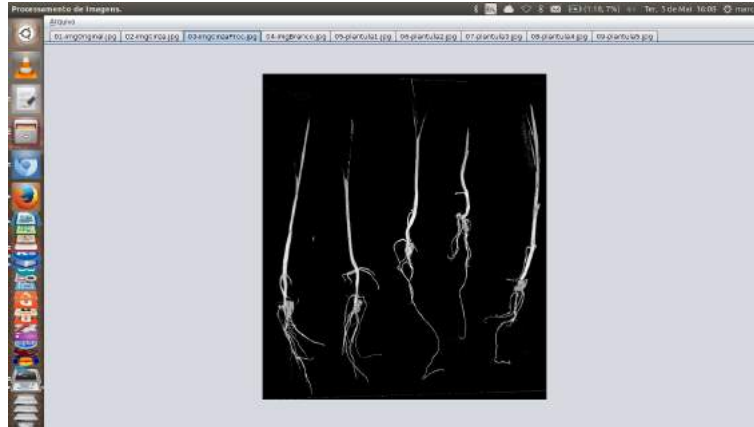


Figura 4.11: Imagem que foi retirada o fundo através da análise do histograma.

outro método é usado para retirar pequenos pontos brancos que poderiam ser confundidos com plântulas, também é retirado nessa parte do processamento, pontos brancos muito pequenos, que podem indicar a sujeira que o método de suavização filtro de média, não conseguiu detectar.



Figura 4.12: Imagem suavizada e transformada para preto e branco.

As Figuras 4.13, 4.14, e 4.15 mostram as plântulas separadas e prontas para serem processadas para medir o seu tamanho. O método criado para fazer a separação das plântulas e com um algoritmo melhor do que o da Fase 1 é o mais extenso e complicado que foi criado no sistema inteiro. Eles estão nos métodos `separarPlantulas()` e `recortarParteImagem(Rectangle rect, int numPlantula)`. Este é utilizado somente para depois de achada as coordenadas, onde está a plântula, ela seja recordada de acordo com os valores de `rect` que é um retângulo. O outro método (`separaPlantulas()`), é o principal, separa a plântula criando uma lista e vai enfileirando os pixels até percorrer todos os pixels da imagem. Quando a plântula for separada e ficar alguns pontos brancos que não pertencem a ela, estes são trocadas para a cor preta, deixando na cor branca somente os pixels que pertencem à plântula.

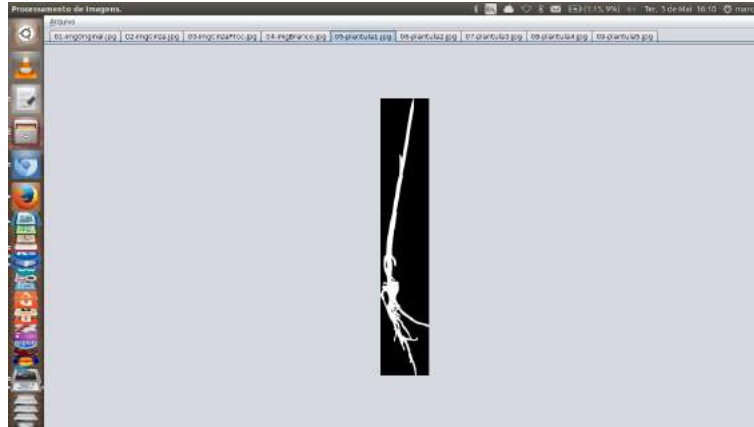


Figura 4.13: Primeira plântula separada.



Figura 4.14: Segunda e terceira plântula separada.

4.2.2 Fase 3 - A opção calibrar e como foi realizada a medição em centímetros.

Nos testes com várias imagens geradas com a *webcam* e um celular, percebeu que a retirada do fundo da imagem pelo histograma, deixava algumas partes do fundo que não eram retiradas e poderiam ser corrigidas com uma interferência manual. Na tarefa de realizar a medida do tamanho da plântula em cm pelo sistema, foi colocado um quadrado vermelho de 1 cm que serve como guia para medir o tamanho da plântula. A detecção da cor vermelha não foi perfeita porque a imagem gerada pela *webcam*, apresentava pontos de outras cores dentro do quadrado. Estes dois problemas foram corrigidos gerando um calibrador como mostra a Figura 4.16 e 4.17. A primeira opção do calibrador refere-se ao fundo do histograma que nessa tela oferece a opção para mudar o valor de limite do histograma, e pode ser visualizado em tempo real no sistema. Para o ponto vermelho, foi usada a mesma técnica do programa da fase 1 que é desenvolvido na linguagem C, sendo selecionados pontos da imagem para que sejam isolados do resto da imagem. Isso é feito em relação ao quadrado vermelho na imagem. As calibrações feitas nessa tela são gravadas em um arquivo chamado `config.properties`, que é salvo no padrão `ascii` e é mostrado o seu conteúdo no anexo D. As informações nele gravadas são os valores `rbg` com limites superior(`r2,g2,b2`) e inferior(`r1,g1,b1`); o valor de correção do histograma e, finalmente a quantidades de pixels que equivalem 1cm na imagem.



Figura 4.15: Quarta e quinta plântula separada.

Para fazer a medida da plântula em centímetros o processo é realizado em cada imagem que já foi separada em arquivos de imagens. O arquivo é aberto e é passado linha por linha até encontrar um pixel branco. Depois disso o sistema vai para próxima linha procurando um pixel branco na mesma coluna que foi encontrada a cor branca na linha anterior. Isso é repetido até a última linha, depois são contados quantas linhas tem a cor branca e, como cada linha é a medida de um pixel, é realizado uma regra de três simples e verificando-se o tamanho da plântula.



Figura 4.16: Opção do sistema para calibrar o histograma.

A caixa de isopor apresentou problemas com a luminosidade interior, porque se colocar uma fonte de luz muito forte vai gerar um foco de luz na imagem, fazendo com que o processamento da imagem precise desse tratamento. Por isso, optou-se por usar um caixa de plástico que deixa a luz natural passar por ela e fica com uma iluminação bem melhor.

Utilizou-se a linguagem Java como ideia inicial, por ela ser amplamente difundida na área, mas não chegou a ser desenvolvida porque foi encontrado um programa na linguagem C que demonstrou ter uma solução para os objetivos do trabalho. No entanto, ao desenvolver a primeira etapa, a linguagem C apresentou impasses, em especial a falta de tempo e o pouco conhecimento de recursos dela para terminar o programa. Diante disso, a solução encontrada foi retornar à Java, linguagem com a qual o sistema foi terminado. Nessa etapa,

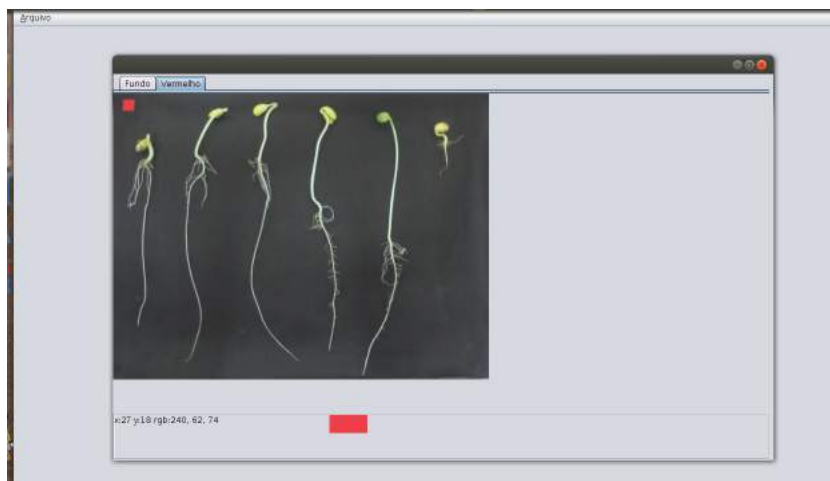


Figura 4.17: Opção do sistema para calibrar a cor vermelha da legenda.

as soluções contaram com o uso de *frameworks ImageJ* e *Swing* que melhoraram o tratamento de imagens e a interface com o usuário do programa.

Capítulo 5

Conclusão

Este capítulo relata as dificuldades encontradas durante o desenvolvimento do sistema, resultados obtidos e perspectivas de trabalhos futuros decorrentes desta pesquisa.

5.1 Dificuldades Encontradas

Durante do desenvolvimento do software e da caixa onde são capturadas as imagens, foram detectadas algumas dificuldades. Com a caixa o maior problema foi a questão da luminosidade e a manipulação das plântulas dentro da caixa que por ter um espaço pequeno não deixava esticar as plântulas de forma que fiquem bem retas para fazer uma medida mais precisa. O posicionamento da *webcam* em uma altura que consiga pegar todas as plântulas e somente a região escura(preta) onde são elas são colocadas ficou difícil porque a caixa tem um tamanho pré-definido se as plântulas são grandes acabam passando dos limites e apresentando uma curva que dificulta a a medida. Com o software as dificuldades foram maiores e abrangem desde que tipo de técnicas seriam usadas para o tratamento das imagens, como também a linguagem de programação a ser utilizada e os problemas normais de desenvolvimento. A linguagem que deu resultados práticos nas primeiras tentativas foi a linguagem C por usar um programa já pronto, mas quando precisou fazer uma interface melhor e processar os pixels da imagem esta linguagem tem resultados muito demorados. Com base em um curso do *framework imagej*, percebi que em java teria mais condições de desenvolver uma melhor interface e teria ajuda para usar técnicas de processamento de imagens. Na primeira fase de desenvolvimento buscamos retirar o fundo da imagem com histograma e os resultados foram muito bons, mas não perfeitos, sendo que a imagem apresenta pontos mais claros e no processamento da imagem isto pode era entendido como uma plântula. Outro problema foi com a legenda vermelha, colocada para identificar o tamanho dos pixels em centímetros e depois usada para calcular o tamanho da plântula, que não ficava toda vermelha, apareciam pontos pretos que dificultavam calcular o tamanho da legenda. Para a correção destes problemas (retirado do fundo e a detecção da legenda) foram criadas opções no sistema para calibrar estes itens, o histograma é calibrado com uma tela onde é mostrado o valor de correção pelo histograma e quando ele é alterado é mostrado na imagem o que o valor faz na imagem, deixando assim visivelmente o resultado. Este valor pode ser gravado em um arquivo texto que depois é lido pela parte que processa a imagem.

Isto também é feito com a legenda vermelha que é mostrada a imagem com ela e depois é guardado o valor que representa a cor vermelha. Uma melhora desta técnica é formar um retângulo envolvendo a legenda para que todas as cores dentro dele sejam consideradas na hora de detectar a legenda.

5.2 Resultados e Perspectivas

No mercado atual existem softwares comerciais que fazem o processamento de imagens para verificar o vigor de plântulas, mas com valores que inviabilizam sua aquisição para pesquisas em instituições públicas. A proposta aqui apresentada é a criação de um conjunto de hardware de baixo custo e software gratuito, para ser usado em pesquisas na área da agricultura que otimizam a análise de vigor das sementes. O funcionamento básico descrito anteriormente é que, por meio de uma câmera, as imagens sejam capturadas e processadas por um sistema desenvolvido na linguagem java com os framework swing e a biblioteca ImageJ da mesma linguagem, em que a imagem capturada passa por processos de retirada de fundo, limpeza de ruídos e termina com a separação das imagens das plântulas, indicando, posteriormente, se ela tem um alto vigor ou não, pelo seu comprimento medido em centímetros. Alguns fundos de imagens com cores diferenciadas foram testados, tais como, a verde, a branca, a azul e a preta, sendo que, para as sementes de milho e soja, a cor mais satisfatória de fundo foi a preta, uma vez que ela possibilitou melhor apresentação na separação entre as imagens das plântulas. Os resultados obtidos pelo protótipo foram promissores, uma vez que foi possível o processamento adequado das imagens das plântulas e a obtenção de medidas compatíveis com as realizadas manualmente. No entanto, novos testes serão realizados com o objetivo de estabelecer as relações entre o comprimento da plântula aferido pelo sistema e o seu vigor. Com isso, pretende-se incorporar dados e informações que permitam a produção de resultados mais detalhadas pelo sistema.

Ressalta-se que a qualidade das imagens utilizadas nos testes realizados até o momento pode ser melhorada com o uso de uma câmera mais moderna e com mais recursos. Dessa forma, imagens com maior resolução possibilitariam resultados mais satisfatórios no processamento das imagens das plântulas e nos resultados produzidos pelo sistema.

O sistema proposto pode tornar mais rápido e eficaz o procedimento de medição do comprimento de plântulas para determinação do vigor de sementes. Para verificação efetiva da rapidez e eficácia serão realizados testes em situações reais através de laboratórios da embrapa. Após os testes efetivos, será submetido o pedido de patente do produto desenvolvido nesta pesquisa.

O baixo custo é outro fator que pode ser apresentado como uma das vantagens do sistema. A interface simples e de fácil entendimento, possibilita a utilização por pesquisadores de diversas áreas de conhecimento, sem a necessidade de um treinamento específico.

A partir dos resultados dos testes a serem realizados pelo novo método de aferição, será possível a comparação com método manual com relação à rapidez, à eficiência e à eficácia. Adicionalmente, com a ajuda de especialista da área de sementes, será possível, a partir das imagens, extrair e utilizar outras informações sobre as sementes, bem como segmentar outras partes das plântulas, possibilitando a obtenção de informações morfológicas mais precisas e

que podem ser utilizadas em outras técnicas e estudos. Outra pesquisa que pode derivar da técnica apresentada por este trabalho, está relacionada com a melhoria das técnicas de segmentação da imagem para a extração de partes específicas das plântulas, em razão da variação de cores e luminosidade, melhorando a aquisição das informações e viabilizando outras aplicações nesta área.

Outra contribuição do trabalho é a técnica implementada para a extração do contorno das sementes, o que possibilitou a determinação do comprimento da plântula. Os testes de velocidade da análise mostraram que o processamento da imagem com um processador i3, com 4 Gb de RAM, 64 bits com o sistema operacional Linux Ubuntu, ficaram próximo de 10 segundos para uma imagem contendo 5 (cinco) plântulas, indicando que são bem mais rápidos que os testes manuais. Isto gera uma economia de tempo em pesquisas nesta área, agilizando e reduzindo custos do processo, além de aumentar a precisão dos dados coletados em relação ao método manual. Como trabalhos futuros, pretende-se realizar estudos sobre técnicas de processamento de imagens e classificação mais modernas que possam ser utilizadas para o refinamento do software e, além disso, realizar testes de vigor complementares, obtendo resultados que servirão como base para a busca de padrões e relações mais diretas e imediatas entre o comprimento obtido pelo sistema proposto e o vigor da semente. Com isso, pretende-se agregar valor ao sistema proposto, possibilitando ainda mais agilidade no procedimento de medição realizado durante os testes de vigor.

Referências Bibliográficas

- [1] CONCI, A.; AZEVEDO, E.; LETA, F. R. *Computação Gráfica*. [S.l.]: Editora Campus, 2008.
- [2] 12G turma. *Teoria da cor - 2.3 Modelo RGB - Resolução e Tamanho - Profundidade de cor*: Site. 2014. Disponível em: <<http://teoriadacor12g.blogspot.com.br/2011/02/23-modelo-rgb-resolucao-e-tamanho.html>>. Acesso em: 20/10/2014.
- [3] FALCAO, A. X. *Resolução espacial e Profundidade da imagem*: Site. 2014. Disponível em: <<http://www.ic.unicamp.br/~cpg/material-didatico/mo815/9802/curso/node7.html>>. Acesso em: 20/10/2014.
- [4] VARELLA, C. A. A. Processamento de imagens digitais na agricultura de precisão. *Universidade Federal Rural do Rio de Janeiro*, 2006.
- [5] PISTORI, H. *LIMIARIZACAO*: Site. 2014. Disponível em: <<https://www.youtube.com/watch?v=ctKa5ppxAK0>>. Acesso em: Março. 2014.
- [6] CARVALHO, N. M. de; NAKAGAWA, J. *Sementes: ciência, tecnologia e produção*. [S.l.]: FUNEP, 2012.
- [7] CASTRO, M. B. D. *Avaliação da qualidade fisiológica de sementes de milho por meio da atividade respiratória*. Tese (Doutorado) — Universidade de Lavras, 2011.
- [8] AOSA, A. Seed vigor testing handbook. *Revista Brasileira de Sementes*, 1983.
- [9] MENDES, C. *Notícias Agrícolas*: Site. 2016. Disponível em: <<http://www.noticiasagricolas.com.br/noticias/usda/155979-usda-reduz-estoques-de-soja-abaixo-a-expectativa-media-e-aumenta-os-de-milho-nos-eua.html#.VpEL6VHoeb4>>. Acesso em: Janeiro. 2016.
- [10] JUNQUEIRA, R. A. R.; MORABITO, R. Planejamento otimizado da produção e logística de empresas produtoras de sementes de milho: Um estudo de caso. *Universidade Federal de São Carlos/UFSCAR, São Carlos, SP.*, 2006.
- [11] MENEZES, N. L. de. *Testes Rápidos Para A Determinação Da Qualidade Das Sementes*: Site. 2015. Disponível em: <<http://coral.ufsm.br/sementes/index.php/component/content/article/2-uncategorised/12-testes-rapidos>>. Acesso em: 10/01/2015.

- [12] COSTA, C. J. *A Importância Do Controle De Qualidade De Sementes.*: Site. 2013. Disponível em: <<http://www.cpac.embrapa.br/download/1445/t>>. Acesso em: Novembro. 2013.
- [13] PERES, W. L. R. Testes de vigor em sementes de milho. 2010.
- [14] OLIVEIRA, A. C. S. et al. Testes de vigor em sementes baseados no desempenho de plântulas. *Revista Científica Internacional*, 2009.
- [15] TEIXEIRA, E. F. Análise de imagens digitais na avaliação de plântulas de milho. *ESALQ/USP*, 2004.
- [16] LIMA, E. L. L. R. Daniel Caio de; JORGE, L. A. de C. A new approach to automate the seed vigor analysis on soybean seedling using digital images. *Proceedings of X Workshop de Visão Computacional WVC 2014*, 2014.
- [17] VEIRA, G. H. M. Vigor em semente. 2011.
- [18] MCDONALD, M. A review and evaluation of seed vigor tests. *Proceedings of the Association of the Official Seed Analysts*, v. 65, n. 1, p. 109–139, 1975.
- [19] DUTRA, A. S.; DAITON, V. R. Envelhecimento acelerado como teste de vigor para sementes de milho e soja. *Centro de Ciências Rurais - Universidade Federal de Santa Maria*, 2004.
- [20] BERTOLIN, D. C.; SÁ, M. E. de; MOREIRA, E. R. Teste de envelhecimento acelerado para sementes de feijão. *Revista Brasileira de Sementes*, p. 104–112, 2011.
- [21] CUSTÓDIO, C. C. Testes rápidos para avaliação do vigor de sementes: uma revisão. *Faculdade de Ciências Agrárias da UNOESTE*, v. 1, 2005.
- [22] BROUWER, H.; MULDER, J. Reduced steeping time for the conductivity vigor test of *Phaseolus vulgaris* L. seed. *Journal of Seed Technology*, v. 7, n. 1, p. 84–96, 1982.
- [23] FILHO, J. M.; SILVA, W. d. Estudo comparativo de métodos para avaliação da qualidade fisiológica de sementes de soja, com ênfase ao teste de condutividade elétrica. *Pesquisa Agropecuária Brasileira*, v. 25, n. 12, p. 1805–1815, 1990.
- [24] NAKAGAWA, J. Testes de vigor baseados no desempenho das plântulas. *ABRATES*, 1999.
- [25] GONZALEZ, R. c.; WOODS, R. E. *Processamento de imagens digitais*. [S.l.]: Editora Edgard Blucher Ltda, 2008.
- [26] GOMES, O. D. F. M. *Processamento e Análise de Imagens Aplicados à Caracterização Automática de Materiais*. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio de Janeiro, 2001.
- [27] HONDA, B.; JORGE, L. A. C. Computação aplicada à agricultura de precisão. *Científica Eletrônica UNISEB.*, v. 1, p. 111–132, 2013.

Apêndice A

Código fonte do programa de análise por limiarização alterado

```
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/nonfree/nonfree.hpp>
#include <opencv2/opencv.hpp> // Header File
#include "cv.h"
#include "cxcore.h"
#include "math.h"
#include <iostream>
#include <fstream>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sstream>
#include <time.h>
#include <sys/stat.h>
#include <pthread.h>
#include <errno.h>
```

```
using namespace cv;
using namespace std;
```

```
//-----
//--- Global Variables
```

```
const char configFileName[] = "../data/config.xml";
const char outFileName[] = "./plantula\\_seg.jpg";
```

```

int bmin = 0; // Minimum value for BLUE channel thresholding (BGR)
int bmax = 20; // Maximum value for BLUE channel thresholding (BGR)
int gmin = 30; // Minimum value for GREEN thresholding
int gmax = 150; // Maximum ...
int rmin = 80; // Minimum value for RED thresholding
int rmax = 255; // Maximum ...
int useMinMax = 1; // Instead of using mean and standard dev, use min and max
values
int stdMult = 4; // Standard deviation multiplier used to calculate the
threshold range (RANGE SIZE)

// Initial position for all the windows (adjusted to a 1920x1080 screen). The
values are read from the configuration file at configFileName
int webcamX, webcamY, wthresholdX, wthresholdY, wcontrolX, wcontrolY = 0;

// Selection of sample pixels that will be used to determine the BGR min a max
values
bool selectObject = false;
// Origin of the sample pixels window (dynamically created by the user using
press and release mouse button actions)
Point origin;
// Sample pixels window
Rect selection;

// Accumulator for statistics calculation
int Mean[3] = {0,0,0}; // Mean values for B, G and R converted to int
int Std[3] = {0,0,0}; // Standard Deviation values for B, G and R converted to int
int Min[3] = {0,0,0}; // Minimum values for B, G and R converted to int
int Max[3] = {0,0,0}; // Maximum values for B, G and R converted to int

IplImage* originalImage;
IplImage* thresholdedImage;

string diretorio = "./imagens";

//-----
//--- Functions Declarations
void saveConfiguration(void);
void readConfiguration(void);
void changeBGRMinMaxValues(Mat roi);
void statisticsChanged( int, void * );
void onMouse( int event, int x, int y, int, void* );

```

```

void createAllWindows(void);
void on\_button(int, void*);
void separateImage(int primPixel, int ultPixel, int imagemCriada);
void saveImage(char* outFileName, IplImage* Imagem);

//-----
//--- Main

int main( int argc, char** argv )
{
    printf("Numero de argumentos inválidos %d",argc);
    if (argc==1) {
        printf("Numero de argumentos inválidos %d",argc);
        exit(0);
    }
    readConfiguration();

//-----
//--- Create windows e move to saved position (configuration file -
data/config.xml)

    cvNamedWindow( "originalImage",1); // Window for the original frames
captured by the WebCam
    setMouseCallback( "originalImage", onMouse, 0 );
    cvMoveWindow("originalImage",400,0);

    cvNamedWindow( "thresholdedImage",2); // Window for the thrholded image
    cvMoveWindow("thresholdedImage", 0 , 0);

    namedWindow( "Configuration", 0); // Window for the trackbars with
adjustable parameters
    cvMoveWindow("Configuration",800,wcontrolY);

    createAllWindows();

    int c = 0;
    {

        // Get the current frame from the webcam
        //originalImage = cvQueryFrame( capture);
        originalImage = cvLoadImage( argv[1]);
    }
}

```

```

if(!originalImage){
    printf("Could not load image file: %s\n",argv[1]);
    exit(0);
}

//-----
//--- Apply Color Thresholding

    thresholdedImage = cvCreateImage(cvGetSize(originalImage),
                                     originalImage->depth,originalImage->nChannels);
    cvCopy(originalImage,thresholdedImage,NULL);

int imagemCriada = 1;
    // Iterate through all the pixels of the image (one by one)
    uchar * data = (uchar *)thresholdedImage->imageData;
    int step = thresholdedImage->widthStep;
    int channels = thresholdedImage->nChannels;
    for (int row=0; row<thresholdedImage->height; row++) {
        for (int col=0; col<thresholdedImage->width; col++) {
            int B = data[row*step+col*channels+0]; //Pixel's Blue (B) value
            int G = data[row*step+col*channels+1]; //Pixel's Green (G) value
            int R = data[row*step+col*channels+2]; //Pixel's Red (R) value
            if( !( B >= bmin && B <= bmax  && G >= gmin && G <= gmax  &&
                R >= rmin && R <= rmax)) {
                for (int channel=0; channel<channels; channel++) {
                    data[row*step+col*channels+channel]=0;
                }
            }
        }
    }

int primPixel = -1;
int ultPixel = -1;
int contCor = 0;
bool tudoPreto = true;
//varre a imagem procura colunas todas pretas, indicando que
    for (int col=0; col<thresholdedImage->width; col++) {
        for (int row=0; row<thresholdedImage->height; row++) {
            int B = data[row*step+col*channels+0];
            int G = data[row*step+col*channels+1];
            int R = data[row*step+col*channels+2];
            if( B != 0 && G != 0 && R !=0 ) {
                tudoPreto = false;
            }
        }
    }

```

```

    }
}

if (!tudoPreto) {
    if (primPixel == -1) {
        primPixel = col;
    }
    contCor++;
} else {
    if (primPixel != -1 && ultPixel == -1) {
        ultPixel = col;
        printf("contador pixel:%d \n", contCor);
        if (contCor > 5) {
            separateImage(primPixel, ultPixel, imagemCriada);
            imagemCriada++;
        }
        primPixel = -1;
        ultPixel = -1;
        contCor = 0;
    }
}
tudoPreto = true;
}

if ( selectObject && selection.width > 0 && selection.height > 0 )
{
    Mat roi(originalImage, selection);
    changeBGRMinMaxValues(roi);
    bitwise_not(roi, roi); // Shows that rectangular region indicating
                           what you selected using mouse dragging
    cvResetImageROI(originalImage);
}

if(!cvSaveImage( outFileName, thresholdedImage))
    printf("Could not save: %s\n",outFileName);

cvShowImage("originalImage", originalImage); // Show the webcam image
cvMoveWindow("thresholdedImage",wthresholdX,wthresholdY);
cvShowImage("thresholdedImage", thresholdedImage); // Show thresholded

cvReleaseImage(&thresholdedImage );
}

while ( c != 27) c = cvWaitKey(30);

saveConfiguration();

```

```

        cvDestroyAllWindows();

    }

    string dataHoraAtual() {
        struct tm *local;
        time_t t;
        string retorno;

        t= time(NULL);
        local=localtime(&t);

        int dia=local->tm\_mday;
        int mes=local->tm\_mon+1;
        int ano=local->tm\_year+1900;
        int hora=local->tm\_hour;
        int minuto=local->tm\_min;

        /* Converte inteiro em string */
        std::stringstream Resultado;
        Resultado << dia;
        Resultado << mes;
        Resultado << ano;
        Resultado << hora;
        Resultado << minuto;

        return Resultado.str();
    }
    //-----
    // Processa a imagem do intervalo passado como parametro
    void separateImage(int primPixel, int ultPixel, int imagemCriada) {
        IplImage* Image1;
        //String nome = "imagem" + imagemCriada;
        string nome = "imagem" ;

        /* Converte inteiro em string */
        std::stringstream Resultado;
        Resultado << imagemCriada;

        nome = diretorio + dataHoraAtual().c\_str() + "/" + nome
            + Resultado.str() + ".jpg";

        Image1 = cvCreateImage(cvGetSize(thresholdedImage),
                                thresholdedImage->depth,thresholdedImage->nChannels);
        cvCopy(thresholdedImage,Image1, NULL);
        int step = Image1->widthStep;

```

```

int channels = Image1->nChannels;
uchar * data = (uchar *)Image1->imageData;
for (int col=0; col<Image1->width; col++) {
    for (int row=0; row<Image1->height; row++) {
        if (col >= primPixel && col <= ultPixel) {
            /*for (int channel=0; channel<channels; channel++) {
                if (channel !=0)
                    data[row*step+col*channels+channel]=0;
            }*/
        } else {
            for (int channel=0; channel<channels; channel++) {
                data[row*step+col*channels+channel]=0;
            }
        }
    }
}

char* cstr = new char[nome.length() +1];
strcpy(cstr, nome.c\_str());
saveImage(cstr, Image1);
delete[] cstr;

//procurar a primeira linha que não é totalmente preta
//ir até o fim da coluna que tem cor
int inicio=0, fim=0, linhaInicio;
//separar uma plantula
if (imagemCriada == 1) {
    printf("entrei no imagemcriada %d",imagemCriada);
    for (int row=0; row<Image1->height; row++) {
        for (int col=0; col<Image1->width; col++) {
            if (fim ==0) {
                int B = data[row*step+col*channels+0];
                int G = data[row*step+col*channels+1];
                int R = data[row*step+col*channels+2];
                if ( B != 0 && G != 0 && R !=0 ) {
                    inicio = col;
                    while ( B != 0 && G != 0 && R !=0 ) {
                        col++;
                        int B = data[row*step+col*channels+0];
                        int G = data[row*step+col*channels+1];
                        int R = data[row*step+col*channels+2];
                    }
                    fim = col;
                    linhaInicio = row;
                }
            } else {
                printf("=== inicio %d fim %d ",inicio, fim);
                for (int channel=0; channel<channels; channel++) {

```

```

        data[row*step+col*channels+channel]=0;
    }
}
}
    if (fim > 0) { break; }
}
printf("inicio %d fim %d", inicio, fim);
int col = inicio; //a coluna da proxima linha é a primeira coluna
                //onde foi achado o pixel diferente de preto
for (int row=linhaInicio; row<Image1->height; row++) {
    int B = data[row*step+col*channels+0];
    int G = data[row*step+col*channels+1];
    int R = data[row*step+col*channels+2];
    while ( B != 0 && G != 0 && R !=0 ) {
        col--;
        int B = data[row*step+col*channels+0];
        int G = data[row*step+col*channels+1];
        int R = data[row*step+col*channels+2];
    }
    inicio = col;
    while ( B != 0 && G != 0 && R !=0 ) {
        col++;
        int B = data[row*step+col*channels+0];
        int G = data[row*step+col*channels+1];
        int R = data[row*step+col*channels+2];
    }
    fim = col;
    for (col=fim; col<Image1->width; col++) {
        for (int channel=0; channel<channels; channel++) {
            data[row*step+col*channels+channel]=0;
        }
        data[row*step+col*channels+2]=255;
    }
}

}

cvNamedWindow( nome.c\_str(), 2); // Window for the thrholded image
cvMoveWindow(nome.c\_str(), 50+imagemCriada *100 , imagemCriada * 80);
cvShowImage(nome.c\_str(), Image1); // Show thresholded Image

cvReleaseImage(&Image1 );
}

//-----

void criarDiretorio(std::string nome) {

```

```

struct stat st;
if(stat(nome.c\_str(),\&st) != 0)
{
mkdir(nome.c\_str(), S\_IRWXU);
}
else
{
std::cout << "Diretorio " << nome.c\_str() << " existe \n";
}
}

// Salvar imagem em disco
void saveImage(char* outFileName, IplImage* Imagem) {
    //Verificar se existe o diretorio
    //criar o diretorio se necessário
    criarDiretorio( diretorio + dataHoraAtual().c\_str() + "/" );
    if(cvSaveImage( outFileName, Imagem)) {
        printf("imagem salva:%s \n", outFileName);
    } else {
        printf("Could not save: %s\n",outFileName);
    }
}

//-----
// Calculate HSV Min and Max from statistics extracted from the image
void adjustBGRMinMax(void) {
    if (useMinMax) {

        bmin = Min[0];
        bmax = Max[0];
        gmin = Min[1];
        gmax = Max[1];
        rmin = Min[2];
        rmax = Max[2];
    }
    else {

        bmin = Mean[0]-Std[0]*stdMult;
        bmax = Mean[0]+Std[0]*stdMult;
        gmin = Mean[1]-Std[1]*stdMult;
        gmax = Mean[1]+Std[1]*stdMult;
        rmin = Mean[2]-Std[2]*stdMult;
        rmax = Mean[2]+Std[2]*stdMult;
    }
    setTrackbarPos( "Bmin", "Configuration", bmin );
    setTrackbarPos( "Bmax", "Configuration", bmax );
}

```

```

        setTrackbarPos( "Gmin", "Configuration", gmin );
        setTrackbarPos( "Gmax", "Configuration", gmax );
        setTrackbarPos( "Rmin", "Configuration", rmin );
        setTrackbarPos( "Rmax", "Configuration", rmax );
    }

//-----
// Extract min, max, mean and standard deviation statistics from the selected
//window
void changeBGRMinMaxValues(Mat roi) {

    Mat BGRMeans(4,1,CV_32F);
    Mat BGRStdDevs(4,1,CV_32F);
    double BGRMins[] = {0,0,0,0};
    double BGRMaxs[] = {0,0,0,0};
    Mat channel;

    // Calculate means and standard values for B, G and R channels
    meanStdDev(roi,BGRMeans,BGRStdDevs);

    for (int i=0; i<3; ++i) {

        Mean[i] = (int) BGRMeans.at<double>(i,0);
        Std[i] = (int) BGRStdDevs.at<double>(i,0);

        extractChannel(roi,channel,i);
        minMaxIdx(channel,&BGRMins[i],&BGRMaxs[i]);

        Min[i] = BGRMins[i];
        Max[i] = BGRMaxs[i];
    }

    adjustBGRMinMax();
}

//-----
// The statistics must be calculated changed and the values must be recalculated
void statisticsChanged( int, void * ) {
    if(originalImage) {

```

```

        Mat roi(originalImage, selection);
        changeBGRMinMaxValues(roi);
        cvResetImageROI(originalImage);
    }
}

//-----
// Deals with the selection of a rectangular region from where the mean and
//standard deviation statistics for pixels BGR values are calculated
void onMouse( int event, int x, int y, int, void* )
{
    if ( selectObject )
    {
        selection.x = MIN(x, origin.x);
        selection.y = MIN(y, origin.y);
        selection.width = std::abs(x - origin.x);
        selection.height = std::abs(y - origin.y);
        selection |= Rect(0, 0, 650, 500);

    }

    switch ( event )
    {
    case CV_EVENT_LBUTTONDOWN:
        origin = Point(x,y);
        selection = Rect(x,y,0,0);
        selectObject = true;
        break;
    case CV_EVENT_LBUTTONUP:
        selectObject = false;
        break;
    }
}

//-----

void createAllWindows(void) {
    createTrackbar( "Bmin", "Configuration", &bmin, 256, 0 );
    createTrackbar( "Bmax", "Configuration", &bmax, 256, 0 );
    createTrackbar( "Gmin", "Configuration", &gmin, 256, 0 );
    createTrackbar( "Gmax", "Configuration", &gmax, 256, 0 );
    createTrackbar( "Rmin", "Configuration", &rmin, 256, 0 );
    createTrackbar( "Rmax", "Configuration", &rmax, 256, 0 );
    createTrackbar( "Use Min Max ?", "Configuration", &useMinMax, 1,
        statisticsChanged );
}

```

```

        createTrackbar( "Standard Deviation", "Configuration", \&stdMult, 20,
                        statisticsChanged );
    }

void on\_button(int, void*){
    cout << "click" << endl;
}

//-----
// Save configuration parameters on disk
void saveConfiguration(void) {
    CvFileStorage* fs=cvOpenFileStorage(configFileName, 0,CV\_STORAGE\_WRITE);
    cvWriteInt( fs, "bmin", bmin );
    cvWriteInt( fs, "bmax", bmax );
    cvWriteInt( fs, "gmin", gmin );
    cvWriteInt( fs, "gmax", gmax );
    cvWriteInt( fs, "rmin", rmin );
    cvWriteInt( fs, "rmax", rmax );
    cvWriteInt( fs, "wwebcamX", wwebcamX);
    cvWriteInt( fs, "wwebcamY", wwebcamY);
    cvWriteInt( fs, "wthresholdX", wthresholdX);
    cvWriteInt( fs, "wthresholdY", wthresholdY);
    cvWriteInt( fs, "wcontrolX", wcontrolX);
    cvWriteInt( fs, "wcontrolY", wcontrolY);
    cvWriteInt( fs, "stdMult", stdMult);
    cvWriteInt( fs, "useMinMax", useMinMax);
    cvReleaseFileStorage(\&fs);
}

//-----
// Recover configuration parameters from disk
void readConfiguration(void) {
    CvFileStorage* fs= cvOpenFileStorage(configFileName, 0, CV\_STORAGE\_READ);
    if (fs) {
        bmin = cvReadIntByName( fs, 0, "bmin" );
        bmax = cvReadIntByName( fs, 0, "bmax" );
        gmin = cvReadIntByName( fs, 0, "gmin" );
        gmax = cvReadIntByName( fs, 0, "gmax" );
        rmin = cvReadIntByName( fs, 0, "rmin" );
        rmax = cvReadIntByName( fs, 0, "rmax" );
        wwebcamX = cvReadIntByName( fs, 0, "wwebcamX");
        wwebcamY = cvReadIntByName( fs, 0,"wwebcamY");
        wthresholdX = cvReadIntByName( fs,0, "wthresholdX");
    }
}

```

```
wthresholdY = cvReadIntByName( fs,0, "wthresholdY");
wcontrolX = cvReadIntByName( fs,0, "wcontrolX");
wcontrolY = cvReadIntByName( fs,0, "wcontrolY");
stdMult = cvReadIntByName( fs,0, "stdMult");
useMinMax = cvReadIntByName( fs,0, "useMinMax");
Mean[0] = (bmin+bmax)/2;
Mean[1] = (gmin+gmax)/2;
Mean[2] = (rmin+rmax)/2;
Std[0] = ((bmax-bmin)/2)/stdMult;
Std[1] = ((gmax-gmin)/2)/stdMult;
Std[2] = ((rmax-rmin)/2)/stdMult;
Min[0] = bmin;
Min[1] = gmin;
Min[2] = rmin;
Max[0] = bmax;
Max[1] = gmax;
Max[2] = rmax;
}
cvReleaseFileStorage( \&fs);
}
```

Apêndice B

Arquivo de informações das plântulas geradas pelo sistema

Nome da imagem: /home/marcos/Documentos/Ifms/mestradoUFMS/dissertação/
imagens2015/soja1.jpg

Quantidade de Plântulas:8

Plântula :1
TotalPixels:133305
x:716
y:0
largura:83
altura:479

Plântula :2
TotalPixels:11523
x:600
y:144
largura:50
altura:288

Plântula :3
TotalPixels:9859
x:470
y:161
largura:51
altura:168

Plântula :4
TotalPixels:21964
x:316
y:117
largura:124

altura:273

Plântula :5
TotalPixels:16981
x:224
y:121
largura:78
altura:279

Plântula :6
TotalPixels:20663
x:205
y:59
largura:258
altura:28

Plântula :7
TotalPixels:8805
x:118
y:145
largura:78
altura:173

Plântula :8
TotalPixels:4041
x:0
y:237
largura:10
altura:145

Apêndice C

Principais métodos da classe Processar Imagem

```
public ProcessarImagem(String nomeArquivo) {
    this.nomeArquivo = nomeArquivo;
    if (!(new File(nomeArquivo).exists())) {
        System.out.println("Arquivo não existe.");
        System.exit(0);
    }
    File file = new File(nomeArquivo);
    niArquivo = file.getName();
    niDiretorio = nomeArquivo.substring(0, nomeArquivo.indexOf(file.getName()));
    niNomeSemExt = niArquivo.substring(0, niArquivo.indexOf("."));
    niNovoDiretorio = niDiretorio + niNomeSemExt;

    img = new ImagePlus(nomeArquivo);
    gerarArquivoImagem(img, "01-imgOriginal.jpg", niNovoDiretorio);

    ImageConverter imageConverter = new ImageConverter(img);
    ImagePlus img2 = new ImagePlus("Imagem em cinza",
                                    img.getProcessor().duplicate());
    imageConverter.convertToGray8();
    ip = img.getProcessor();
    gerarArquivoCsv("imagemCinza1.csv", ip); //mpv
    gerarHistograma();
}

public void processarImagemCinza() {
    //processar imagem em escalas de cinza
    for (int x = 0; x < larguraImagem(); x++) {
        for (int y = 0; y < alturaImagem(); y++) {
            //Verifica quantos pixels aparecem mais de 1500 vezes
            //Quando a imagem é pequena este valor não funciona
        }
    }
}
```

```

        if (histog[ip.getPixel(x, y)] > HISTOGRAMAFUNDO) {
            ip.putPixel(x, y, 0);
        }
    }
}

private void gerarHistograma() {
    //histograma
    for (int i = 0; i < histog.length; i++) {
        histog[i] = 0;
    }
    for (int x = 0; x < larguraImagem(); x++) {
        for (int y = 0; y < alturaImagem(); y++) {
            //System.out.println("valor " + x + "," + y + " =" + ip.getPixel(x, y));
            histog[ip.getPixel(x, y)]++;
        }
    }
}

```

```

public void retirarRuidoImagem() {
    //retirar ruido da imagem
    // Pedir banco de imagens para sao carlos usp e esalq ups
    double pixelNovo;
    for (int x = 0; x < larguraImagem(); x++) {
        for (int y = 0; y < alturaImagem(); y++) {
            if (x > 1 && y > 1 && x < (larguraImagem() - 1) &&
                y < (alturaImagem() - 1)) {

                pixelNovo = ip.getPixel(x - 1, y - 1) / 9;
                pixelNovo += ip.getPixel(x - 1, y) / 9;
                pixelNovo += ip.getPixel(x - 1, y + 1) / 9;

                pixelNovo += ip.getPixel(x, y - 1) / 9;
                pixelNovo += ip.getPixel(x, y) / 9;
                pixelNovo += ip.getPixel(x, y + 1) / 9;

                pixelNovo += ip.getPixel(x + 1, y - 1) / 9;
                pixelNovo += ip.getPixel(x + 1, y) / 9;
                pixelNovo += ip.getPixel(x + 1, y + 1) / 9;

                ip.putPixel(x, y, (int) pixelNovo);
            }
        }
    }
}

```

```

public void pintarBranco() {
    for (int x = 0; x < larguraImagem(); x++) {
        for (int y = 0; y < alturaImagem(); y++) {
            //System.out.println(ip.getPixel(x, y));
            if (ip.getPixel(x, y) > 0) {
                ip.putPixel(x, y, 255);
            }
        }
    }
}

public void separarPlantulas() {
    ImageProcessor novoIp = ip;
    List lista = new ArrayList();
    List listaPlantulas = new ArrayList();
    List listaSujeira = new ArrayList();
    List pixelsPlantulas = new ArrayList();
    //Colocar cor branca nos pontos que não são pretos(fundo)
    //que já foram processados antes
    for (int x = 0; x < larguraImagem(); x++) {
        for (int y = 0; y < alturaImagem(); y++) {
            //System.out.println(novoIp.getPixel(x, y));
            if (novoIp.getPixel(x, y) > 0) {
                novoIp.putPixel(x, y, VALORBRANCOPROCESSAMENTO);
            }
        }
    }

    int x, y, quantPixels, numVezes = 0;
    x = y = quantPixels = 0;
    while (x < larguraImagem() && y < alturaImagem()) {
        if (novoIp.getPixel(x, y) == VALORBRANCOPROCESSAMENTO) {
            //Verifica se o x ou y iguais a zero para desprezar a primeira
            //linha coluna
            if (x == 0 || y == 0) {
                novoIp.putPixel(x, y, VALORPRETOPROCESSAMENTO);
                continue;
            }
            numVezes++; //Conta quantas vezes achou um ponto branco que faz
                        // parte de uma objeto na imagem
            Point ponto = new Point(x, y);
            lista.add(ponto);
            //System.out.println("x=" + ponto.x + " y=" + ponto.y);
            quantPixels = 1;
        }
        x++;
        if (x == larguraImagem()) {
            y++;
            x = 0;
        }
    }
}

```

```

while (!lista.isEmpty()) {
    ponto = (Point) lista.remove(lista.size() - 1);
    //Muda a cor dos pontos que estão sendo processados para preto
    novoIp.putPixel(ponto.x, ponto.y, VALORPRETOPROCESSAMENTO);
    for (int px = ponto.x - 1; px <= ponto.x + 1; px++) {
        for (int py = ponto.y - 1; py <= ponto.y + 1; py++) {
            if (novoIp.getPixel(px, py) == VALORBRANCOPROCESSAMENTO) {
                Point ponto1 = new Point(px, py);
                lista.add(ponto1);
                //System.out.println("x=" + ponto1.x + " y=" + ponto1.y);
                quantPixels++;
            }
        }
    }
}
}

//for (int py = ponto.y
}

//for (int px = ponto.x
}

//while (!lista.isEmpty())
if (quantPixels > QUANTIDADEPIXELSPLANTULA) {
    listaPlantulas.add(new Point(x, y));
    pixelsPlantulas.add(quantPixels);
} else {
    listaSujeira.add(new Point(x, y));
}

//gerarArquivoCsv("imagem-"+x+"-"+y+ ".csv", ip);
}

//if (novoIp.getPixel(x, y) == 255)

y++;
if (y == alturaImagem()) {
    x++;
    y = 0;
}

}

}

//while (x < larguraImagem() && y < alturaImagem())

//A partir deste ponto do código VALORPRETOPROCESSAMENTO são os valores
//das imagens que o programa achou e já separou em plantulas e sujeira

//gerarArquivoCsv("imagem_ps.csv", novoIp);

System.out.println("Plantulas=" + listaPlantulas.size());
for (Object elem : listaPlantulas) {
    Point ponto = (Point) elem;
    System.out.println("x = " + ponto.x + " y = " + ponto.y);
}

System.out.println("Sujeira=" + listaSujeira.size());
for (Object elem : listaSujeira) {
    Point ponto = (Point) elem;
    System.out.println("x = " + ponto.x + " y = " + ponto.y);
}

}

//Processa todas as plantulas mudando a cor para branco

```

```

// e retirando as coordenadas iniciais e finais na imagem
int px1, py1, px2, py2;
px1 = py1 = px2 = py2 = 0;
while (!listaPlantulas.isEmpty()) {
    Point ponto = (Point) listaPlantulas.remove(listaPlantulas.size() - 1);
    quantPixels = (int) pixelsPlantulas.remove(pixelsPlantulas.size() - 1);
    if (lista.isEmpty()) {
        px1 = ponto.x;
        py1 = ponto.y;
        px2 = px1;
        py2 = py1;
    }
    lista.add(ponto);
    while (!lista.isEmpty()) {
        ponto = (Point) lista.remove(lista.size() - 1);
        if (px1 > ponto.x) {
            px1 = ponto.x;
        }
        if (px2 < ponto.x) {
            px2 = ponto.x;
        }
        if (py1 > ponto.y) {
            py1 = ponto.y;
        }
        if (py2 < ponto.y) {
            py2 = ponto.y;
        }
    }

    //Muda a cor dos pontos que estão sendo processados para preto
    novoIp.putPixel(ponto.x, ponto.y, 255);
    for (int px = ponto.x - 1; px <= ponto.x + 1; px++) {
        for (int py = ponto.y - 1; py <= ponto.y + 1; py++) {
            if (novoIp.getPixel(px, py) == VALORPRETOPROCESSAMENTO){
                Point ponto1 = new Point(px, py);
                lista.add(ponto1);
            }
        }
    }
    Plantulas plantulas = new Plantulas(new Rectangle(px1, py1,
                                                        px2 - px1, py2 - py1), quantPixels);
    retanguloPlantulas.add(plantulas);
}

while (!listaSujeira.isEmpty()) {
    Point ponto = (Point) listaSujeira.remove(listaSujeira.size() - 1);
    lista.add(ponto);
}

```

```

        while (!lista.isEmpty()) {
            ponto = (Point) lista.remove(lista.size() - 1);
            //Muda a cor dos pontos que estão sendo processados para preto
            ip.putPixel(ponto.x, ponto.y, 0);
            for (int px = ponto.x - 1; px <= ponto.x + 1; px++) {
                for (int py = ponto.y - 1; py <= ponto.y + 1; py++) {
                    if (ip.getPixel(px, py) == VALORPRETOPROCESSAMENTO) {
                        Point ponto1 = new Point(px, py);
                        lista.add(ponto1);
                    }
                }
            }
        }
    }
}

//while (!listaPlantulas.isEmpty())
//duplicarMostrarImagem("plantulas brancas");
System.out.println("plantulas brancas");
int numPlant = 1;
//Percorre as plantulas na ordem inversa para salvar na ordem certa
for (int i = retanguloPlantulas.size() - 1; i >= 0; i--) {
    Plantulas object = (Plantulas) retanguloPlantulas.get(i);
    recortarParteImagem(((Plantulas) object).getRectangle(), numPlant);
    numPlant++;
}

}

public void recortarParteImagem(Rectangle rect, int numPlantula) {
    //Corta imagem e mostra
    ip.setRoi(rect);
    ImageProcessor ip2 = ip.crop();
    String title = " (cropped)";
    ImagePlus imp2 = new ImagePlus(title, ip2);
    //imp2.show();

    String NomeImg = (numPlantula + 4) + "-plantula" + numPlantula + ".jpg";
    if ((numPlantula + 4) <= 9) {
        NomeImg = "0" + (numPlantula + 4) + "-plantula" + numPlantula + ".jpg";
    }
    gerarArquivoImagem(imp2, NomeImg, niNovoDiretorio);
    //gerarArquivoCsv(niNovoDiretorio+"/"+NomeImg+".csv", ip);
}

private void gerarArquivoCsv(String nomeArquivo, ImageProcessor ip) {
    if (ip == null) {
        ip = this.ip;
    }
    File arq = new File(nomeArquivo);
}

```

```
BufferedWriter out;
try {
    out = new BufferedWriter(new FileWriter(arq));

    for (int y = 0; y < alturaImagem(); y++) {
        for (int x = 0; x < larguraImagem(); x++) {
            out.write(ip.getPixel(x, y) + ";");
        }
        out.newLine();
    }
    out.close();

} catch (IOException ex) {
    System.out.println("Erro:" + ex.getMessage());
}
System.out.println("Arquivo gravado com sucesso.");
}
```