

UNIVERSIDADE DE PASSO FUNDO
INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM
COMPUTAÇÃO APLICADA

UMA PLATAFORMA DE AQUISIÇÃO E
PROCESSAMENTO DE IMAGENS AÉREAS

Gustavo César Visentini

Passo Fundo

2018

UNIVERSIDADE DE PASSO FUNDO
INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

**UMA PLATAFORMA DE AQUISIÇÃO
E PROCESSAMENTO DE IMAGENS
AÉREAS**

Gustavo César Visentini

Dissertação apresentada como requisito parcial
à obtenção do grau de Mestre em Computação
Aplicada na Universidade de Passo Fundo.

Orientador: Prof. Dr. Willingthon Pavan

Passo Fundo

2018

CIP – Catalogação na Publicação

V829p Visentini, Gustavo César
Uma plataforma de aquisição e processamento de imagens
aéreas / Gustavo César Visentini. – 2018.
78 f. : il. color. ; 30 cm.

Orientador: Prof. Dr. Willingthon Pavan.

Dissertação (Mestrado em Computação Aplicada) –
Universidade de Passo Fundo, 2018.

1. Softwares. 2. Programas de computador. 3. Drones. 4.
Aplicativos. I. Pavan, Willingthon, orientador. II. Título.

CDU: 004.41

Catalogação: Bibliotecária Marciéli de Oliveira - CRB 10/2113

**ATA DE DEFESA DO
TRABALHO DE CONCLUSÃO DE CURSO DO ACADÊMICO**

GUSTAVO CÉSAR VISENTINI

Aos sete dias do mês de dezembro do ano de dois mil e dezoito, às 14 horas, realizou-se, no Instituto de Ciências Exatas e Geociências, prédio B5, da Universidade de Passo Fundo, a sessão pública de defesa do Trabalho de Conclusão de Curso **“Uma Plataforma de aquisição e processamento de Imagens Aéreas”**, de autoria de Gustavo César Visentini, acadêmico do Curso de Mestrado em Computação Aplicada do Programa de Pós-Graduação em Computação Aplicada – PPGCA/UPF. Segundo as informações prestadas pelo Conselho de Pós-Graduação e constantes nos arquivos da Secretaria do PPGCA, o aluno preencheu os requisitos necessários para submeter seu trabalho à avaliação. A banca examinadora foi composta pelos doutores Willingthon Pavan, Rafael Rieder, José Maurício Cunha Fernandes e Genei Antonio Dalmago. Concluídos os trabalhos de apresentação e arguição, a banca examinadora considerou o candidato APROVADO. Foi concedido o prazo de até quarenta e cinco (45) dias, conforme Regimento do PPGCA, para o acadêmico apresentar ao Conselho de Pós-Graduação o trabalho em sua redação definitiva, a fim de que sejam feitos os encaminhamentos necessários à emissão do Diploma de Mestre em Computação Aplicada. Para constar, foi lavrada a presente ata, que vai assinada pelos membros da banca examinadora e pela Coordenação do PPGCA.



Prof. Dr. Willingthon Pavan- UPF
Presidente da Banca Examinadora
(Orientador)



Dr. Genei Antonio Dalmago – Embrapa
(Avaliador Externo)



Prof. Dr. Rafael Rieder - UPF
(Avaliador Interno)



Prof. Dr. Rafael Rieder
Coordenador do PPGCA



Prof. Dr. José Maurício Cunha Fernandes –
UPF
(Avaliador Interno)

“A persistência é o menor caminho do êxito”
(Charles Chaplin)

UMA PLATAFORMA DE AQUISIÇÃO E PROCESSAMENTO DE IMAGENS AÉREAS

RESUMO

A ampliação do campo de visão de algo de interesse pode trazer vários benefícios, dentre eles, a obtenção e análise de dados provenientes de cultivos no meio agrícola de maneira mais ágil. Este é um processo no qual várias imagens capturadas em sequência devem ser alinhadas de acordo com pontos em comum entre elas, formando assim um mosaico ou um grande mapa. Neste contexto, este trabalho apresenta o desenvolvimento de uma plataforma de aquisição e processamento de imagens aéreas, que integra dois softwares: um aplicativo iOS de controle autônomo de Drones, utilizado para fazer a captura das imagens; e um software de mosaicagem utilizando o framework OpenCV. Como estudo de caso, utilizaram-se dados de um experimento de responsabilidade da Embrapa Trigo junto à área experimental II no município de Coxilha - RS. Utilizando a plataforma e um sensor multiespectral TETRACAM acoplado a um Drone, dados NDVI dos cultivos foram coletados e posteriormente uma análise explica que o NDVI obtido pelos sensores apresentou relação direta com a produtividade cultivar.

Palavras-Chave: Drones, Aplicativos, iOS, OpenCV, Automático.

A PLATFORM FOR ACQUISITION AND PROCESSING OF AERIAL IMAGES

ABSTRACT

The amplification of the field of vision of something of interest can bring several benefits, among them, the obtaining and analysis of data from cultivars in the agricultural environment in a more agile way. This is a process in which multiple images captured in sequence must be aligned according to points in common between them, thus forming a mosaic or a large map. In this context, this work presents the development of an aerial image acquisition and processing platform, which integrates two softwares: an iOS application of autonomous Drones control, used to capture the images; and mosaic software using the OpenCV framework. As a case study, data from an experiment of Embrapa Trigo's responsibility were used in experimental area II in Coxilha - RS. Using the platform and a TETRACAM multispectral sensor coupled to a Drone, NDVI data of the cultivars were collected and later an analysis explains that the NDVI obtained by the sensors presented a direct relationship with the cultivar productivity.

Keywords: Drones, Applications, iOS, OpenCV, Automatic.

LISTA DE FIGURAS

| | | |
|------------|--|----|
| Figura 1. | Primeira imagem aérea capturada por James Wallace Black em seu balão (a) na cidade de Boston EUA (b) (adaptado do site do Museu Metropolitano) [10]. . . | 19 |
| Figura 2. | Mosaico de fotos aéreas da barra da tijuca - RJ em 1984. (adaptado do site da UB) [11]. | 20 |
| Figura 3. | Pontos de controle do Software Photoscan (adaptado o site Agisoft Photoscan) [12]. | 21 |
| Figura 4. | Mosaico de duas imagens utilizando a técnica de mosaico não controlado . . | 22 |
| Figura 5. | Mosaico semi-controlado | 22 |
| Figura 6. | Modelo 3D do Terreno gerado pelo software PIX4Dmapper. | 23 |
| Figura 7. | Câmeras Industriais | 23 |
| Figura 8. | Câmeras Industriais Dispersas na cena. | 24 |
| Figura 9. | Mosaico NDVI | 24 |
| Figura 10. | Detecção de Objetos Utilizando o <i>framework</i> OpenCV. | 28 |
| Figura 11. | Técnica de "Costura" criada por Juan e Gwun[27]. | 28 |
| Figura 12. | Distorção Radial | 29 |
| Figura 13. | Distorção Radiométrica | 30 |
| Figura 14. | Equação para cálculo de Índice de vegetação por diferença normalizada . . . | 30 |
| Figura 15. | Índice de vegetação por diferença normalizada | 31 |
| Figura 16. | Câmera multispectral Tetracam Micro ADC [43]. | 32 |
| Figura 17. | Sensor Multispectral | 32 |
| Figura 18. | Câmera Micasense RedEdge-M. | 33 |
| Figura 19. | Exemplo de diagnostico de desenvolvimento de plantação. | 34 |
| Figura 20. | Diagrama de atividades | 36 |
| Figura 21. | Instalação do gerenciador de dependências Cocoapods. | 37 |
| Figura 22. | Configuração do arquivo Podfile do projeto no XCODE. | 38 |
| Figura 23. | Instalação e integração do SDK com o projeto. | 38 |
| Figura 24. | Chave de acesso para a utilização do DJISDK | 38 |
| Figura 25. | Tela inicial do aplicativo Auto Capture. | 39 |
| Figura 26. | Tela de configuração de uma nova missão no Auto Capture. | 40 |
| Figura 27. | Telas do aplicativo Auto Capture. | 41 |
| Figura 28. | Função para coletar waypoints do MapView. | 42 |
| Figura 29. | Função de upload da missão para o Drone. | 42 |

| | | |
|------------|--|----|
| Figura 30. | Referencia dos botões do aplicativo | 43 |
| Figura 31. | Informações da tela em tempo real. | 43 |
| Figura 32. | Encerramento de missão do Drone | 44 |
| Figura 33. | <i>Overlapping</i> entre duas imagens vizinhas. | 45 |
| Figura 34. | DJI Phantom 3 Professional. | 46 |
| Figura 35. | Drone DJI DIY montado no grupo de pesquisas Mosaico UPF. | 46 |
| Figura 36. | Imagem provenientes dos sensores dos Drones. | 47 |
| Figura 37. | Código para criação de um panorama utilizando a classe <i>Stitcher</i> | 52 |
| Figura 38. | Exemplo de saída disponível na documentação do OpenCV. | 53 |
| Figura 39. | Resultado do processamento de imagens de um campo. | 53 |
| Figura 40. | Pipeline do Stitching API. | 54 |
| Figura 41. | Técnica de retângulo da área de interesse, onde o algoritmo determina a região da imagens aonde deve-se procurar os pontos em comum para a mosaicagem. | 55 |
| Figura 42. | Áreas de cálculo do ROI em cada uma das imagens. | 56 |
| Figura 43. | Chamada das funções de mosaicagem e encontro dos ROIs. | 57 |
| Figura 44. | Diretório do Software com o executável e as pastas de entrada e saída de imagens. | 57 |
| Figura 45. | Mosaicos RGB gerados pelo Software desenvolvido. Pode-se observar uma alteração na cor dos cultivares ao decorrer dos dias. | 58 |
| Figura 46. | Mosaicos NDVI gerados pelo software. Estas imagem ainda não foram processadas por um software que faz o cálculo de NDVI. | 59 |
| Figura 47. | Croqui do experimento | 63 |
| Figura 48. | Equipamentos utilizados nas coletas, (48(a)) Drone utilizado, (48(b)) Câmera Multispectral Micro ADC [43]. | 64 |
| Figura 49. | Gráficos de correlação entre os valores de NDVI e Biomassa das culturas Parrudo, Toruk, Guamirim e Tarrumã. Foram utilizados os dados coletados nas 3 primeiras coletas, 10/08/2017, 24/08/2017, 06/09/2017. | 65 |
| Figura 50. | Gráfico de correlação das 4 cultivares | 66 |
| Figura 51. | Correlação entre diferentes doses de N e NDVI das cultivares. | 67 |

LISTA DE TABELAS

| | | |
|-----------|--|----|
| Tabela 1. | Modelos que permitem desenvolvimento pelo SDK. | 27 |
|-----------|--|----|

SUMÁRIO

| | | |
|----------|--|-----------|
| 1 | INTRODUÇÃO | 17 |
| 1.1 | PROBLEMA DE PESQUISA | 17 |
| 1.2 | JUSTIFICATIVA | 17 |
| 1.3 | OBJETIVOS | 18 |
| 1.3.1 | OBJETIVO GERAL | 18 |
| 1.3.2 | OBJETIVOS ESPECÍFICOS | 18 |
| 2 | REVISÃO DE LITERATURA | 19 |
| 2.1 | MOSAICAGEM | 19 |
| 2.1.1 | Histórico da Mosaicagem | 19 |
| 2.1.2 | Tipos de Mosaicagens | 20 |
| 2.1.2.1 | Mosaico Controlado | 20 |
| 2.1.2.2 | Mosaico não Controlado | 21 |
| 2.1.2.3 | Mosaico Semi-Controlado | 21 |
| 2.1.3 | Softwares de Mosaicagem | 22 |
| 2.1.4 | O uso da Mosaicagem | 23 |
| 2.2 | DESENVOLVIMENTO DE SOFTWARES | 25 |
| 2.2.1 | IDE de desenvolvimento XCODE | 25 |
| 2.2.2 | Swift | 25 |
| 2.2.3 | Cocoapods | 26 |
| 2.2.4 | SOFTWARE DEVELOPMENT KIT (SDK) DJI | 26 |
| 2.2.5 | Drones DJI | 26 |
| 2.2.6 | OpenCV | 27 |
| 2.2.6.1 | Correções Geométricas | 29 |
| 2.2.6.2 | Distorção Radial | 29 |
| 2.2.6.3 | Distorção Radiométrica | 29 |
| 2.3 | NDVI | 30 |
| 2.3.1 | Normalized Difference Vegetation Index - NDVI | 30 |
| 2.3.2 | Sensores NDVI | 31 |
| 2.3.3 | Uso de dados NDVI na agricultura | 33 |

| | | |
|----------|--|-----------|
| 3 | APLICATIVO IOS PARA COLETA AUTÔNOMA DE IMAGENS E MONITORAMENTO AGRÍCOLA | 35 |
| 3.1 | RESUMO | 35 |
| 3.2 | INTRODUÇÃO | 35 |
| 3.3 | TRABALHOS RELACIONADOS | 35 |
| 3.4 | DESENVOLVIMENTO DO APLICATIVO | 36 |
| 3.4.1 | Preparação do ambiente e integração com o DJISDK | 37 |
| 3.4.2 | Características e interface do Aplicativo | 39 |
| 3.4.3 | Desenvolvimento das funcionalidades do aplicativo | 41 |
| 3.5 | USO DO APLICATIVO NO MEIO AGRÍCOLA. | 44 |
| 3.5.1 | Tipos de Drones e equipamentos utilizados nos testes | 45 |
| 3.6 | CONCLUSÕES | 47 |
| 4 | APLICATIVO PARA MOSAICAGEM DE IMAGENS DIGITAIS | 49 |
| 4.1 | RESUMO | 49 |
| 4.2 | INTRODUÇÃO | 49 |
| 4.3 | TRABALHOS RELACIONADOS | 50 |
| 4.4 | DESENVOLVIMENTO DO SOFTWARE | 51 |
| 4.4.1 | Obtenção do Mosaico com a stitching API | 51 |
| 4.4.2 | Otimização da Mosaicagem | 54 |
| 4.5 | TESTES E VALIDAÇÕES | 58 |
| 4.6 | CONCLUSÕES | 60 |
| 5 | RELAÇÃO ENTRE BIOMASSA E NDVI EM DIFERENTES CULTIVARES DE TRIGO E DOSES DE NITROGÊNIO | 61 |
| 5.1 | RESUMO | 61 |
| 5.2 | INTRODUÇÃO | 61 |
| 5.3 | MATERIAIS E MÉTODOS | 62 |
| 5.4 | RESULTADOS | 64 |
| 5.5 | CONCLUSÕES | 67 |
| 6 | CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS | 69 |
| 7 | CONCLUSÕES | 71 |
| | REFERÊNCIAS | 73 |

1. INTRODUÇÃO

1.1 PROBLEMA DE PESQUISA

A criação de um mosaico de imagens digitais é um processo no qual várias fotos capturadas em sequência devem ser alinhadas de acordo com pontos em comum entre elas, formando assim um mosaico ou um grande mapa em alta resolução [1].

Após o surgimento dos primeiros aviões (1903) a aerofotografia tornou-se um campo novo, o qual tem como objetivo capturar fotos que relatam a topografia terrestre. O limite de altura de voo dos primeiros aviões não permitiam cobrir grandes áreas, forçando que especialistas construíssem mosaicos com as imagens, foram os primeiros grandes mapas de imagens aéreas construídos manualmente [2]. A necessidade de construção de mosaicos continuou a aumentar. Com o surgimento dos satélites, imagens de alta resolução do planeta começaram a ser capturadas, gerando mais interesse pela área. Deste modo técnicas de mosaicagem computacional foram criadas e os primeiros mapas utilizando imagens digitais começaram a dar forma a um grande mosaico de imagens do planeta Terra.

Com o surgimento dos drones, uma nova ferramenta para obtenção de imagens aéreas se torna disponível. As imagens capturadas por estes equipamentos cobrem áreas menores do que imagens provenientes de satélites, no entanto, apresentam maior flexibilidade de resolução temporal e espacial, pois não possuem intervalos de revisita e a altura do voo pode ser ajustada. A grande quantidade de imagens capturadas, geralmente de forma sequencial, requer um pós-processamento para a união e a geração de mosaicos, resultando em imagens que retratam toda área de interesse.

A construção de mosaicos e o seu uso na topografia vem se tornando cada vez mais importantes [3]. Desde os primeiros métodos de mosaicagem, técnicas e equipamentos fotográficos foram criados e melhorados [4]. Deste modo a utilização de equipamentos e o processamento de dados para o mapeamento aéreo faz uso de múltiplas estratégias e tecnologias, incluindo uma série de passos a serem efetuados para alcançar um produto final de mapeamento, um mosaico em alta resolução. Entretanto a operação dos equipamentos e o processo a ser executado como um todo, torna-se difícil quando se pensa em algo prático e ágil, o que dificulta o uso para pessoas com pouca experiência na área [5]. Além disso, em alguns casos ocasiona demora na obtenção dos resultados, devido a quantidade de atividades a serem executadas para se chegar a um resultado final satisfatório [6].

1.2 JUSTIFICATIVA

Com o avanço tecnológico, os drones surgem como novas opções para o geo-mapeamento, bem como captura de imagens e dados em tempo menor do que utilizando outros meios como satélites e aviões. Outro fator que agrega vantagens ao uso de drones, é a variedade de equipamentos disponíveis no mercado, podendo-se adquirir com facilidade os equipamentos para seu uso pessoal, por exemplo.

A captura das imagens deve ser feita pelo Drone de forma sequencial, uniforme e com altitude constante, é um fator importante para que posteriormente o software de mosaicagem execute tal processo. Por sua vez, é importante também que seja feito um levantamento da área a ser sobrevoada, a preparação de todo o equipamento a ser utilizado, além da sua devida calibração. Por último, o operador deve estar preparado para que então possa executar todo o processo antes planejado.

Atualmente o uso de drones no Brasil torna-se um aliado na Agricultura de Precisão, permite detectar e monitorar grandes áreas quase em tempo real. Por meio de imagens geradas consegue-se identificar onde combater as pragas, ou receber reforços de adubação no solo de forma mais específica, evitando os desperdícios, e conseqüentemente trazendo ao produtor um aumento de produtividade [7].

Pensando no meio urbano, a utilização de mosaicos pode contribuir para o correto crescimento da cidade, de forma que o levantamento geográfico feito de forma rápida minimiza custos operacionais, além de trazer imagem atuais do estado em que a área se encontra. Segundo Rosseti [8], a visualização de grandes áreas por meio de mosaicos aerofotogramétricos, mostra-se muito importante para o correto planejamento da expansão da malha urbana.

Neste contexto, a dificuldade no processo de mosaicagem de imagens abre a possibilidade de desenvolvimento de uma plataforma que possibilite automatizar este processo por meio de aplicações mobile que façam o controle do Drone, juntamente com softwares que utilizem visão computacional para realizar o processo de mosaicagem de imagens, disponibilizando ao usuário final um resultado confiável de forma prática e ágil.

1.3 OBJETIVOS

1.3.1 OBJETIVO GERAL

Implementar uma plataforma de aquisição e processamento de imagens utilizando técnicas de Visão Computacional.

1.3.2 OBJETIVOS ESPECÍFICOS

- Definir uma metodologia de coleta de imagens quanto á altura de voo, distancia entre uma foto e outra, velocidade de voo, dados que devem ser coletados.
- Desenvolver um algoritmo de mosaicagem seguindo a metodologia proposta.
- Desenvolver um aplicativo para a plataforma iOS, que faça o controle automático de Drones DJI, utilizados para a coleta de imagens.
- Desenvolver uma aplicação para criar mosaicos de imagens.

2. REVISÃO DE LITERATURA

2.1 MOSAICAGEM

A construção de um mosaico de imagens é um processo que consiste em sobrepor uma imagem sobre a outra, levando em consideração pontos em comum, gerando assim um mosaico, o que para nós amplia o campo de visão de algo de interesse. Deste modo a utilização de mosaicos pode se tornar importante quando necessita-se fazer o mapeamento de grandes áreas, para tal, precisa-se capturar uma grande quantidade de fotos e devido a vários fatores como por exemplo a altura de onde ela foi capturada, de forma que uma única imagem capturada não consegue cobrir toda a área desejada, fazendo-se então necessário o uso desta técnica [1].

2.1.1 Histórico da Mosaicagem

Os primeiros registros sobre a mosaicagem de imagens são de tempos antes da era dos computadores digitais, um processo fotográfico desenvolvido em 1839. Sua utilização mostrava-se muito eficiente para o mapeamento topográfico [9]. Várias imagens coletadas por meio do topo de montanhas ou por balões (Figura 1) foram sendo montados um ao lado do outro, exatamente como montar um quebra cabeça, formando assim um dos primeiros métodos de mosaicagem.

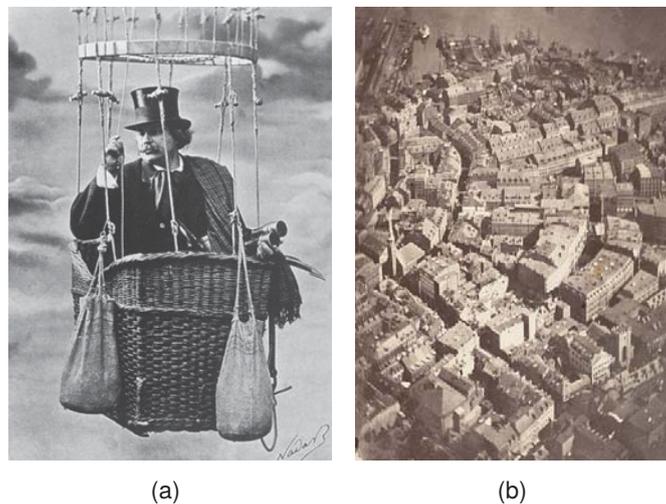


Figura 1. Primeira imagem aérea capturada por James Wallace Black em seu balão (a) na cidade de Boston EUA (b) (adaptado do site do Museu Metropolitano) [10].

Após o desenvolvimento dos primeiros aviões (1903) [2] a aerografia tornou-se um campo novo, a qual tinha como objetivo capturar fotos que relatavam a topografia terrestre. O limite de altura de voo dos primeiros aviões, e a necessidade de grandes mapas, forçaram especialistas em imagens a construir mosaicos de forma em que uma imagem é sobreposta sobre a outra, tornando assim estes

os primeiros grandes mosaicos construídos manualmente. Na Figura 2, pode-se observar um exemplo destes primeiros mosaicos.



Figura 2. Mosaico de fotos aéreas da barra da tijuca - RJ em 1984. (adaptado do site da UB) [11].

A Necessidade de construção de mosaicos continuou a aumentar e com a chegada da era dos satélites, onde os mesmos podiam enviar imagens do planeta Terra em alta resolução, o interesse por esta área começou a aumentar. Sendo assim novas técnicas de mosaicagem computacional precisaram ser criadas e os primeiros mapas utilizando imagens reais foram dando forma a um grande mapa do planeta Terra.

A construção de mosaicos e o seu uso na topografia vem se mostrando importante há bastante tempo [3]. Desde os primeiros métodos de mosaicagem muitas técnicas e equipamentos fotográficos foram criados e melhorados [4]. Deste modo, a utilização de equipamentos e o processamento de dados para mapeamento aéreo faz uso de múltiplas estratégias e tecnologias, incluindo uma série de passos e trabalhos a serem efetuados para poder chegar ao produto final de um mapeamento, um mosaico em alta resolução.

2.1.2 Tipos de Mosaicagens

Com o passar dos anos e a constante evolução da computação e das técnicas empregadas na construção de mosaicos alguns padrões foram sendo mais usados. Deste modo, segundo Wolf e Dewitt [9] a criação de mosaicos provenientes de fontes aéreas geralmente são divididos em três classes: Mosaicos Controlados, Mosaicos Não Controlados e Mosaicos Semi-controlados.

2.1.2.1 Mosaico Controlado

Um mosaico controlado é o mais acurado e preciso dos três tipos. No processo manual, estes tipos de mosaicos são preparados a partir de fotografias que podem ser retificadas, isto significa que todas as impressões são realizadas equivalentes a fotografias verticais, as quais tem a mesma escala nominal. Quando são montados, as posições das feições imageadas, comuns nas fotos adjacente, são

emparelhadas o mais próximo possível. Para uma maior acurácia global de montagem, um arranjo de pontos de controle é preparado para a mesma escala como as fotos retificadas. Então, no processo de concatenação das várias fotos para formar o mosaico, os pontos de controle nas imagens são também sobrepostos com os seus correspondentes pontos de controle marcados para restringir a posição das fotos. Na figura 3, podemos observar pontos de controle em cores azul sendo processadas pelo software Agisoft Photoscan [12].

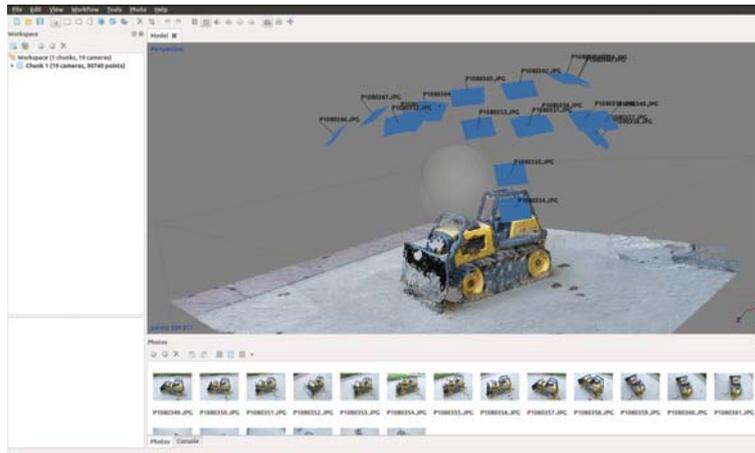


Figura 3. Visualização dos pontos de controle inseridos previamente no software Photoscan pelo usuário (adaptado do site Agisoft Photoscan) [12].

2.1.2.2 Mosaico não Controlado

Este tipo de mosaico é preparado por simples junção entre as imagens de detalhes em fotos adjacentes. Não são usados pontos de controle no solo. Mosaicos não controlados são mais rápidos para serem preparados do que os controlados. Deste modo eles não possuem a mesma acurácia dos mosaicos controlados, além de que quanto mais imagens devem ser mosaicadas, maior será a dificuldade do algoritmo reconhecer pontos em comum entre as mesmas, e conseqüentemente maior será o tempo de processamento. Mas para uso geral, ele pode se tornar viável, como por exemplo, unir apenas duas imagens para se obter um maior panorama da área como pode-se observar na Figura 4.

2.1.2.3 Mosaico Semi-Controlado

Os mosaicos semi-controlado são construídos utilizando combinações de especificações entre os controlados e os não controlados. Este tipo de mosaico pode ser construído, por exemplo pelo uso de pontos de controle no solo empregando imagens que não são retificadas e planificadas. Outro tipo de combinação que pode ser utilizada é o uso de imagens retificadas e planificadas, e não utilizar os pontos de controle em solo. Este tipo de mosaico é um meio termo entre acurácia e economia de processamento e maior rapidez na sua construção. A utilização de pontos de controle nesta técnica de mosaicagem pode ser efetuado com as coordenadas geográficas de cada foto, Latitude e Longitude.



Figura 4. Mosaico de duas imagens utilizando a técnica de mosaico não controlado (Observa-se na parte superior esquerda da imagem, perto do galpão, que existe algumas falhas na sobreposição das duas imagens).

Utilizando este metadados, podemos organizar em um panorama, a ordem de cada foto, tornando assim o processo de mosaicagem mais rápido que o Mosaico Não Controlado [13].

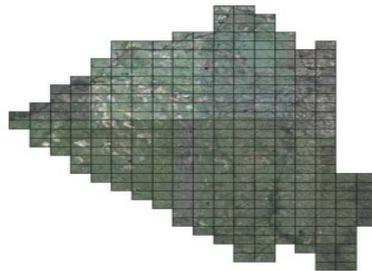


Figura 5. Mosaico semi-controlado, mostrando imagens organizadas para a mosaicagem. [13]

2.1.3 Softwares de Mosaicagem

A área topográfica onde a mosaicagem se encaixa é bastante competitiva em relação a softwares de alto desempenho. Existem algumas opções no mercado, que vão de um software extremamente elaborado como o PIX4Dmapper [14], até alguns aplicativos gratuitos como o Microsoft ICE [15], o qual pode unir imagens para formar uma panorama.

Pix4Dmapper é um aplicativo da Pix4D que calcula as posições e orientações das imagens originais de forma automática por meio da Triangulação Aérea (AAT) e ajustamento de feixes em bloco (Bundle Block Adjustment, BBA). Com base em nuvem de pontos 3D obtidos durante a AAT e BBA, é gerado um Modelo Digital de Superfície (MDS) por ligação destes pontos. Deste modo o ortomosaico é criado a partir da projeção e combinação das imagens originais com o MDS. [14]. Na Figura 6 pode-se observar um Modelo 3D gerado pelo PIX4Dmapper.

Seguindo no ramo de softwares proprietários, tem-se o Photoscan da empresa Agisoft. A lista de funcionalidades inclui foto-triangulação aérea e terrestre com exportação dos dados para outros



Figura 6. Modelo 3D do Terreno gerado pelo software PIX4Dmapper.

softwares fotogramétricos, geração de nuvens de pontos e modelos poligonais triangulares 3D, MDT e MDS, ortofoto e processamento de imagens multi espectral. As sobreposições devem ser de 80 e 60 % para longitudinal e lateral, respectivamente. Tanto o Pix4Dmapper quanto o Photoscan fazem uso das técnicas de mosaicagem controlada, semi-controlada e não controlada, as quais podem ser ajustadas pelo usuário. [12].

Por fim, o Microsoft Image Compositor Editor (Microsoft ICE), de uso simplificado também realiza construções de mosaicos, porém com menos recursos comparados aos anteriores. Este software realiza tanto mosaicagens de imagens verticais, ou também imagens panorâmicas, obtendo por fim um resultado satisfatório. Neste caso o software utiliza o método de mosaicagem não controlado, pois na construção do mosaico, em nenhum momento é feito a inserção de pontos de controle, característicos de softwares que utilizam técnicas de mosaico controlado [15].

2.1.4 O uso da Mosaicagem

Mosaicos de imagens digitais podem ser aplicadas para diferentes propósitos como, por exemplo, no monitoramento de áreas industriais. Segundo os autores [16], o cenário encontrado, consiste em um ambiente de inspeção industrial com quatro câmeras posicionadas em forma de uma matriz 2x2, como mostrado na Figura 7.

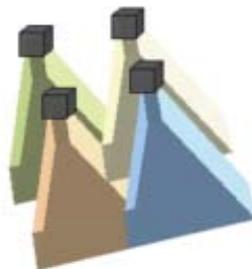


Figura 7. Cenário com quatro câmeras de vigilância. As imagens possuem uma borda de sobreposição com a outra, tornando assim possível unir elas em uma só imagem [16]

Desta forma, existem regiões da cena que são imageadas por uma ou mais câmeras. Nela, pode ser observado que existe uma região comum entre as imagens da região 1 e da 2. Uma imagem resultante da agregação destas duas imagens terá outra região em comum com a região 3, assim como a região 4. Desta forma, as imagens das regiões 1, 2, 3 e 4 podem ser agrupadas gerando um mosaico, e desta forma criando um panorama muito maior no qual poderá ser mais fácil inspecionar o ambiente industrial.

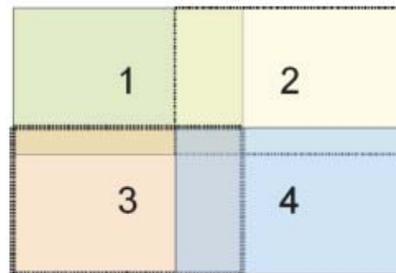


Figura 8. Vista superior, com as regiões imageadas representadas por cada câmera. [16]

Pensando no meio urbano, a utilização de mosaicos pode ser utilizada para o levantamento geográfico, como forma de minimizar custos operacionais e manter o mapa atualizado da área de interesse. Segundo Rossetti [8], a visualização de grandes áreas por meio de mosaicos aerofotogramétricos mostra-se muito importante para o correto planejamento da expansão da malha urbana.

Quando considera-se o meio agrícola, pode-se observar que os Drones já fazem parte do dia-a-dia de diversas atividades. Por meio de captura de imagens multiespectrais é possível obter-se índices de vegetação como o *Normalized Difference Vegetation Index* (NDVI). Considerado um bom estimador de biomassa, o NDVI possui forte correlação com o crescimento das culturas, sendo utilizado para avaliar o estado da vegetação [17]. Como esperado, as culturas agrícolas ocupam grandes áreas, necessitando capturar inúmeras imagens para cobrir toda a área, fazendo-se necessário a utilização de técnicas de mosaicagem. A Figura 9 demonstra um mosaico NDVI de uma área agrícola.

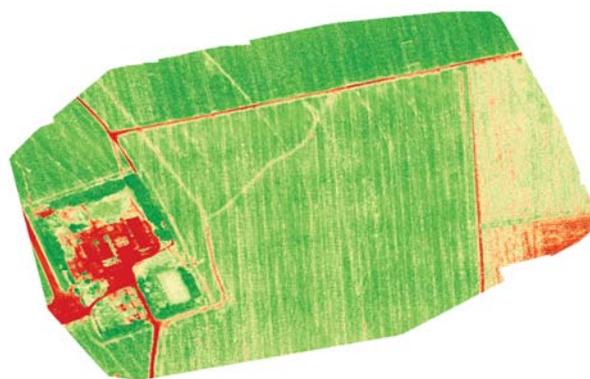


Figura 9. Mosaico construído a partir de imagens NDVI.

2.2 DESENVOLVIMENTO DE SOFTWARES

Várias ferramentas para desenvolvimento de softwares estão disponíveis para uso, bem como linguagens de programação. Para este trabalho, optou-se pelo uso de algumas delas, descritas nas próximas subseções.

2.2.1 IDE de desenvolvimento XCODE

Atualmente, todos os aplicativos iOS, MacOS, AppleTV ou WhatchOS, são desenvolvidos por meio da IDE XCODE [18]. Ele é um ambiente de desenvolvimento desenhado e construído para permitir um fluxo de trabalho mais suave. Com a integração do *Interface builder*, ele permite integrar a edição do código-fonte, construção, compilação e depuração da aplicação. O *Interface builder* é um designer de interface fácil de usar que permite projetar os aspectos de interface do usuário de aplicativos Mac OS X e iOS.

Também é disponibilizado junto ao XCODE um simulador de todos dispositivos que podem utilizar a aplicação. Desta maneira fornece um meio de testar sua aplicação e ver como ela aparecerá no dispositivo real. O Simulador garante que a interface de usuário funcione e se comporte da maneira desejada e facilite a depuração do aplicativo. O Simulador contém algumas limitações, que não podem ser usadas para testar recursos certificados, portanto, é sempre melhor testar no dispositivo real.

2.2.2 Swift

Em 2014, na tradicional Worldwide Developers Conference (WWDC) da Apple, realizado em San Francisco, nos Estados Unidos, foi apresentado ao público uma nova linguagem de programação batizada de Swift. A linguagem tem como objetivo substituir a linguagem Objective-C, a qual formou a base do OS X e iOS. Juntamente com a linguagem, o XCODE e os simuladores foram apresentados com novidades e suporte a nova linguagem.

Em menos de um ano e meio após o seu primeiro lançamento, o Swift tornou-se uma das linguagens de programação mais populares do mundo, considerando diferentes medidas de popularidade. Uma parte significativa de este sucesso é devido ao controle estrito da Apple sobre seu ecossistema, e a mensagem clara de que substituirá a Objective-C em um futuro. Segundo a Apple, “Swift é uma poderosa e intuitiva linguagem de programação[...]. Escrever código Swift é interativo e divertido, a sintaxe é concisa, mas expressiva” [19].

2.2.3 CocoaPods

CocoaPods é um gerenciador de dependências para Swift e Objective-C [20]. Ele possui mais de 49 mil bibliotecas e é usado em mais de 3 milhões de aplicativos para direcionar caminhos e integrar os recursos de cada projeto.

O CocoaPods foi construído utilizando *Ruby* e pode ser instalado com o *Ruby* padrão disponível no OS X. Sua utilização, de forma simplificada, consiste em colocar em um arquivo de configuração o nome da dependência obtida no site do CocoaPods, e bastando via *prompt de comando*, executar os comandos de configurar e instalar. Desta forma, é gerado um novo atalho de projeto, contendo todas as dependências já configuradas e funcionando.

2.2.4 SOFTWARE DEVELOPMENT KIT (SDK) DJI

O mapeamento e a captura de imagens no meio aéreo cresceram muito nos últimos anos, tanto para fins recreativos, como para fins profissionais (fotografias, sensoriamento remoto, segurança), cresceram a partir de 2010. Por sua vez os Drones destinados ao Geomapeamento têm um custo elevado devido ao software e hardware embarcado, além do conhecimento na área que o usuário deve conter para operar o mesmo.

Desta maneira, novas ideias e metodologias são propostas para automatizar o processo de captura, utilizando Drones de baixo custo que disponibilizem um SDK de desenvolvimento Mobile.

Com o objetivo de facilitar e incentivar o desenvolvimento de novos aplicativos e soluções, a DJI disponibilizou um SDK de desenvolvimento para usuários de seus Drones. Segundo a documentação do site [21] o SDK esta disponível em três diferentes segmentos .

- **SDK Mobile:** Permite o desenvolvimento de aplicação Mobile tanto para a plataforma iOS, tanto para Android. Seu objetivo é disponibilizar o acesso a todos os recurso do Drone, tornando possível a customização do uso do aparelho.
- **SDK Onboard:** Sistema dedicado para comunicar com o controlador de vôo DJI através de uma conexão serial direta. Monitorar e controlar o comportamento de voo da aeronave com as funções da API de bordo, enquanto utilizando os modos de navegação inteligentes integrados para criar rotas autônomas.
- **SDK Guidance:** Permite desenvolver facilmente todos os tipos de aplicações de orientação e visão do Drone, através dos sensores anti-colisão que são acoplados ao mesmo.

2.2.5 Drones DJI

Em 2014, mais de 500.000 Drones foram comercializados nos EUA, sendo que cerca de 1/3 dos mesmo foram do modelo DJI Phantom [22]. A fabricante DJI disponibiliza alguns modelos

de Drones que possibilitam o desenvolvimento de aplicações por meio de um Software Development KIT - (SDK), que permite acessar todas as funcionalidade por meio de comunicação serial, tendo assim controle total sobre o Drone. Na Tabela 1 estão listados alguns modelos de Drones da DJI que permitem o uso deste SDK.

Tabela 1. Modelos que permitem desenvolvimento pelo SDK.

| Modelo | Câmera Padrão | Câmera Cambiável | Preço |
|---------------|------------------|------------------|-------------|
| Phantom 3 Pro | Std/4K/12MP/2.8f | Não | 999,00 U\$ |
| Phantom 4 | Std/4K/12MP/2.8f | Não | 1399,00 U\$ |
| Inspire 1 | X3/4K/12MP/2.8f | Sim | 1999,00 U\$ |
| Inspire 1 PRO | X5/4K/12MP/1.7f | Sim | 3899,00 U\$ |
| Matrice 100 | X3/4K/12MP/2.8f | Sim | 3299,00 U\$ |
| Matrice 600 | X5/4K/12MP/1.7f | Sim | 4599,00 U\$ |

Levando em consideração a questão financeira, bem como a praticidade, atualmente não existem outros modelos de Drones a venda no mercado que possuam uma solução completa, utilizando câmeras com boa resolução. Porém, em alguns casos podem ser melhoradas e até trocadas por câmeras com filtros específicos para a agricultura como por exemplo o NDVI [23], e também disponibilizam SDK de desenvolvimento para o usuário.

2.2.6 OpenCV

OpenCV (Open Source Computer Vision) é uma biblioteca de programação de código aberto, que inicialmente foi desenvolvida pela Intel Corporation. Ela implementa uma variedade de ferramentas para interpretação de imagens, indo desde operações simples como um filtro de ruído, até operações complexas, tais como reconstruções 3D, reconhecimentos de padrões e análise de movimentos, por meio de funções em linguagem C e algumas classes C++, o OpenCV mostra-se muito robusto também para operação em matrizes .

A OpenCV trabalha com dois paradigmas: O processamento de imagens e a visão computacional. O processamento de imagens é onde a entrada do sistema é uma imagem, e a saída é um conjunto de valores numéricos, que por fim podem ou não compor uma nova imagem. Por sua vez a visão computacional procura emular a visão humana, portanto também possui como entrada uma imagem, mas a saída é uma interpretação da imagem com um todo ou parcialmente. Como exemplo, Gazcón et. al [24] apresentam uma aplicação na qual devem identificar uma placa de um veículo, mas ele está um tanto quanto escuro. Então, utilizando técnicas de processamento de imagens, a imagem é clareada, para que então posteriormente, a visão computacional, possa identificar as letras e números da placa do veículo.

A Biblioteca OpenCV possui mais de 500 funções [25]. A biblioteca está subdividida em cinco grupos de funções. Processamento de imagens; Análise estrutural; Análise de movimento e rastreamento de objetos; Reconhecimento de padrões e Calibração de câmera e reconstrução 3D.

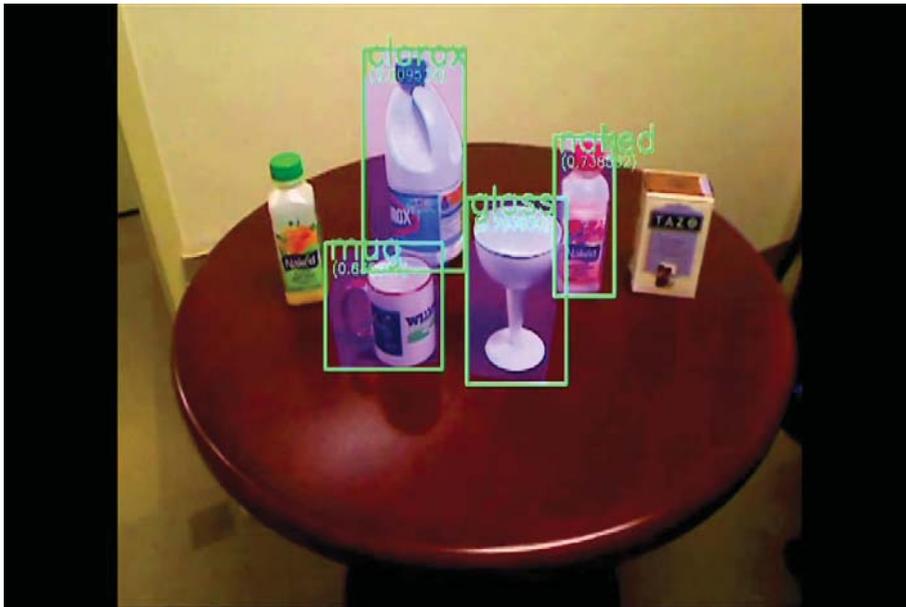


Figura 10. Detecção de Objetos Utilizando o *framework* OpenCV.

Por intermédio de uma solução com OpenCV, Leemans et. al [26] apresentam uma alternativa para a estimativa de Índice de Área folhar (IAF) na cultura do Trigo, por meio do processamento de imagens coletadas. Foram realizadas capturas de imagens em campo e em laboratórios. Já Juan e Gwun [27] desenvolveram uma solução com OpenCV para combinar diferentes imagens de um mesmo cenário (Figura 11) e criar uma imagens panorama. Este tipo de técnica, também é conhecida como mosaicagem ou Stitching (em português, costura), [28] é muito estudada e aplicada na solução de diferentes problemas, entre eles, para realizar a combinação (mosaicagem) de imagens aéreas coletadas por meio de Drones[27].



Figura 11. Técnica de "Costura" criada por Juan e Gwun[27]. Input são as imagens de entrada e output o resultado da mosaicagem das imagens.

2.2.6.1 Correções Geométricas

As transformações geométricas são ferramentas importantes para quem trabalha ou vai trabalhar com desenhos ou imagens em ambiente computacional. Desta forma esta afirmação também aplica-se à fotogrametria Digital e mosaicagem de imagens. Pode-se citar também dois pontos importantes que fazem parte da correção geométrica: A correção das distorções Radial e Radiométrica a seguir apresentadas.

2.2.6.2 Distorção Radial

A distorção causada pela lente, conhecida como distorção radial, pode ser classificada em distorção positiva, conhecida como *pincushion*, e a distorção negativa, conhecida como barril ou *barrel*. Na Figura 12 observa-se estes tipos de distorções.

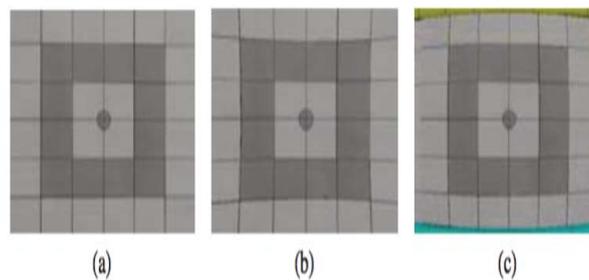


Figura 12. Imagem sem distorção (a) Imagem com distorção positiva (*pincushin*) (b) imagem com distorção negativa (*barrel*) (c) [29]

2.2.6.3 Distorção Radiométrica

O principal critério da distorção radiométrica é a equação fundamental de formação radiométrica em uma imagem. Esta equação define que a variação da intensidade de iluminação mostrada na imagem decresce do centro da imagem para a periferia da mesma, em forma circular em relação ao $\cos^4(x)$, onde x é ângulo de abertura.

A distorção radiométrica ocorre quando é feita a aquisição da imagem, e devido a curvatura da lente pode ser descoberta pela distância da câmera (lentes) em relação à imagem de calibração e do ponto periférico (localizado na mesma reta do ponto central da imagem) à lente da câmera, formando um triângulo. Assim descobrindo o ângulo de abertura do ponto central ao ponto periférico da imagem de calibração em relação à lente da câmera, pode-se determinar os coeficientes para o processo de remoção da distorção radiométrica [29]. Na Figura 13, observa-se o processo de formação da radiometria da imagem.

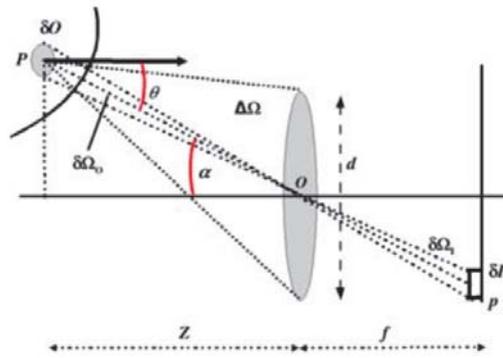


Figura 13. Processo de formação da radiometria da imagem [29].

2.3 NDVI

O NDVI do inglês Normalized Difference Vegetation Index ou em português Índice de vegetação por diferença normalizada é considerado um bom estimador de biomassa. Ele foi proposto em 1973 por Rouse et. al [30], a partir da normalização do DVI do intervalo de -1 a 1. Quando mais próximo de 1, maior é o índice de presença de vegetação, e quanto mais próximo de -1, o índice de solos descobertos, rochas ou vegetação seca é maior.

2.3.1 Normalized Difference Vegetation Index - NDVI

O NDVI é utilizado para mapear o teor de massa verde de dados de uma determinada área e baseia-se no princípio de que a massa verde (plantas) absorve radiação na região visível do espectro eletromagnético, convertendo a energia da luz em energia química (processo de fotossíntese). Ao contrário, o infravermelho próximo (Near Infra Red – NIR) não é utilizado para a fotossíntese, sendo assim, refletido pela planta. Segundo Royo[31], a refletância das plantas pode ser afetada por um número razoável de fatores, como a sua espécie, o seu estado nutricional, o seu estágio fenológico, entre outros.

Desta forma, o NDVI é calculado pela diferença de refletância entre a faixa do NIR e a faixa do vermelho visível (RED) do Espectro Eletromagnético[32]. Este cálculo se dá por meio da Equação 1, onde NIR é a refletância do infravermelho próximo e RED é refletância da faixa de vermelho visível. Duas plantas, uma saudável e outra em situação de estresse, com seus respectivos índices resultantes do cálculo NDVI, são ilustradas na Figura 15.

$$NDVI = \frac{NIR - RED}{NIR + RED}$$

Figura 14. Equação para cálculo de NDVI [30].

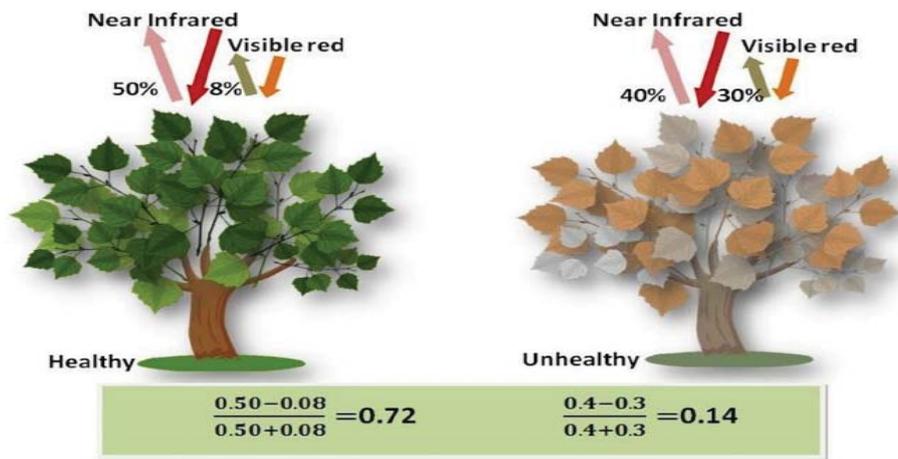


Figura 15. Resultado do cálculo NDVI em uma vegetação saudável e uma estressada [30].

Devido a maior absorção da radiação eletromagnética pela clorofila, a faixa do vermelho, por ser a faixa de maior comprimento de onda do espectro visível, privilegia-se em relação a do azul e a do verde[33].

Segundo Moges[34], índices de diferença normalizados baseados em vermelho (RNDVI) e verde (GNDVI) são comumente utilizados para avaliar a saúde da planta, estimar sua biomassa e seus nutrientes. Correlacionando estes dois índices, por meio de sensores ativos, como biomassa, coberturas de nitrogênio e rendimento de grãos do trigo, Moges concluiu que mesmo em alguns estádios do cultivo o RNDVI mostrou-se consistentemente mais correlacionável com a biomassa. Contudo nenhum dos índices apresentaram vantagens significativas e ambos apresentando boas correlações com as variáveis. Ainda segundo Moges, a correlação entre biomassa e RNDVI tende a diminuir conforme o avanço do estágio da planta.

2.3.2 Sensores NDVI

Trabalhos na literatura como os de Perry et al. [35] e Sugawara [36] apresentam soluções para a agricultura ao utilizarem dados provenientes de sensores orbitais, sendo que outros como FU et al [37], já utilizam sensores terrestres ativos, ou seja, estes sensores que não dependem da luz solar para realizar medições das vegetações. A larga utilização de Drones neste meio abre precedentes para a criação e uso de sensores cada vez menores e mais eficientes [38, 39].

O mercado oferece câmeras com suporte nativo de bandas espectrais que compõem as imagens NDVI, como por exemplo, as câmeras FUJI IS-1 e a canon XNite 450, apresentadas por Rissini[40] e Dworak[41]. É possível também encontrar os chamados sensores multispectrais, os quais possibilitam, além da obtenção de imagens NDVI, a captura de outras bandas do espectro para cálculo de diferentes índices. Um dos sensores referenciado na literatura[42] é fabricados pela empresa Tetracam. A câmera Tetracam ADC (Agriculture Digital Câmera, Figura 16) possui alta resolução e consegue capturar imagens com comprimento de onda a 250nm da faixa visível e 920nm da faixa de infravermelho próximo.



Figura 16. Câmera multispectral Tetracam Micro ADC [43].

A câmera tem um peso de 90 gramas e sua dimensão é de 75 mm × 59 mm x 33 mm, uma câmera adequada para uso em quase todos os tipos de plataforma de Drones existentes. O sensor da câmera (6.55 mm × 4.92 mm, tamanho de pixel de 3.12 microm) é um sensor CMOS no qual adquire imagens nas bandas vermelha (R), verde (G) e infravermelho próximo (NIR), trabalhando com o comprimento de 250nm da faixa visível e 920nm da faixa de infravermelho próximo (Figura 17). As três bandas Tetracam ACD Micro são equivalentes as do Satélite Landsat TM2, TM3 e TM4[44].

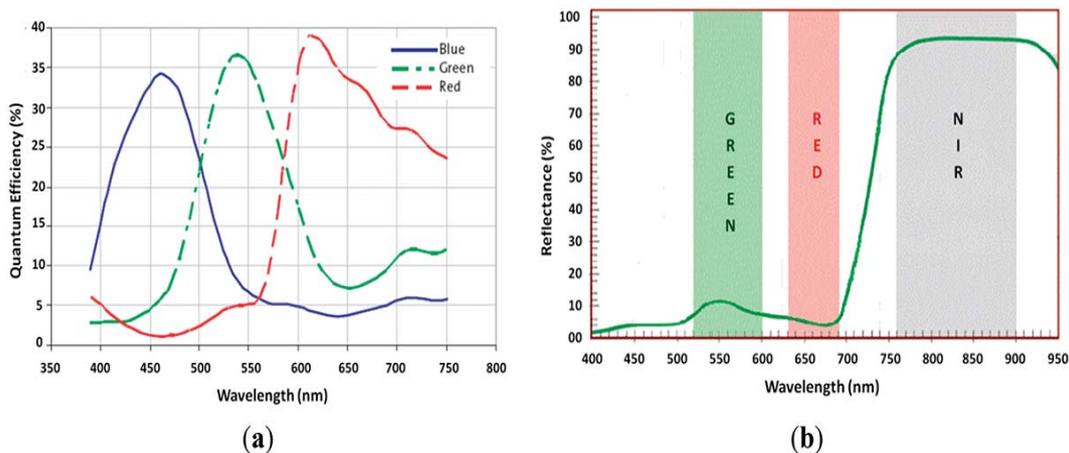
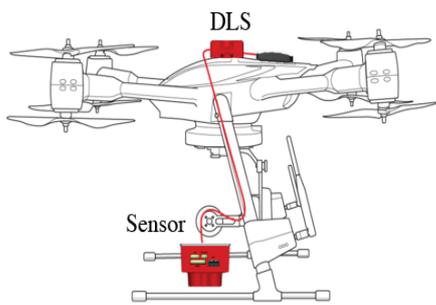


Figura 17. Resposta do sensor Aptina CMOS para as diferentes bandas visíveis dos filtros R,G e B **a**. Bandas da Tetracam ADC Micro com sua faixa espectral e assinatura espectral da vegetação **(b)**[44].

Além dos sensores passivos citados acima, existem também outros tipos de sensores, como o produzido pela empresa Micasense [45]. O sensor RedEdge - M é um sensor que trabalha com uma tecnologia chamada DLS (*Downwelling Light Sensor*), não sofrendo interferência da posição solar ou temporal, pois ajusta a entrada dos dados no sensor e lê a situação no momento (Figura 18). Este sensor pode capturar 5 bandas diferentes do espectro visível, sendo elas a azul, verde, vermelho, borda vermelha e o infravermelho próximo. Seu peso é de 173 gramas e sua dimensão é de 94mm x 63mm x 46mm, sendo que a mesma possui kits de integração com diversos Drones do mercado.



(a) Montagem do sensor Micasense RedEdge - M [45].



(b) Câmera Micasense RedEdge - M [45].

Figura 18. Câmera Micasense RedEdge-M.

2.3.3 Uso de dados NDVI na agricultura

Os Drones estão sendo utilizados para diferentes situações, tanto no meio urbano ou na agricultura como um todo. As aplicações no meio agrícola podem ser as mais diversas, como uma simples coleta de imagem ou até a aplicação de fungicida utilizando estes equipamentos. Seu uso no meio agrícola e em missões de reconhecimento vem sendo favorecida e facilitada pelo desenvolvimento tecnológico atual, principalmente pela redução do custo e do tamanho dos equipamentos e pela necessidade de otimização dos cultivos. Existem várias perspectivas para a agricultura, podendo destacar-se o monitoramento de recursos naturais, meio ambiente, atmosfera, imageamento multiespectral, observações de rios e lagos, bem como práticas agrícolas e uso do solo[46].

Como já abordado anteriormente, o NDVI é um índice de vegetação muito utilizado na agricultura de precisão pois reflete o acúmulo de biomassa, que é altamente influenciado pela saúde das plantas. Tal comportamento ocorre devido a utilização das bandas espectrais do vermelho e do infravermelho próximo em seu cálculo, que servem de indicadores da atividade fotossintética da planta. Assim, caso haja uma menor atividade fotossintética pela planta, os valores de NDVI irão refletir esta alteração. Na agricultura de precisão, vários estudos têm indicado o NDVI como um bom avaliador do crescimento vegetativo da cultura, principalmente porque ele permite uma avaliação temporal do cultivo. Desta forma, todo o desenvolvimento da cultura pode ser monitorado por meio de imagens aéreas, reduzindo os custos de produção.

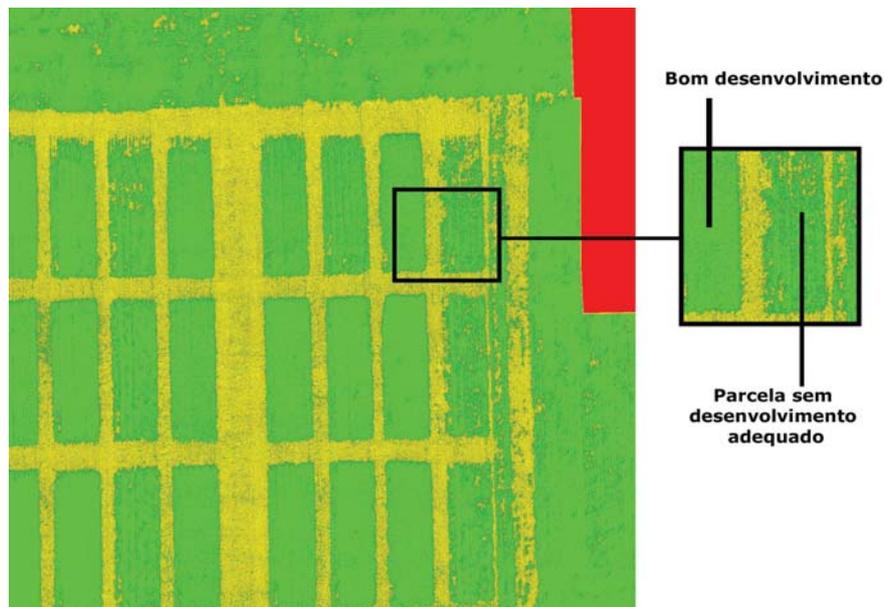


Figura 19. Exemplo de diagnóstico de desenvolvimento do cultivo por meio da agricultura de precisão.

Como o NDVI é sensível às condições biofísicas da vegetação, ele demonstra as áreas onde há alguma anormalidade no cultivar, o que pode significar a presença de doenças, necessidade de adubação ou até mesmo uma escassez hídrica (Figura 19). Isso facilita a tomada de decisões, o planejamento da produção e, principalmente, diminui desperdícios, já que o agricultor passa a saber onde e quanto investir, seja em adubação, defensivo ou irrigação. Cabe lembrar que a utilização dos índices exige conhecimento técnico sobre sensoriamento remoto, agronomia e fenologia do cultivo em questão para obter uma interpretação precisa [47].

Como exemplo, pode-se citar um caso bem sucedido do autor Breademeier [48], onde foi verificada por meio de análise do NDVI, a relação espacialmente inversa entre as áreas de maior produtividade do milho e as áreas com maior quantidade de proteína contido nos mesmos. Desta maneira, segundo o autor, foi possível detectar que as doses de nitrogênio utilizadas estavam inadequadas.

3. APLICATIVO IOS PARA COLETA AUTÔNOMA DE IMAGENS E MONITORAMENTO AGRÍCOLA

3.1 RESUMO

O monitoramento agrícola tem contado com inúmeros métodos e tecnologias, sendo o uso de Drones um dos mais atuais. Neste trabalho apresenta-se o desenvolvimento de um aplicativo para controle e mapeamento aéreo utilizando Drones DJI, desenvolvido para a plataforma iOS. Com o aplicativo pode-se coletar de forma simplificada e automatizada imagens para a criação de mosaicos, os quais podem auxiliar nos mais variados projetos.

3.2 INTRODUÇÃO

Drones podem ser considerados aliados à Agricultura de Precisão pois permitem detectar e monitorar grandes áreas em tempo real. Por meio do uso de imagens pode-se identificar, por exemplo, pragas que possam prejudicar o cultivar ou áreas com necessidade de adubação, buscando auxiliar na tomada de decisões, assim como na busca de um aumento de produtividade e redução de custos na produção [7].

Entretanto, a operação destes equipamentos não é uma tarefa simples, demandando treinamento e conhecimento específico, tornando-se algo difícil quando se pensa em praticidade e agilidade. Desta forma o uso destes equipamentos para pessoas com pouca experiência na área torna-se difícil, e também em alguns casos ocasiona demora na obtenção dos resultados, devido a quantidade de passos a serem executados para chegar a um resultado final satisfatório.

Em sua maioria, Drones podem ser equipados com diferentes tipos de sensores, como câmeras, barômetros, giroscópios e GPS. Diante deste contexto, faz-se necessário utilizar modelos que possuam um SDK (*Software Development Kit*), possibilitando assim acessar suas funcionalidades e podendo desenvolver aplicativos para automatizar processos como a captura de imagens [49].

Neste contexto, este capítulo apresenta a criação de um aplicativo simples e de fácil uso para os usuários do meio agrícola que visam utilizar novas tecnologias no seu dia-a-dia, deixando todo o trabalho de cálculos e configurações do Drone para o próprio aplicativo, de forma oculta para o usuário.

3.3 TRABALHOS RELACIONADOS

Com a disponibilização de SDKs pela DJI ao público, novas possibilidades e tecnologias puderam ser desenvolvidos. Manuel [49] demonstram como pode-se desenvolver aplicações móveis para Drones. Segundo o autor, ambientes multimídias e interativos são aqueles que a computação

é usada para imperceptivelmente melhorar as atividades as atividades comuns. Como por exemplo, pode-se citar o uso de um smartphones como meio de transmissão de dados do Drone para o usuário.

Desta forma, Drones tornam-se poderosas plataformas de computação ubíqua, possuem interfaces necessárias para a criação desde pequenos projetos até aplicação mais elaboradas. Pode-se citar uma simples transmissão de vídeo em tempo real para o smartphone, até o planejamento de uma missão autônoma. De maneira simplificada, o autor descreve em seu trabalho como deve ser preparado o ambiente de desenvolvimento de uma aplicação mobile para a plataforma Android, desenvolvida por meio do Android Studio. O resultado é uma simples aplicação que transmite vídeo em tempo real, utilizando o SDK de desenvolvimento da fabricante DJI.

3.4 DESENVOLVIMENTO DO APLICATIVO

O aplicativo Auto Capture consiste em uma ferramenta mobile que visa simplificar o processo de coletas de imagens, de forma que o usuário não precisa se preocupar com cálculos e procedimentos que são feitos na coleta de imagens de modo manual. De forma simplificada o usuário só deve se preocupar com a demarcação do local de voo por meio de *waypoints* inseridos no mapa, altura de voo e a porcentagem da intersecção que as imagens devem possuir ao serem coletadas.

Com o propósito de facilitar o entendimento do uso do aplicativo, na figura 20 é apresentado um diagrama de atividades, exemplificando como o usuário deve proceder para realizar a captura de imagens com o aplicativo.

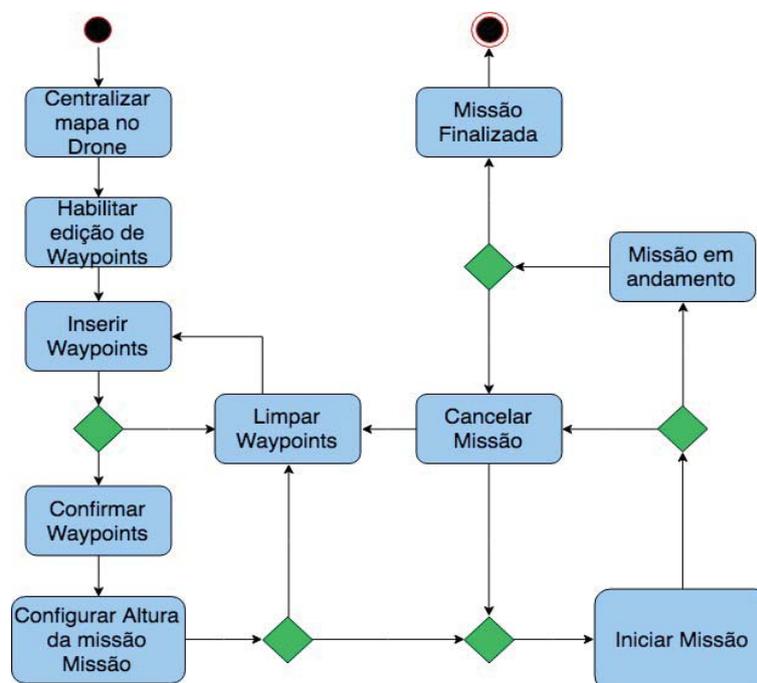


Figura 20. Diagrama de atividades em que o usuário deve seguir para utilizar o aplicativo Auto Capture.

O aplicativo foi desenvolvido para o sistema operacional iOS, utilizando um computador Mac com sistema operacional OS X 10.12 ou posterior, juntamente com sua IDE de desenvolvimento oficial

XCODE®. O mesmo inclui funcionalidades necessárias para projetar, desenvolver e debugar aplicativos iOS. O XCODE também conta com o kit de desenvolvimento de software iOS, que tem como objetivo incluir as ferramentas, compiladores, simuladores e bibliotecas específicas da plataforma [50].

Ele foi desenvolvido na linguagem de programação Swift, o qual possui código aberto sob a licença Apache 2.0 e binários disponíveis para os sistemas operacionais OS X e Linux [51]. Seu desenvolvimento segue o padrão MVC (*Model View Controller*), sendo:

- **Model:** Responsável pela leitura e escrita de dados, como a comunicação com o Drone, e também de suas validações.
- **View:** É responsável pela interação com o usuário, exibe informação como status da missão, além de colher informação na hora de criar uma nova missão.
- **Controller:** É a interface entre o Model e o View, controla toda a lógica do aplicativo, como o cálculo de velocidade de uma missão.

Diferentes bibliotecas e estruturas nativas e de terceiros foram usadas para fornecer todas as funcionalidades necessárias para o aplicativo, estes serão descritos a seguir.

3.4.1 Preparação do ambiente e integração com o DJISDK

A comunicação do aplicativo com o Drone é feito por meio de um SDK (*Software Development Kit*), fornecido pela fabricante DJI [21]. Para este propósito utiliza-se o SDK Mobile, o qual disponibiliza por exemplo, funcionalidades de acesso em tempo real ao vídeo da câmera UltraHD (4K), acesso aos arquivos da câmera, gerenciar recursos de voo e receber informações em tempo real diretamente do controlador de voo do Drone, receber informações da bateria e gerenciar seu correto uso, controlar o Drone por meio de missões pré programadas no aplicativo, acompanhar em tempo real seu desempenho enquanto executa a missão, salvar dados de Latitude e Longitude da imagem capturada, permite também desenvolver um Joystick virtual para fazer o papel de controle no próprio Smartphone, além de inúmeros outras funções disponíveis na documentação do SDK.

Com o ambiente de desenvolvimento configurado, utiliza-se o gerenciador de repositórios Cocoapods [20] para integrar o SDK ao projeto. Primeiramente, instala-se o gerenciador de repositórios Cocoapods (Figura 21), de modo que após o término, deve-se configurar o arquivo *Podfile* (Figura 22), e executa-se os comandos para a instalação e integração do SDK com o projeto (Figura 23).

```
# Xcode 8 + 9
$ sudo gem install cocoapods
```

Figura 21. Instalação do gerenciador de dependências Cocoapods.

```
platform :ios, '8.0'
use_frameworks!

target 'MyApp' do
  pod 'AFNetworking', '~> 2.6'
  pod 'ORStackView', '~> 3.0'
  pod 'SwiftyJSON', '~> 2.3'
end
```

Figura 22. Configuração do arquivo Podfile do projeto no XCODE.

Now you can install the dependencies in your project:

```
$ pod install
```

Make sure to always open the Xcode workspace instead of the project file when building your project:

```
$ open App.xcworkspace
```

Now you can import your dependencies e.g.:

```
#import <Reachability/Reachability.h>
```

Figura 23. Instalação e integração do SDK com o projeto.

A utilização do gerenciador de dependências Cocoapods facilita a configuração do projeto pois cria-se um novo inicializador do projeto com todas as configuração para o uso do SDK, poupando tempo e trabalho para o desenvolvimento do aplicativo.

Para utilizar o SDK, deve-se ter um cadastro ativo no site de desenvolvedor DJI [52]. Como apresenta-se na Figura 24, é criado um novo *app*, colocando o *Bundle Identifier* do projeto, categoria, descrição e plataforma do projeto, em seguida ele disponibiliza uma chave de acesso (AppKEY) que deverá se utilizada no aplicativo para ter acesso as funcionalidades do SDK. Desta forma o projeto está pronto para iniciar o desenvolvimento utilizando todos os recursos que o DJISDK dispõe.

APP INFORMATION

| | |
|-------------------|--|
| SDK Type | Mobile SDK |
| APP Name | Auto Capture |
| Software Platform | iOS |
| Bundle Identifier | br.upf.Auto-Capture |
| App Key | 720c51d1019812b7f05a908e |
| Category | Agricultural applications |
| Description | Aplicativo para mapear áreas agrícolas |

EDIT

DELETE

Figura 24. Chave de acesso para a utilização do SDK

3.4.2 Características e interface do Aplicativo

A interface de usuário foi desenvolvida de uma maneira limpa e intuitiva, podendo o usuário efetuar seu trabalho em poucos passos. A barra superior do aplicativo (Figura 25) contém informações do modo de voo, quantidade de satélites GPS em que o Drone está sincronizado, velocidade vertical e velocidade horizontal, altitude da aeronave e também a quantidade de bateria que resta no Drone. Na parte inferior, existem apenas dois principais botões, sendo que o primeiro da direita para a esquerda, em formato de avião, tem a finalidade de centralizar o mapa aonde está o Drone, poupando este trabalho ao usuário. Já o segundo tem como finalidade abrir as ferramentas de edição de um novo plano de voo.

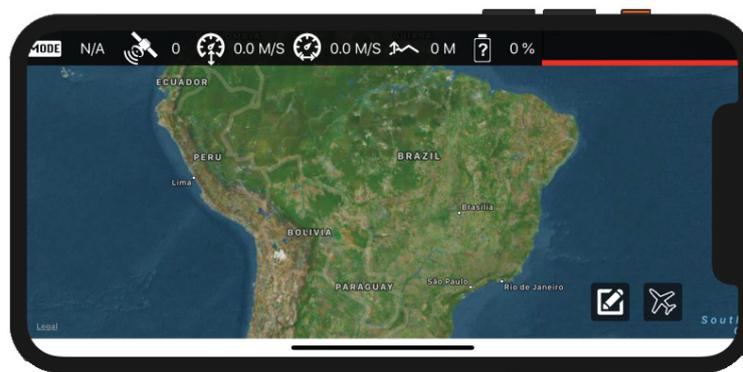


Figura 25. Tela inicial do aplicativo Auto Capture.

Ao abrir o editor de plano de voo, alguns alertas são disparados ao usuário, informando qual padrão deve ser seguido na inserção dos *waypoints* (Figura 26(a)), assim como a quantidade mínima de *waypoints* a serem inseridos para criar um plano de voo válido. Em seguida, outros três botões são mostrados ao usuário, inserção de *waypoints* no mapa, limpar os *waypoints*, e o último para fazer a configuração do plano de voo. Na figura 26(b), apresenta-se uma área com um plano de voo inserido no mapa.

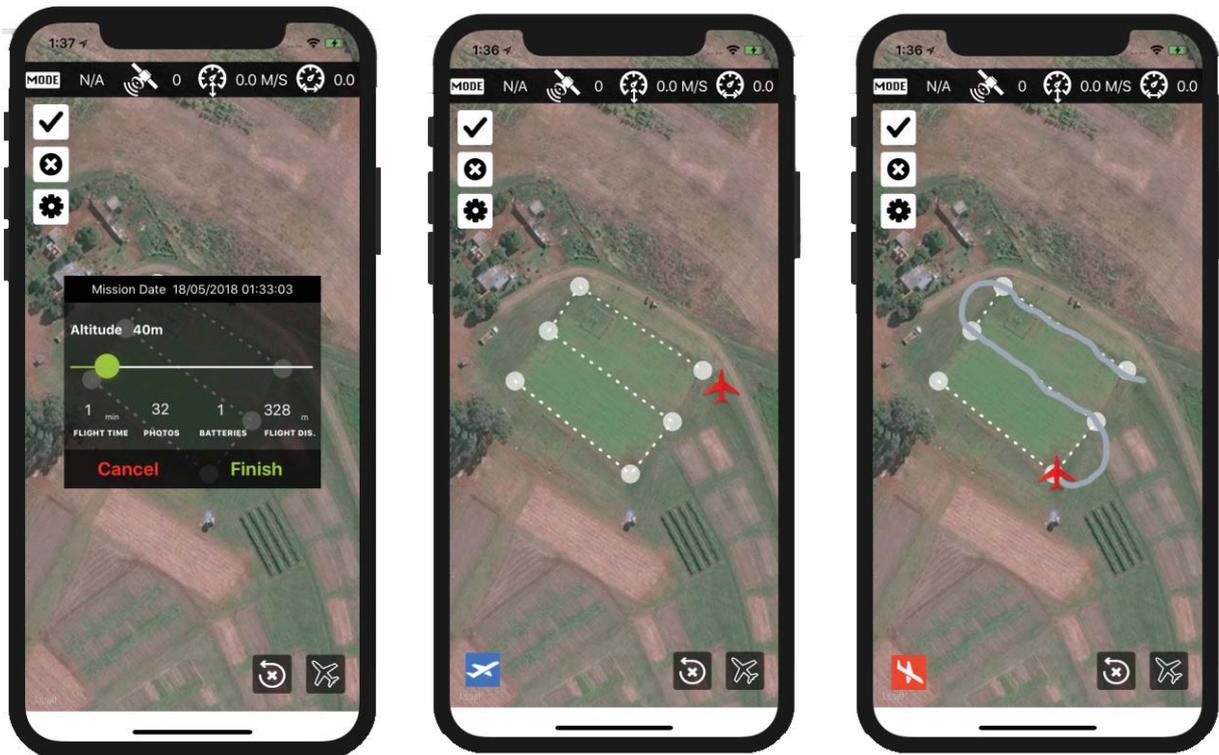


(a) Informações apresentadas ao usuário na criação de uma nova missão. (b) *Waypoints* inseridos em um cultivar de Trigo.

Figura 26. Tela de configuração de uma nova missão no Auto Capture.

Quando a inserção de *waypoints* no mapa for concluída, pode-se abrir as configurações da missão, como apresentada na Figura 27(a). Esta janela possui um *slider* ajustável, permitindo selecionar a altura do voo, de modo que o restante é configurado automaticamente, sendo eles: Tempo de voo estimado; Número de fotos capturadas; Baterias necessárias para completar a missão e distância total percorrida. Para finalizar a configuração da missão, utiliza-se o botão FINISH, apresentando um botão azul de início de missão, como apresentada na Figura 27(b), de modo que quando iniciada, é alterado para vermelho, podendo ser utilizado para cancelar a missão.

Com o Drone ligado e configurado de maneira correta, após o término da configuração acima descrita e nenhuma mensagem de erro mostrada ao usuário, pode-se iniciar a captura das imagens. Na figura 27(c), é apresentada uma missão em andamento, podendo ser cancelada utilizando o botão vermelho. Ao fim do processo de captura, o Drone volta automaticamente para o local de onde iniciou a missão.



(a) Janela de configuração da missão.

(b) Missão pronta para iniciar.

(c) Missão em andamento, podendo ser cancelada usando o botão vermelho.

Figura 27. Telas do aplicativo Auto Capture.

3.4.3 Desenvolvimento das funcionalidades do aplicativo

Todo o controle de mapa e sua apresentação foram desenvolvidos utilizando o MapKit, disponibilizado pela própria Apple® no conjunto de bibliotecas nativas do iOS. Na figura 28, é apresentado a função "add"(linha 19), a qual é responsável por coletar os marcadores inseridos no mapa e armazená-los em um vetor que posteriormente será transformado em *waypoints* e enviados ao Drone, tornando-se a rota que o mesmo deverá cumprir. Na linha 31, apresenta-se a função "clear" com o objetivo de remover os marcadores da tela, além de remove-los do vetor de *waypoints*.

```

16  /**
17  * Add Waypoints in Map View
18  */
19  func add(_ point: CGPoint, with mapView: MKMapView) {
20      let coordinate: CLLocationCoordinate2D = mapView.convert(point, toCoordinateFrom: mapView)
21      let location = CLLocation(latitude: coordinate.latitude, longitude: coordinate.longitude)
22      editPoints.append(location)
23      let annotation = MKPointAnnotation()
24      annotation.coordinate = location.coordinate
25      mapView.addAnnotation(annotation)
26  }
27
28  /**
29  * Clean All Waypoints in Map View
30  */
31  func cleanAllPoints(with mapView: MKMapView) {
32      var annotation = mapView.annotations
33      for i in 0..

```

Figura 28. Função responsável por armazenar os marcadores inseridos no mapa e armazená-los em um vetor.

As bibliotecas responsáveis pela comunicação com o Drone são providas pelo DJISDK (DJI Software Development Kit) [21]. Uma das principais características do aplicativo desenvolvido é a captura automática de imagens destacado na Figura 29 é apresentado um trecho de código responsável pelo upload de *waypoints* (linha 661). Esta ação é realizada por uma "completion" a qual retorna algum tipo de erro, podendo ele ser tratado ou não. Neste caso se o retorno do erro for diferente de *nil*, o upload da missão falhou, caso contrário uma mensagem de sucesso é disparada na tela do aplicativo.

```

651     waypointMission?.maxFlightSpeed = CFloat(10)
652     waypointMission?.autoFlightSpeed = CFloat(self.flightVelocity)
653     waypointMission?.headingMode = .auto
654     waypointMission?.finishedAction = .goHome
655     waypointMission?.rotateGimbalPitch = true
656     waypointMission?.gotoFirstWaypointMode = .safely
657     waypointMission?.flightPathMode = .curved
658
659     missionOperator().load(waypointMission!)
660
661     missionOperator().uploadMission(completion: {(_ error: Error?) -> Void in
662         if error != nil {
663             let failedMessage: String? = "Upload Mission Failed: \(error.debugDescription)"
664             self.showAlertView(withTitle: "Mission Status", withMessage: failedMessage!)
665         }
666         else {
667             self.showAlertView(withTitle: "Mission Status", withMessage: "Upload Mission Finished")
668             self.statusColor.backgroundColor = UIColor.green
669             self.startBtn.isHidden = false
670             self.stopBtn.isHidden = true
671             self.addBtn.isHidden = true
672             self.clearBtn.isHidden = true
673             self.configBtn.isHidden = true
674         }
675     })

```

Figura 29. Trecho de código responsável pelo upload de *waypoints* para o Drone.

Na linha 651 a 657, os atributos do objeto *waypointMission* são configurados com parâmetros pré-definidos, de modo que na linha 659, a função *missionOperator().load* carrega estes parâmetros para o Drone.

Todos componentes visuais que estão presentes na *View* devem ser referenciadas no *Controller*. Como exemplo, botões devem ser do tipo "IBAction", ou seja, ao serem clicados executam um trecho de código atribuído a nele. Na figura 30 apresenta-se referência a alguns botões do aplicativo do tipo "IBAction".

```

152
153     @IBAction internal func centralizarMapa(_ sender: Any)
154
155     @IBAction internal func editBtnAction(_ sender: Any)
156
157     @IBAction internal func addBtnAction(_ sender: Any)
158
159     @IBAction internal func clearBtnAction(_ sender: Any)
160
161     @IBAction internal func configBtnAction(_ sender: Any)
162
163     @IBAction internal func altSliderAction(_ sender: Any)
164
165     @IBAction internal func startBtnAction(_ sender: Any)

```

Figura 30. Referência dos botões e tipo de ação que será tomada por eles.

Com o intuito de acompanhar informações em tempo real durante a execução de uma missão, o SDK disponibiliza a função *missionOperator().addListener(ToExecutionEvent)* (Figura 31). Como é apresentado na linha 751, a variável *waypoint*, fica recebendo em tempo real qual será o próximo *waypoint* que o Drone passara. Informações como altura atual, velocidade horizontal, velocidade vertical, nível de bateria, modo de operação do Drone e quantidade de satélites de GPS que estão presentes na barra superior do aplicativo são obtidas dentro desta função.

```

748     missionOperator().addListener(toExecutionEvent: self, with: DispatchQueue.main)
749     { (DJIWaypointMissionExecutionEvent) in
750
751         let waypoint = self.missionOperator().latestExecutionProgress?.targetWaypointIndex
752         self.statusGeralLabel.text = "\(String(describing: waypoint))"
753     }

```

Figura 31. Função *missionOperator().addListener(ToExecutionEvent)*, responsável por alimentar a tela do aplicativo com informação do Drone em tempo real.

A posição atual do Drone no mapa (Figura 27(c)) também é recebida por meio desta função, sendo que após o recebimento da coordenada geográfica, a posição do marcador (Drone), no mapa é atualizada constantemente, podendo assim o usuário acompanhar em tempo real a posição do seu equipamento.

Outra função implementada foi a *missionOperator().addListener(ToFinished)*, a qual tem como objetivo, verificar se o Drone terminou a missão (chegou ao último *waypoint*). A Figura 32 apresenta o trecho de código responsável por detectar este fim de missão. Se bem sucedida, irá disparar mensagens de sucesso ao usuário e parar a captura de imagens. Estas duas funções *listener* apresentadas

aqui, são iniciadas após o botão de início de missão ser ativado, sendo que só irão parar de monitorar o Drone, após que o mesmo estiver em solo e desarmado.

```

677     missionOperator().addListener(toFinished: self, with: DispatchQueue.main,
678                                 andBlock: {(_ error: Error?) -> Void in
679         if error != nil {
680             let failedMessage: String? = "Mission Execution Failed: \(error.debugDescription)"
681             self.showAlertView(withTitle: "Mission Status", withMessage: failedMessage!)
682         }
683     }
684     else {
685         self.showAlertView(withTitle: "Mission Status", withMessage: "Mission Execution Finished")
686         self.stopCapture()
687         self.startBtn.isHidden = true
688         self.stopBtn.isHidden = true
689         self.addBtn.isHidden = false
690         self.clearBtn.isHidden = false
691         self.configBtn.isHidden = false
692     }
693 }
694 })
695

```

Figura 32. Função `missionOperator().addListener(ToFinished)`, responsável verificar se o Drone chegou ao seu ultimo *waypoint*.

3.5 USO DO APLICATIVO NO MEIO AGRÍCOLA.

Para a correta obtenção de imagens, faz-se necessário manter uma sobreposição (*overlapping*) entre as imagens coletadas. Segundo Ferreira [53] para obter-se um bom resultado final no processamento das imagens deve-se respeitar uma sobreposição lateral (*sidelap*) de 60% e uma sobreposição longitudinal (*frontlap*) de 70% em média.

Com o intuito de fazer ajustes no aplicativo e chegar ao *overlapping* ideal, testes foram feitos em um campo experimental de trigo situado na cidade de Coxilha-RS, pertencente a EMBRAPA TRIGO (Lat: 28.186418 Long: -52.325192). Após alguns ajustes de posicionamento, velocidade horizontal de voo e intervalo de disparo da câmera do Drone, chegou-se ao resultado apresentado na figura 33. Estas duas imagens sobrepostas capturadas em sequencia, possuem um *overlapping* de aproximadamente 70%.



Figura 33. Duas imagens vizinhas com um *overlapping* de 70% representado pelo retângulo tracejado.

Como a área a ser monitorada possui um tamanho reduzido, as coletas foram pensadas de maneira que o mosaico final da plantação seja em uma alta resolução, o que traz maior confiabilidade. O primeiro ponto a ser abordado é a altura das coletas, sendo que as mesmas foram efetuadas com uma média de 30 metros em relação ao solo. Outro ponto é a velocidade, que como demonstrada anteriormente é calculada automaticamente pelo aplicativo, desta maneira quanto menor a altitude, menor a velocidade horizontal do Drone. Como resultado final, cada pixel das imagens coletadas obtiveram um tamanho de 0,04cm x 0,04cm capturados do solo, que se comparado a uma imagem de satélite, pode chegar até 400 metros quadrados por pixel.

3.5.1 Tipos de Drones e equipamentos utilizados nos testes

Para permitir o uso do aplicativo com diferentes veículos de voo, testou-se a solução proposta com dois Drones DJI, com características e equipamentos diferentes acoplados a eles.

O primeiro Drone utilizado foi o DJI Phantom 3 Professional (Figura 34), que tem o propósito de agilidade e facilidade operacional na captura de imagens RGB. Este equipamento apresenta sensores eletrônicos de bússola, acelerômetro com controladora eletrônica de velocidade, giroscópio de seis eixos e sistema GNSS (GPS e Glonass) de alta sensibilidade. É equipado com câmera RGB Sony EXMOR de 12,4 megapixel de 20 mm (formato equivalente 35 mm), acoplada a um gimbal de três eixos para estabilização eletrônica [54].



(a) Drone DJI Phantom 3 Professional sendo utilizado no campo experimental da Embrapa Trigo



(b) Drone DJI Phantom 3 Professional [54]

Figura 34. DJI Phantom 3 Professional.

Como parte deste trabalho, foi utilizado um Drone montado e configurado no grupo de pesquisas MOSAICO UPF. Denominado Drone DJI DIY, Figura 35(a) possui frame de fibra de carbono com braços retráteis, 4 motores DJI 930kv, ESCs DJI 30a, controladora NAZA V2 with Datalink bluetooth, além de possuir um gimbal que pode suportar diversos tipos de sensores, possibilitando capturar diferentes tipos de imagens em uma plantação. Na figura 35(b), apresenta-se o Drones montado para este trabalho.



(a) Drone DJI DIY em uma das coletas efetuadas no campo experimental de trigo da Embrapa Trigo.

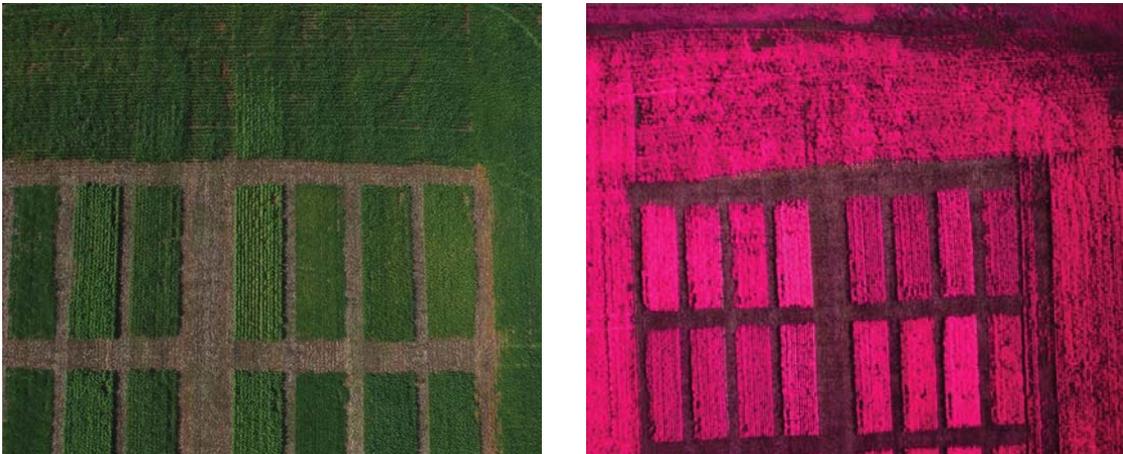


(b) Vista superior do Drone DJI DIY com os braços recolhidos. Este tipo de frame pode facilitar seu transporte.

Figura 35. Drone DJI DIY montado no grupo de pesquisas Mosaico UPF.

Acoplado ao Drone DJI DIY está um sensor multispectral Tetracam ADC (*Agriculture Digital Câmera*), a qual consegue capturar imagens com comprimento de onda de 250nm da faixa visível e 920nm da faixa de infravermelho próximo. A câmera tem um peso de 90 gramas e sua dimensão é de 75 mm × 59 mm × 33 mm, sendo assim torna-se uma câmera adequada para uso em quase todos os

tipos de plataforma de Drones existentes. Na figura 36(a), apresenta-se uma imagem RGB obtida por meio do Drone DJI Phantom 3 Professional e na Figura 36(b), apresenta-se uma imagem capturada por meio do Drone DJI DIY e câmera Tetracam ADC.



(a) Imagem RGB obtida por meio do sensor do Drone DJI Phantom 3 Professional.

(b) Imagem Obtida por meio do sensor NDVI acoplado ao Drone DJI DIY.

Figura 36. Imagem provenientes dos sensores dos Drones.

3.6 CONCLUSÕES

Com os testes efetuados em diferentes datas na EMBRAPA TRIGO, pode-se chegar a conclusão que o aplicativo conseguiu chegar ao objetivo de efetuar coletas de imagens de forma autônoma e simples. Outro ponto que agrega valor ao aplicativo é a exatidão no ponto de coleta de cada imagem, onde foram obtidos bons resultados, sendo que cada imagem teve uma média próxima a desejada de 70% de *overlapping* com sua vizinha.

Algumas limitações foram encontradas durante o uso do MapKit da Apple, o uso de marcadores no mapa é limitado. Como solução a isso e também como trabalho futuro, sugere-se o uso da API do Google Maps, a qual possui um leque maior de funcionalidades utilizando marcadores em um Mapa. Outro fator a ser estudado futuramente é o desenvolvimento deste aplicativo para a plataforma Android, abrangendo assim a maioria de aparelhos smartphones disponíveis no mercado.

O crescimento do mercado atual de Drones pode ocasionar uma vinda rápida de novos modelos para o mercado, principalmente da empresa DJI. Com o intuito de poder utilizar novos modelos de Drones, o uso de novas funcionalidades da DJI SDK pode ser considerada como trabalho futuro, podendo ampliar as opções de uso para o usuário final.

4. APLICATIVO PARA MOSAICAGEM DE IMAGENS DIGITAIS

4.1 RESUMO

O aumento na utilização de Drones no meio agrícola tem se mostrado uma ótima ferramenta no monitoramento de cultivares. Para isso, automatizar o processamento destas imagens de maneira simples, de forma que não seja necessário a compra ou utilização de softwares específicos de GIS (*Geographics Information Systems*), torna-se importante para o diagnóstico e tomada de decisão do agricultor em tempo hábil. Desta forma este trabalho apresenta o desenvolvimento de uma ferramenta Desktop para a mosaicagem de imagens RGB e NDVI. A Ferramenta foi desenvolvida na linguagem C++ utilizando o *framework* OpenCV.

4.2 INTRODUÇÃO

Segundo Rong [1] a construção de um mosaico de imagens é um processo que consiste em sobrepor uma imagem sobre a outra, levando em consideração pontos em comum, gerando assim um mosaico, o que amplia o campo de visão da região de interesse.

A construção de mosaicos e o seu uso na topografia vem se mostrando importante há bastante tempo [3]. Desde os primeiros métodos de mosaicagem muitas técnicas e equipamentos fotográficos foram criados e melhorados [4]. Deste modo a utilização de equipamentos e o processamento de dados para mapeamento aéreo faz uso de múltiplas estratégias e tecnologias, incluindo uma série de passos e trabalhos a serem efetuados para poder chegar ao produto final de um mapeamento, um mosaico em alta resolução.

Deste modo a utilização de mosaicos pode se tornar importante quando necessita-se fazer o mapeamento de grandes áreas, mas para tal, precisa-se capturar várias fotos. Devido a vários fatores como tamanho de pixel, altura de onde ela foi capturada, uma única imagem não consegue cobrir grandes áreas. Outras aplicações que pode-se citar é aumentar o campo de visão de um UAV (Veículo aéreo não Tripulado) em tempo real, utilizando mais do que uma câmera, e por fim gerar uma imagem só destas câmeras, em um grande mosaico.

Com a constante evolução da computação e das técnicas empregadas na construção de mosaicos alguns padrões foram sendo mais usados. Deste modo, segundo Wolf e Dewitt [9] a criação de mosaicos provenientes de fontes aéreas geralmente podem ser divididos em três classes: Mosaicos Controlados, Mosaicos Não Controlados e Mosaicos Semi-controlados. Neste trabalho foi utilizada a técnica de Mosaicos Não Controlados, sendo que a detecção de pontos chaves para a posterior sobreposição, utilizou de recursos do *framework* OpenCV.

4.3 TRABALHOS RELACIONADOS

Tarallo [55] discute formas alternativas aos dos softwares proprietários para criar mosaicos de imagens digitais com um resultado satisfatório. Segundo o autor, utilizando a biblioteca de visão computacional OpenCV, é possível criar uma ferramenta de mosaicação automática.

No cenário descrito por ele, a forma manual de mosaicar imagens consiste em obter o modelo digital do terreno, fazer a ortorretificação de imagens e colocação manual de bandeiras marcadoras, que são usadas para que um software possa reconhecer e construir um mosaico deste terreno. Utilizando este tipo técnica, a construção de mosaicos pode ser demorada e trabalhosa.

A solução proposta, consiste em utilizar uma técnica chamada SIFT (*Scale Invariant Feature Transform*), sendo uma alternativa para a não utilização da forma manual de mosaicação. O método SIFT consiste em transformar uma imagem em uma coleção de vetores de características locais (descritores de características), e cada um desses vetores são invariantes à escala, rotação e parcialmente invariante à mudanças de iluminação e ponto de vista. Esta técnica é considerada como um mosaico não controlado, onde as imagens são combinadas por características comuns entre elas.

O primeiro estágio do SIFT é a Detecção de extremos no Espaço Escala. Neste primeiro estágio é feito a procura por todas as escalas e locais de uma imagem, onde para isto é utilizada uma função conhecida como diferença da matriz Gaussiana, servindo como identificação dos potenciais pontos de interesse que são invariantes à escala e orientação. Esta é a parte que mais demanda poder computacional do algoritmo. A segunda etapa é a localização dos pontos chaves (*KeyPoints*), onde para cada local na imagem que foi selecionado como possível *KeyPoints* é determinada sua posição e escala, e, por fim, eles realmente são selecionados baseados em medidas de sua estabilidade. A terceira etapa é chamada de Definição da Orientação. Uma ou mais orientações são atribuídas para cada ponto chave localizado, baseada em direções do gradiente. Todas as operações posteriores são realizadas sobre os dados da imagem que foram considerados ponto chave e que foram transformados em relação à orientação, escala e localização, proporcionando invariância a estas transformações. Por fim, a etapa Descritor dos Pontos Chave é responsável por mensurar os gradientes na escala selecionada na região ao redor de cada ponto chave, sendo criados histogramas de orientações para compor o descritor.

Outros autores demonstram outras técnicas para a mosaicação não controlada semelhantes ao SIFT, o método SURF (*Speeded Up Robust Features*). Este é um detector de características invariante à escala de rotação. Semelhante ao SIFT, ele também é baseado na teoria do espaço de escala, mas no entanto, utiliza a matriz Hessiana de toda a imagem para estimar os máximos locais em diferentes espaços de escala [56].

Existem outros métodos discutidos em alguns trabalhos, sendo um deles os mosaicos baseados em característica de baixo nível. Este método não exige que as imagens tenham uma grande área de sobreposição entre elas, pois com a detecção de alguns pontos já é possível fazer a mosaicação. As características detectadas por este método são bordas, canto, pixel, cor, dentre outras [57]. Este método inclui o detector de cantos FAST, o qual, detecta cantos em comum nas imagens.

4.4 DESENVOLVIMENTO DO SOFTWARE

O software denominado Auto Mosaicing tem como objetivo mosaicar imagens digitais de maneira simples e rápida, resultando em um mosaico de imagens digitais com qualidade suficiente para detectar como por exemplo, falhas de plantação, má desenvolvimento da planta, níveis de biomassa, entre outros. Desenvolvido e compilado para a plataforma MacOS, ele também poderá ser compilada para outras plataformas como Windows e distribuições Linux diversas, pois seu código, desenvolvido em C++ permite esta portabilidade, bastando apenas o usuário ter instalado e configurado em seu computador um compilador GCC com o *framework* OpenCV instalado e suas bibliotecas ligadas ao projeto.

4.4.1 Obtenção do Mosaico com a stitching API

A criação de um mosaico envolve basicamente quatro etapas de processamento: Registro, Re-projeção, Stitching e Mesclagem. Embora o estado da arte indique avanços nessa área de pesquisa nos últimos anos, o mosaico de imagens continua a ser um desafio devido a fatores que podem mudar de um cenário para outro, aonde provavelmente não poderá ser utilizado um mesmo modelo para todos os cenários [58]. A seguir, cada uma destas etapas são brevemente descritas:

- **Registro:** Para registrar um conjunto de imagens, é necessário estimar as transformações geométricas que alinham as imagens de acordo com uma imagem de referência dentro deste conjunto. O conjunto pode consistir de duas ou mais imagens tiradas de uma única cena em momentos diferentes, a partir de pontos de vistas diferentes.
- **Re-projeção:** Refere-se ao alinhamento das imagens para um sistema de coordenadas comum utilizando as transformações geométricas calculadas.
- **Stitching:** O objetivo desta etapa é sobrepor as imagens alinhadas em uma composição maior, combinando valores de pixels das partes sobrepostas e retendo pixels aonde não ocorre sobreposição.
- **Mesclagem:** Diferenças fotométricas globais frequentemente podem ocorrer entre imagens em uma sequência, resultando na visibilidade da emenda entre elas. Esta etapa é utilizada para minimizar esse efeito e homogenizar a aparência global do mosaico.

Toda documentação do *framework* está disponível no site do OpenCV [25], juntamente com exemplos de implementações. Na Figura 37 é apresentado o código para criação de um panorama utilizando a classe *Stitcher*.

```

1  #include <iostream>
2  #include <fstream>
3  #include "opencv2/imgcodecs.hpp"
4  #include "opencv2/highgui.hpp"
5  #include "opencv2/stitching.hpp"
6  using namespace std;
7  using namespace cv;
8  bool try_use_gpu = false;
9  Stitcher::Mode mode = Stitcher::SCANS;
10 vector<Mat> imgs;
11 string result_name = "result.jpg";
12 int main()
13 {
14     Mat pano;
15     Ptr<Stitcher> stitcher = Stitcher::create(mode, try_use_gpu);
16     Stitcher::Status status = stitcher->stitch(imgs, pano);
17     if (status != Stitcher::OK)
18     {
19         cout << "Can't stitch images, error code = " << int(status)
20             << endl;
21         return -1;
22     }
23     imwrite(result_name, pano);
24     return 0;
25 }

```

Figura 37. Código para criação de um panorama utilizando a classe *Stitcher*

Na linha 14 da Figura 37, é declarada uma variável do tipo *Mat* que será armazenada o panorama final, sendo um tipo de variável disponibilizado pela classe *Stitcher*. Na linha 15 é criado um objeto "create", passando para ele o atributo "mode", aonde pode ser PANORAMA (*imagens lado a lado*) ou SCANS (*imagens dispersas para todos os lados*). O segundo atributo "try_use_gpu" é responsável por habilitar ou não o processamento em GPU (*Graphics Processing Unit*) se disponível no hardware utilizado para o processamento.

Na linha 16 o objeto "status" recebe o retorno do processamento do objeto "stitch", sendo que este por sua vez, tem como entrada um Array de imagens, e uma variável para armazenar o resultado do processamento. Nesta etapa, o código realiza todas as etapas descritas anteriormente de forma automática.

Este código disponibilizado na documentação pode retornar bons resultados se a entrada de imagens não exceder a quantidade de 5 a 8 imagens de entrada no modo "SCANS". Na figura 38 é apresentado um exemplo de saída utilizando 5 imagens de entrada no modo "PANORAMA", aonde a sequência das imagens já está definida, o que facilita o trabalho do algoritmo.

```
./cpp-example-stitching --mode panorama <path to opencv_extra>/testdata/stitching/boat*
```



Figura 38. Exemplo de saída disponível na documentação do OpenCV.

Testes foram efetuados com este mesmo código utilizando as imagens coletadas anteriormente com o aplicativo descrito no Capítulo 3. O resultado obtido é apresentado na Figura 39. Como a coleta foi efetuada de maneira que a sobreposição das imagens formavam uma matriz, utilizou-se o modo SCANS.



Figura 39. Resultado do processamento de imagens de um campo no código de exemplo 1.

Como pode ser observado na figura 39, o algoritmo não resultou uma correta mosaicagem das imagens, contendo várias falhas de costuras em sua construção. Isso deve-se a falha no encontro de características, sendo esta pertencente á a etapa de registro do mosaico. Com o intuito de entender melhor o funcionamento do algoritmo, a seguir é demonstrado o pipeline do algoritmo [25].

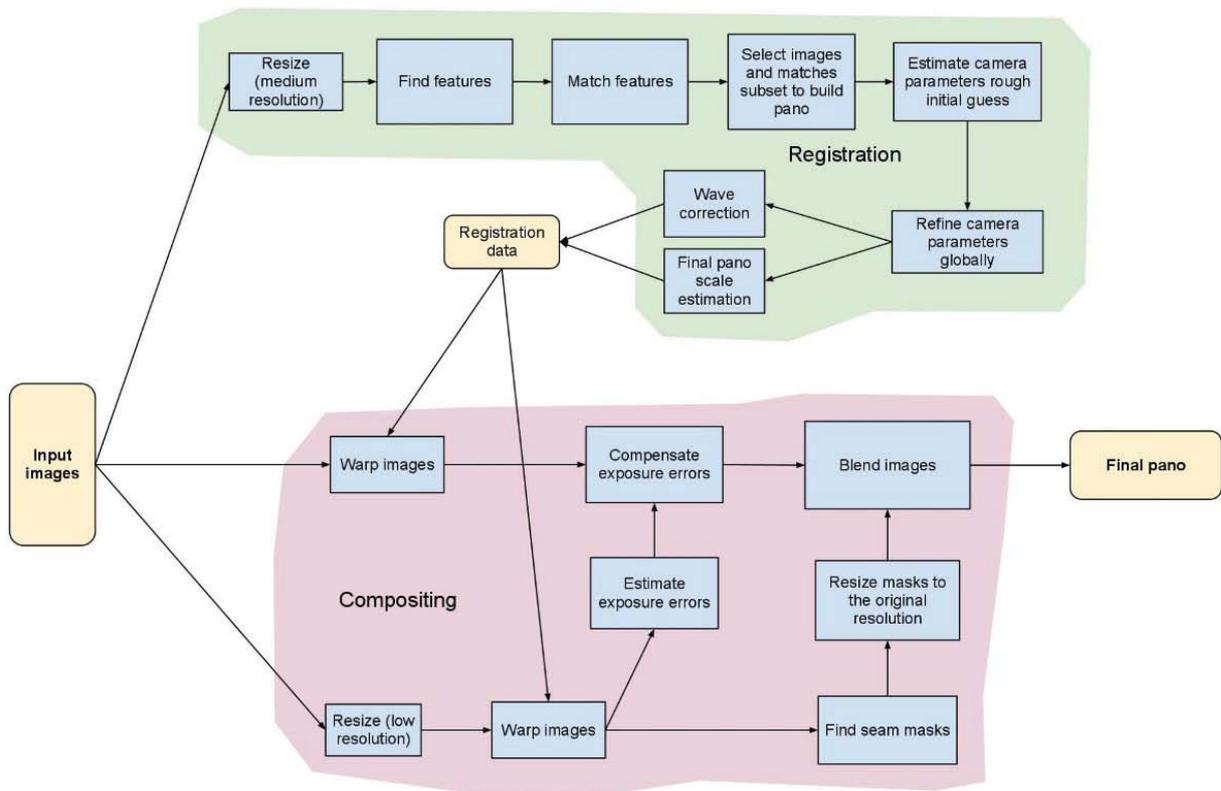


Figura 40. Pipeline do Stitching API [59].

A figura 40 ilustra o pipeline do módulo de costura implementado na classe *Stitcher*. Usando essa classe é possível configurar ou remover algumas etapas, ou seja, ajustar o encadeamento de acordo com as necessidades específicas. Todos os blocos de construção do pipeline estão disponíveis na sessão "*detail*", podendo combiná-los e usá-los separadamente. O pipeline de costura apresentado é muito semelhante ao proposto pelo autor Brow et. al [60].

4.4.2 Otimização da Mosaicagem

O algoritmo utilizado estava falhando em uma das etapas do pipeline de costura, sendo ela a segunda (*Find Features*). Isso deve-se ao fato de que o algoritmo procura por características em todo o espaço da imagem, o que poderia por coincidência, marcar pontos que na verdade não são os mesmos em outra imagem, causando um efeito em cascata em todo o restante do pipeline. Na figura 39 pode ser observado a falta de partes do mosaico.

Com o intuito de melhorar a construção de panoramas utilizando esta funcionalidade da *Stitcher* class e seguindo a solução de outros autores, como a de Ferreira [53], aonde o *Overlapping* entre duas imagens deve ser de aproximadamente 70%, foi desenvolvido uma função em que é delimitada a área em que deve-se obter as características entre duas imagens. Como proposta para uso deste software, deve-se utilizar o aplicativo proposto por este trabalho para fazer a coleta de imagens, pois

o mesmo, realiza a coleta utilizando como *Overlapping* 70% em média, característica importante para o correto funcionamento do algoritmo de mosaicagem.

```

1
2  int VALUE = 70;
3  int OVERLAP = imgs.at(0).rows*VALUE/100;
4  int IMGSIZE = imgs.at(0).rows*50/100;
5
6  vector<vector<Rect>> rois;
7  vector<Rect> roisVect;
8  //Images of the columns
9  for (int j = 0; j < imgs.size(); j++) {
10     //First top image Rect(x,y,width,height)
11     if (j == 0) {
12         Rect rect1 = Rect(0, 0, imgs.at(j).cols, IMGSIZE);
13         cout << rect1.size() << endl;
14         roisVect.push_back(rect1);
15         rois.push_back(roisVect);
16         roisVect.clear();
17
18     //Last image
19     } else if (j == imgs.size() - 1) {
20         Rect rect2 = Rect(0, OVERLAP, imgs.at(j).cols, IMGSIZE);
21         cout << rect2.size() << endl;
22         roisVect.push_back(rect2);
23         rois.push_back(roisVect);
24         roisVect.clear();
25
26     //Image between first and last
27     } else{

```

Figura 41. Técnica de retângulo da área de interesse, onde o algoritmo determina a região da imagens aonde deve-se procurar os pontos em comum para a mosaicagem.

A solução encontrada para este problema, foi a criação da função na Figura 41, aonde seu principal objetivo é delimitar a área em que existe um *Overlapping* verdadeiro. Desta maneira o algoritmo sabe que somente nesta área deve-se fazer a procura de características entre duas imagens, poupando processamento, tempo de execução e deixando o algoritmo mais assertivo para gerar um mosaico de imagens.

O código apresentado utiliza um conceito chamado ROI (rectangular region of interest (ROI)), ou seja, retângulo da área de interesse. Esse ROI é criado utilizando exatamente a área em que existe uma sobreposição (*Overlapping*) entre duas imagens. Observando o código criado, na linha 3, é definido qual a porcentagem utilizada no *Overlapping*, nas linhas 6 e 7, são criado os vetores chamados *rois*, aonde serão armazenados as posições das linhas e colunas da imagem que contém o *Overlap-*

ping. Após, utilizando um laço de repetição, é percorrido imagem por imagem do vetor de imagens que anteriormente foi carregado para o programa. Neste laço, por meio de testes condicionais, é identificado qual a posição da imagem no mosaico, e então é criada uma área de interesse de 70% e armazenado em um vetor de *ROIS*. Para um melhor entendimento, na figura 42 é apresentado um diagrama que demonstra quais as áreas da imagem são utilizadas para a criação do *ROI*.

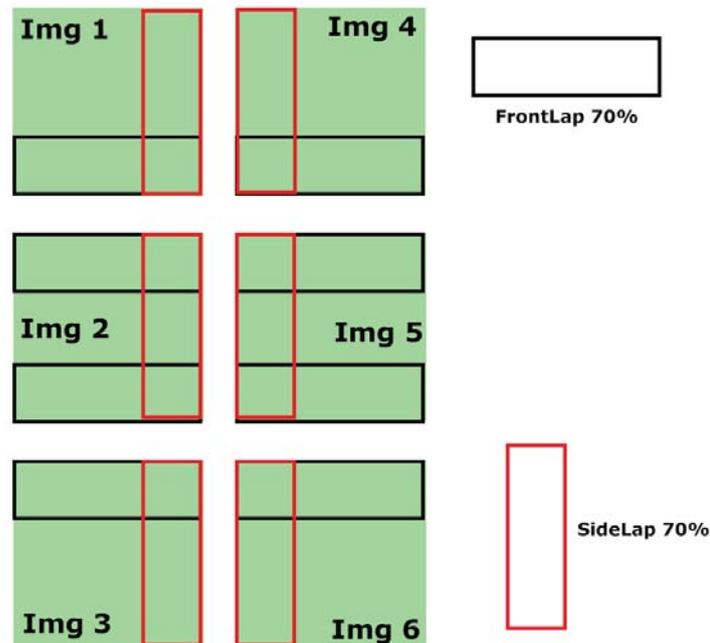


Figura 42. Áreas de cálculo do ROI em cada uma das imagens, levando em consideração sua posição. (*O tamanho dos retângulos não corresponde á 70% da imagem, é meramente uma ilustração de sua posição dentro da imagem.)

Como a captura das imagens é feita de maneira em que o *Overlapping* corresponda a 70% e sua posição no vetor de imagens já está definido, o algoritmo sabe aonde calcular a área de interesse (ROI) de cada imagem. Como observado na figura 42, quando carregada as imagens para o vetor, o algoritmo sabe que o mosaico deve ser formado por 6 imagens dispostas em uma matriz 2x3, de forma que a primeira imagem possui dois pontos de sobreposição (sidelap: img 4 e frontlap: img 2), e assim sucessivamente até a última imagem.

Após o termino deste processo, o algoritmo comporta-se da mesma maneira que foi demonstrada anteriormente, com exceção da linha 3 do algoritmo apresentado na Figura 43. Além do vetor de imagens e uma variável para armazenar o resultado do processamento, é incluído o vetor de ROIS anteriormente processado. Com isso os passos seguintes do algoritmo são verificar se existe algum retorno de erro do processamento e gravar o resultado final na pasta de destino.

```

1  Mat pano;
2  Ptr<Stitcher> stitcher = Stitcher::create(mode, try_use_gpu);
3  Stitcher::Status status = stitcher->stitch(imgs, rois, pano);
4  cout << endl << status;
5  cout << endl << pano.size();
6
7  if (status != Stitcher::OK)
8  {
9      cout << "Can't stitch images, error code = " << int(status) <<
10         endl;
11     return -1;
12 }
13 imwrite(result_dest, pano);
14 return 0;

```

Figura 43. Chamada das funções de mosaicagem e encontro dos ROIs.

O software desenvolvido foi compilado para a plataforma MacOS, sem interface gráfica, sendo sua execução feita por meio do terminal. Para a criação de um mosaico de imagens, deve-se enviar as imagens coletadas utilizando o Aplicativo desenvolvido no capítulo 3, evitando falhas na construção do mosaico. A pasta do aplicativo contém o arquivo binário compilado e duas pastas, IN e OUT, aonde deverão ser colocadas as imagens de entrada (IN) e o resultado será armazenado na pasta (OUT).

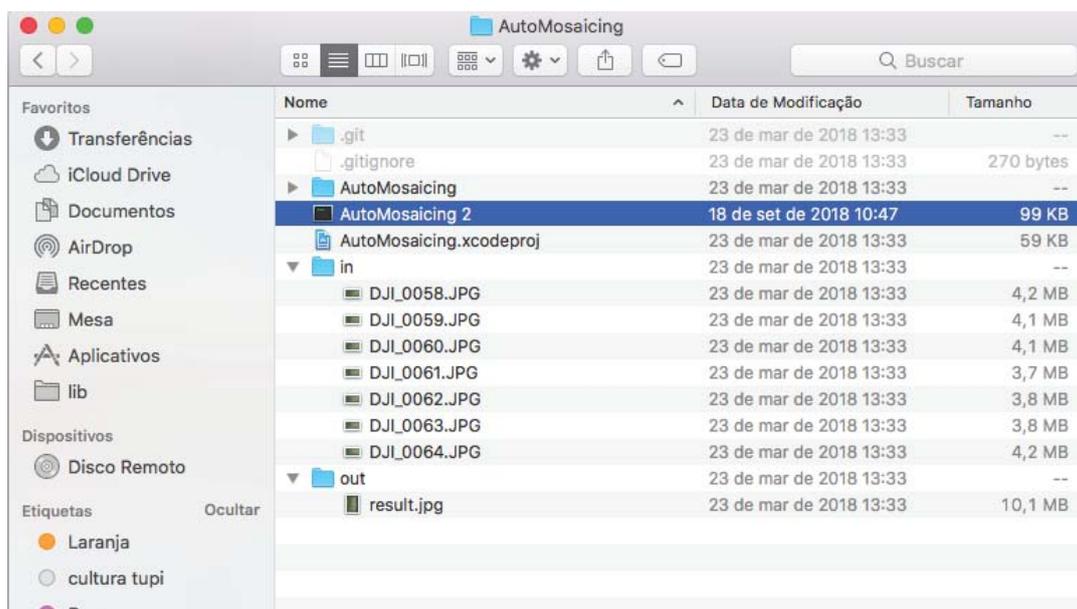
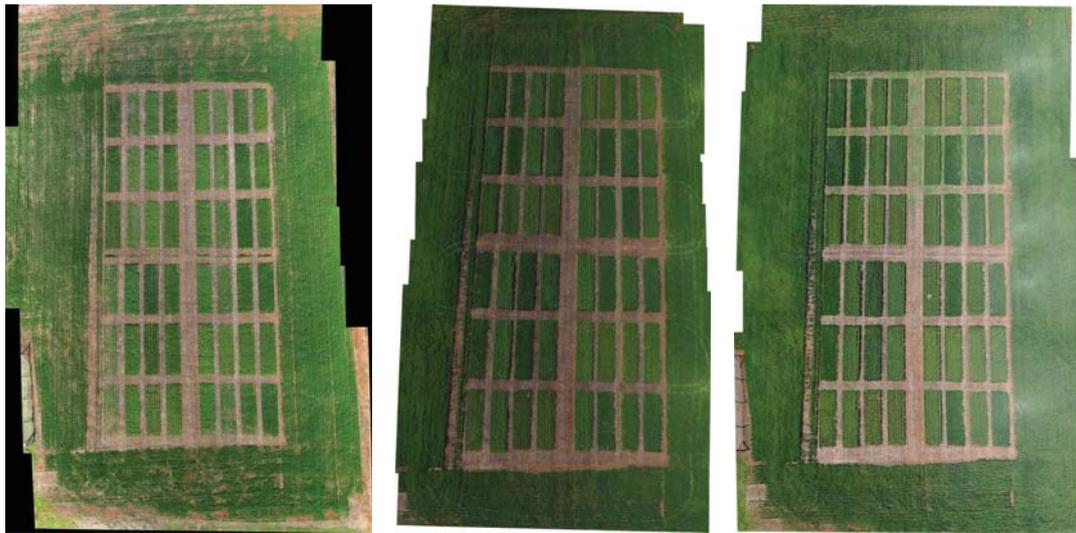


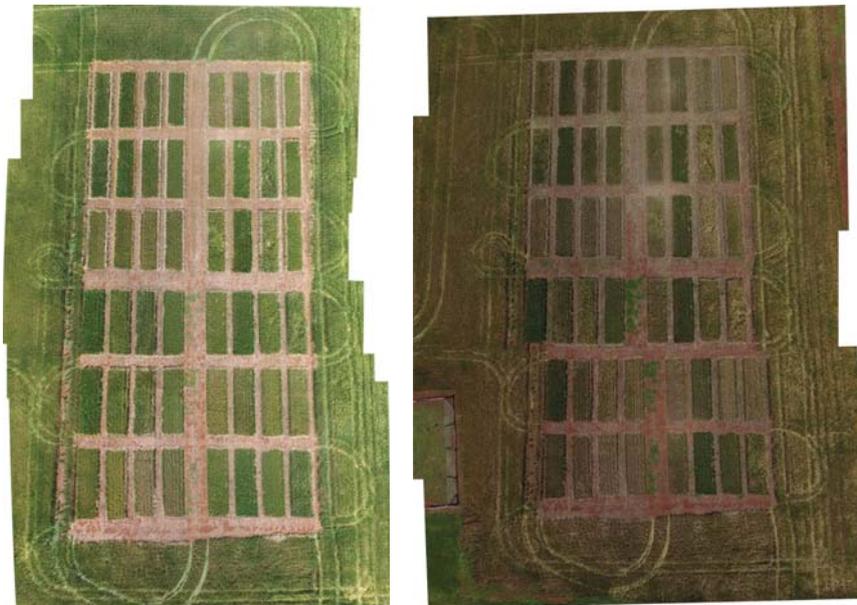
Figura 44. Diretório do Software com o executável e as pastas de entrada e saída de imagens.

4.5 TESTES E VALIDAÇÕES

O aplicativo desenvolvido foi utilizado para fazer o mosaico das imagens anteriormente coletadas com o aplicativo Auto Capture na área da Embrapa Trigo - Passo Fundo já citada anteriormente. Foram feitas um total de 5 coletas em diferentes datas, utilizando dois tipos de câmeras diferentes, uma RGB e outra NDVI. Na Figura 45 pode-se observar os Mosaicos gerados a partir das imagens da câmera RGB e na Figura 46 os mosaicos NDVI.



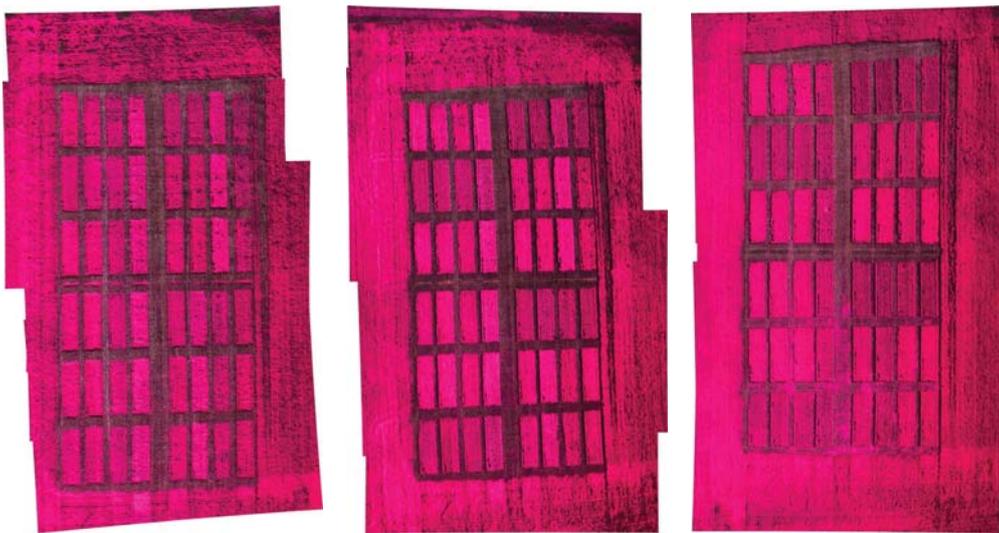
(a) Mosaico RGB proveniente da coleta do dia 24/08/2017. (b) Mosaico RGB proveniente da coleta do dia 06/09/2017. (c) Mosaico RGB proveniente da coleta do dia 21/09/2017.



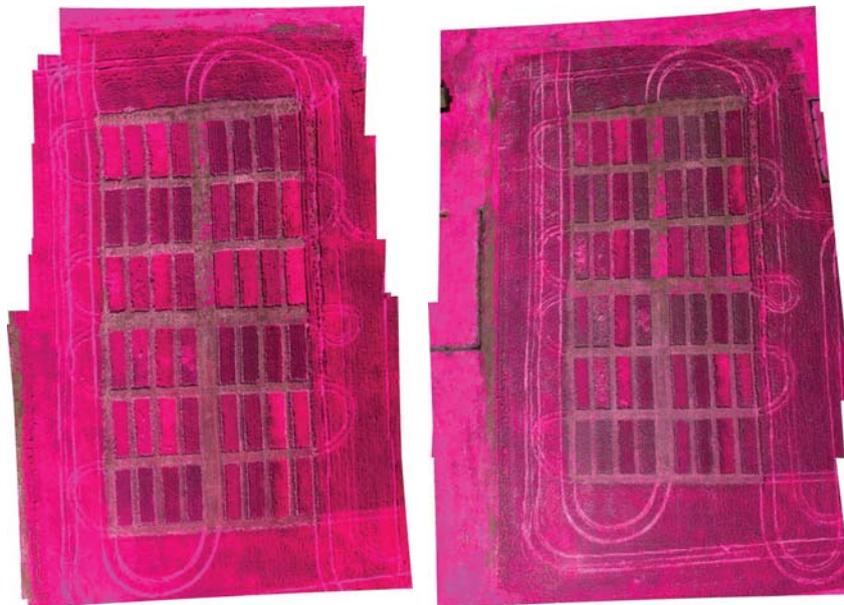
(d) Mosaico RGB proveniente da coleta do dia 17/10/2017. (e) Mosaico RGB proveniente da coleta do dia 01/11/2017.

Figura 45. Mosaicos RGB gerados pelo Software desenvolvido. Pode-se observar uma alteração na cor dos cultivares ao decorrer dos dias.

A construção dos mosaicos RGB obtiveram um bom resultado final comparado as primeiras tentativas de mosaicagem sem a utilização da técnica de ROIs. Foram utilizadas de 6 a 10 imagens para a construção de cada mosaico. pode-se observar na Figura 45(e) que a área total do mosaico é maior que os demais mosaicos, devido a quantidade de imagens de entrada fornecidas.



(a) Mosaico NDVI proveniente da coleta do dia 24/08/2017. (b) Mosaico NDVI proveniente da coleta do dia 06/09/2017. (c) Mosaico NDVI proveniente da coleta do dia 21/09/2017.



(d) Mosaico NDVI proveniente da coleta do dia 17/10/2017. (e) Mosaico NDVI proveniente da coleta do dia 01/11/2017.

Figura 46. Mosaicos NDVI gerados pelo software. Estas imagem ainda não foram processadas por um software que faz o cálculo de NDVI.

4.6 CONCLUSÕES

A utilização da técnica de pontos de interesse (ROI) tornou-se um fator importante para este trabalho, visto que o algoritmo anteriormente testado não obteve sucesso na mosaicagem justamente por não ter um ponto de partida para a etapa de registro da mosaicagem. Para este trabalho, seus resultados podem ser considerados satisfatórios, pois as imagens utilizadas aqui foram posteriormente analisadas e processadas, gerando resultados condizentes com a utilização de outros aplicativos e softwares proprietários.

Todos os mosaicos foram criados utilizando o mesmo cenário, mas em datas diferentes, o que não garante seu pleno funcionamento em ambientes grandes, visto que o número de imagens mosaicadas não passou de 12. O algoritmo pode funcionar utilizando mais imagens, desde que seja utilizado o aplicativo Auto Capture.

Como trabalhos futuros, destaca-se a necessidade de desenvolvimento de uma interface gráfica, o que poderá ajudar pessoas leigas no assunto a utilizarem a plataforma, além de um aprimoramento na função que faz o cálculo da área de interesse (ROI).

5. RELAÇÃO ENTRE BIOMASSA E NDVI EM DIFERENTES CULTIVARES DE TRIGO E DOSES DE NITROGÊNIO

5.1 RESUMO

A utilização da biomassa em seu pico de produtividade pode fornecer informações importantes sobre o desenvolvimento da planta. A cultura do trigo se encaixa a este fator, e utilizando o índice NDVI, este que está intimamente correlacionado com a biomassa, pode-se estimar qual a correta dosagem de nitrogênio deve ser aplicada no cultivar. Neste trabalho, utilizando dados de um experimento de responsabilidade da Embrapa Trigo no município de Coxilha, RS, coletados utilizando a plataforma desenvolvida nos capítulos anteriores, dados NDVI foram gerados e estes correlacionados com a Matéria Seca (MS) coletados em solo. A análise mostrou que a variável NDVI obtida pelos sensores apresentou relação direta com a produtividade para os estádio fenológicos avaliados.

5.2 INTRODUÇÃO

Estimativas de biomassa de certas culturas no pico de sua produtividade podem fornecer informações valiosas do seu estado nutricional da planta. Um exemplo disto é a cultura do trigo, onde variáveis como matéria seca (MS) e quantidade de clorofila na folha são componentes importantes, que quando mensurados, podem ser utilizadas para a estimativa do potencial produtivo da cultura, que por sua vez, pode ser utilizada na quantificação da dose de nitrogênio N a ser aplicada.

O nitrogênio (N) é um dos elementos mais importantes para o desenvolvimento das plantas e também o mais preocupante ao meio ambiente. A recomendação da quantidade de fertilizante nitrogenado a se aplicar na cultura do trigo varia em função do nível de matéria orgânica do solo, da cultura precedente e da expectativa de rendimento de grão da cultura, a qual depende da interação de vários fatores de produção e das condições meteorológicas [61]. Essas variáveis são pouco eficientes já que variam com o clima de cada ano, região e também não consideram as variações espaciais de cada área agrícola. Assim, as necessidades reais da planta muitas vezes não são supridas ou quantidades significativas de fertilizantes nitrogenados são lixiviadas no perfil do solo [62]. Para tanto, há a necessidade de técnicas eficientes na determinação da dose de N a ser aplicada.

Diversos trabalhos vem sendo realizados para obter doses mais precisas de N, de acordo com necessidades específicas de cada cultivar e local. As ferramentas de agricultura de precisão consideram a variabilidade espacial, onde técnicas de sensoriamento remoto capturam e registram a energia refletida pelo alvo, no caso, as plantas. O sol, como fonte de energia eletromagnética, incide sobre a superfície terrestre, onde parte dessa energia é refletida. A vegetação apresenta picos de absorção da radiação incidente decorrentes dos pigmentos fotossintéticos, em especial das clorofilas [63]. Logo, dados da resposta espectral do dossel da cultura, caracterizados pelo processo de reflectância de ondas eletromagnéticas incidentes, podem ser utilizados como indicadores indiretos do

status nutricional da cultura, bem como da biomassa e conseqüentemente do potencial produtivo da cultura.

Trabalhos recentes apresentam a possibilidade da utilização de sensores para coletar essa energia eletromagnética e transformá-la em índices de vegetação, como o Normalized Difference Vegetation Index (NDVI), afim de estimar, entre outros a biomassa, nitrogênio e carbono[64]. Segundo Blaschke [65], estes índices fazem parte dos modernos dados provenientes do sensoriamento remoto e são obtidos, tipicamente por meio de satélites, aviões e mais recentemente por Drones.

Seguindo nesta área, a reflexão espectral da faixa visível, bem como a do infravermelho próximo, podem ser combinadas para produzir índices, que são sensíveis aos aspectos biofísicos, bem como variações químicas na vegetação. Estes índices estão ligados a qualidade das imagens coletadas no campo, a utilização de equipamentos adequados, como sensores específicos podem expressar resultados muito próximos do real [66].

Com o intuito de poder validar a plataforma que foi desenvolvida neste trabalho, este capítulo abordará um estudo prático de coleta e obtenção dos valores NDVI de cultivares de trigo, análises estatísticas, bem como um cruzamento de dados com os coletados em solo.

5.3 MATERIAIS E MÉTODOS

Para este trabalho, foi utilizado o campo experimental cedido pela Embrapa Trigo - Passo fundo - RS, localizado no município de Coxilha, Rio Grande do Sul (Lat: 28.186418 Long: -52.325192). O desenho experimental consistiu em parcelas dispostas em faixas com quatro repetições para cada cultura e três diferentes doses de Nitrogênio (kg ha⁻¹). Na Figura 47 é apresentado um croqui do experimento, onde a faixa horizontal representa as cultivares de trigo e na vertical as doses de Nitrogênio aplicado em cobertura, sendo as cultivares de trigo Tarumã, Toruk, Guamirim e Parrudo escolhidas para este experimento.

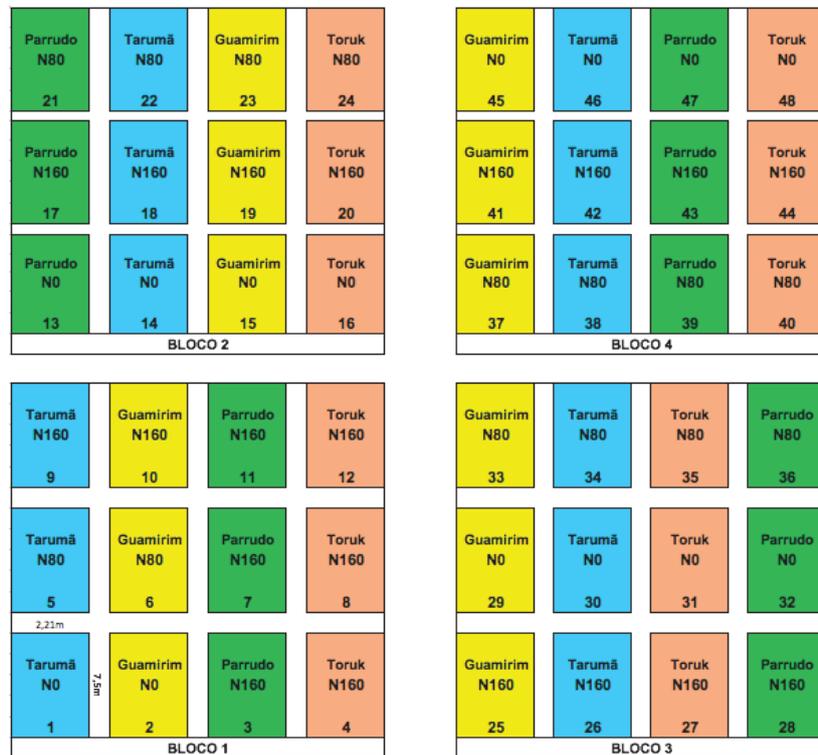


Figura 47. Croqui do experimento onde estão dispostos a semeadura da cultivar e a quantidade de nitrogênio que é aplicada. Existem quatro cultivares, quatro repetições e três doses de nitrogênio aplicado a eles.

A semeadura do trigo foi realizada no dia 29/06/2017, sendo utilizada como adubação base 300 kg/ha de NPK 05-25-25, com uma densidade de 80 sementes viáveis por metro. O desenho experimental foi de parcelas subdivididas onde a parcela principal foi a cultivar e as doses de Nitrogênio as sub-parcelas. Os tratamentos com nitrogênio seguiram um padrão com dosagem de N0 = 0 kg de N/ha em cobertura, N80 = 80 kg de N/ha em cobertura e N160 = 160 kg de N/ha em cobertura. Como é apresentada na Figura 47, foram numeradas as parcelas de 1 a 48, onde nelas apresenta-se a cultivar de trigo semeada, qual o tratamento a ser feito (N) e seu número de parcela. Cada um dos quatro blocos representa uma repetição das culturas, sendo que a área de cada parcela possui 16,57m² (7,5m x 2,21m).

A reflectância do experimento foi avaliada pelo sensor TETRACAM acoplado ao Drone (Figura 48(b)). Esse sensor consegue capturar imagens com comprimento de onda de 250nm da faixa visível a 920nm da faixa de infravermelho próximo, podendo fornecer o (NDVI). Os voos foram realizados a 30 metros de altura, entre as 11h30 e 13h da tarde em quatro diferentes estádios da cultura, sendo o primeiro no dia 10 de agosto de 2017, no final do afilamento. O segundo, no dia 24 de agosto de 2017, início do alongamento. O terceiro, no dia 06 de setembro de 2017, no final do alongamento. O quarto voo, no dia 21 de setembro de 2017, no florescimento, além de um último voo já no final de seu ciclo no dia 17 de outubro 2017. Com essas imagens, realizou-se georreferenciamento da parcela, registro, mosaico e recorte da área de estudo, utilizando o software Auto Mosaicing desenvolvido como parte deste trabalho. Na sequência, aplicou-se os processos de conversão dos valores digitais (ND) para fatores de reflectância e normalização radiométrica, no software PixelWrench2 desenvolvido pela

fabricante da câmera TETRACAM, gerando índices de vegetação para análises da cultura em estudo. Com isso Os dados foram submetidos a análise de correlação entre os valores de NDVI e biomassa para diferentes doses de N para cada cultivar estudada.



(a) Drone com a câmera TETRACAM acoplada utilizado para realizar as coletas.

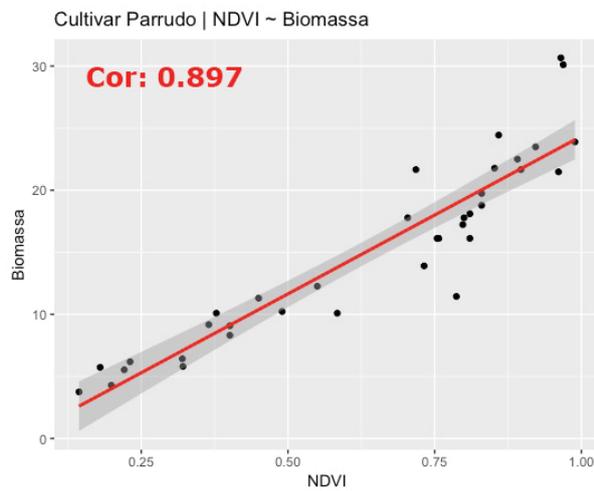


(b) Câmera Multispectral Micro ADC [43]

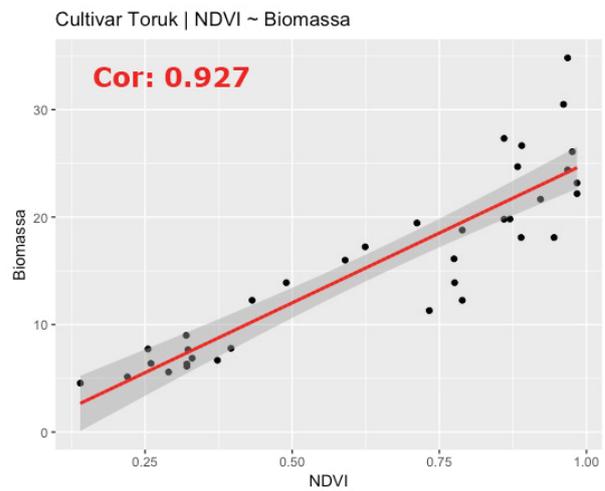
Figura 48. Equipamentos utilizados nas coletas, (48(a)) Drone utilizado, (48(b)) Câmera Multispectral Micro ADC [43].

5.4 RESULTADOS

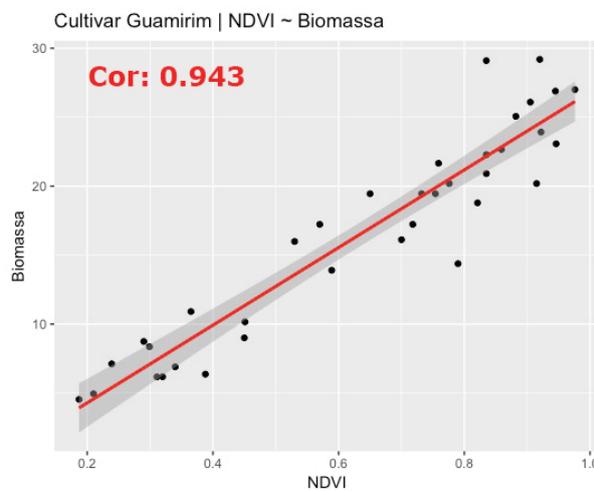
Os valores de NDVI foram correlacionados com os valores de biomassa, sendo que seu comportamento do NDVI entre as cultivares variou em função das doses de N e estágio fenológico, sendo que a cultivar Guamirim apresentou o maior coeficiente de correlação. Esse fato deve-se principalmente pelo fato de que durante o período de coletas, esta é uma das culturas com ciclo mais precoce. Na figura 49 pode-se observar os gráficos gerados.



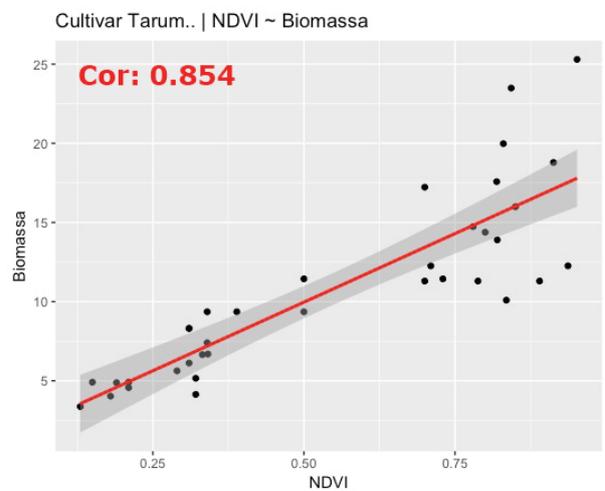
(a) Gráfico de correlação da cultivar Parrudo.



(b) Gráfico de correlação da cultivar Toruk.



(c) Gráfico de correlação da cultivar Guamirim, sendo este o maior coeficiente de correlação obtido.



(d) Gráfico de correlação da cultivar Tarumã.

Figura 49. Gráficos de correlação entre os valores de NDVI e Biomassa das culturas Parrudo, Toruk, Guamirim e Tarrumã. Foram utilizados os dados coletados nas 3 primeiras coletas, 10/08/2017, 24/08/2017, 06/09/2017.

Hatfiel [67], em estudo sobre a variação espectral de oitenta e dois genótipos de trigo, entre a fase de vegetativa e ponto de colheita, observou variabilidade na resposta espectral dos genótipos. O autor concluiu que as diferenças encontradas estão associadas não apenas às variáveis agrônômicas como índice de área foliar ou acúmulo de biomassa, mas também altamente dependente da geometria de incidência da radiação solar e sua interação com as características da arquitetura dos dosséis vegetativos.

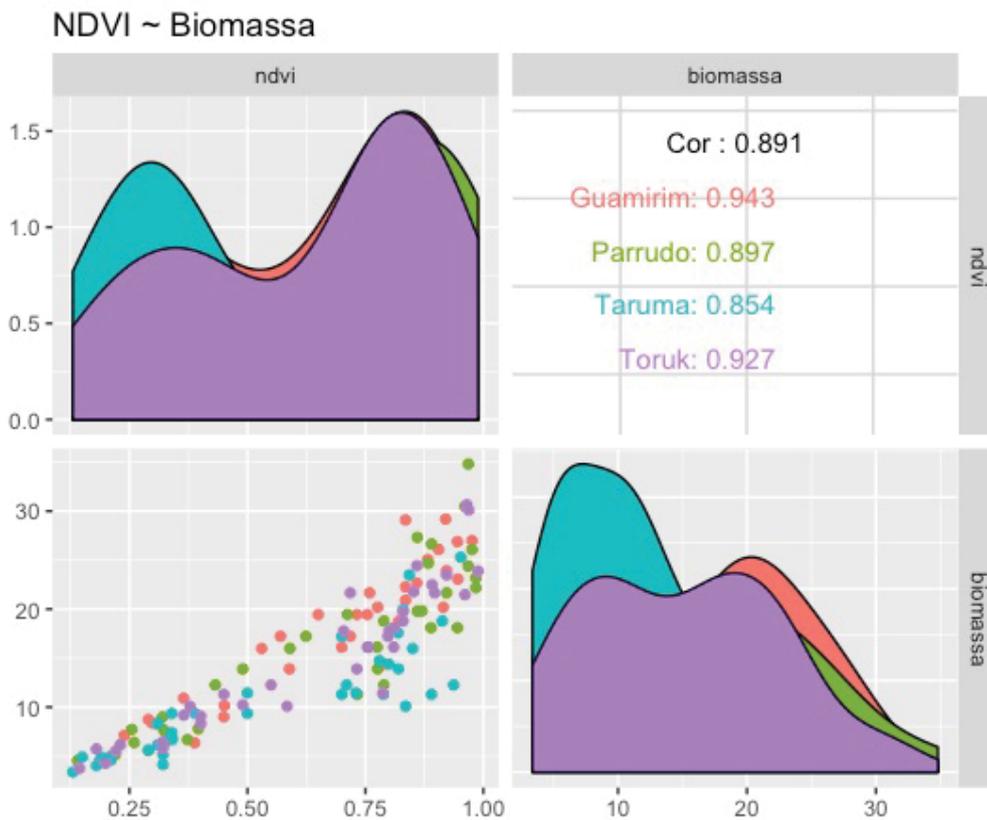


Figura 50. A média de correlação das quatro cultivares foi de 0,898. Neste gráfico pode-se observar melhor como foi o comportamento de cada cultivar em relação outra.

Vian [48] trabalhando com NDVI e biomassa em trigo, também encontrou distintos valores de NDVI para cada cultivar analisada devido aos diferentes incrementos de biomassa seca da parte aérea. Conforme Moges [34], a biomassa acumulada na parte aérea é uma das variáveis que melhor se correlaciona com o valor de NDVI. Ainda, de acordo com Liu [68], o acúmulo de biomassa na parte aérea em trigo se relaciona diretamente com a produtividade final de grãos, que, por sua vez, também se relaciona com as doses de N aplicadas.

A correlação entre os valores de NDVI e as diferentes doses de N é apresentada na Figura 51, para cada cultivar estudada. De tal maneira, as correlações apresentaram valores acima de 0,99. Esses resultados corroboram com os de Povh [69], Rissini [40], Bredemier & Mundstock [48] que também encontraram valores de r entre 0,80 e 0,99 entre NDVI e biomassa.

Essa forte correlação deve-se, principalmente, ao fato que o aumento de reflectância na região do infravermelho próximo, em função de maior acúmulo de biomassa, e maior absorbância na região do vermelho, em função de maior quantidade de clorofila na folha, são proporcionadas pela maior disponibilidade de N promovida por doses crescentes aplicadas na base.

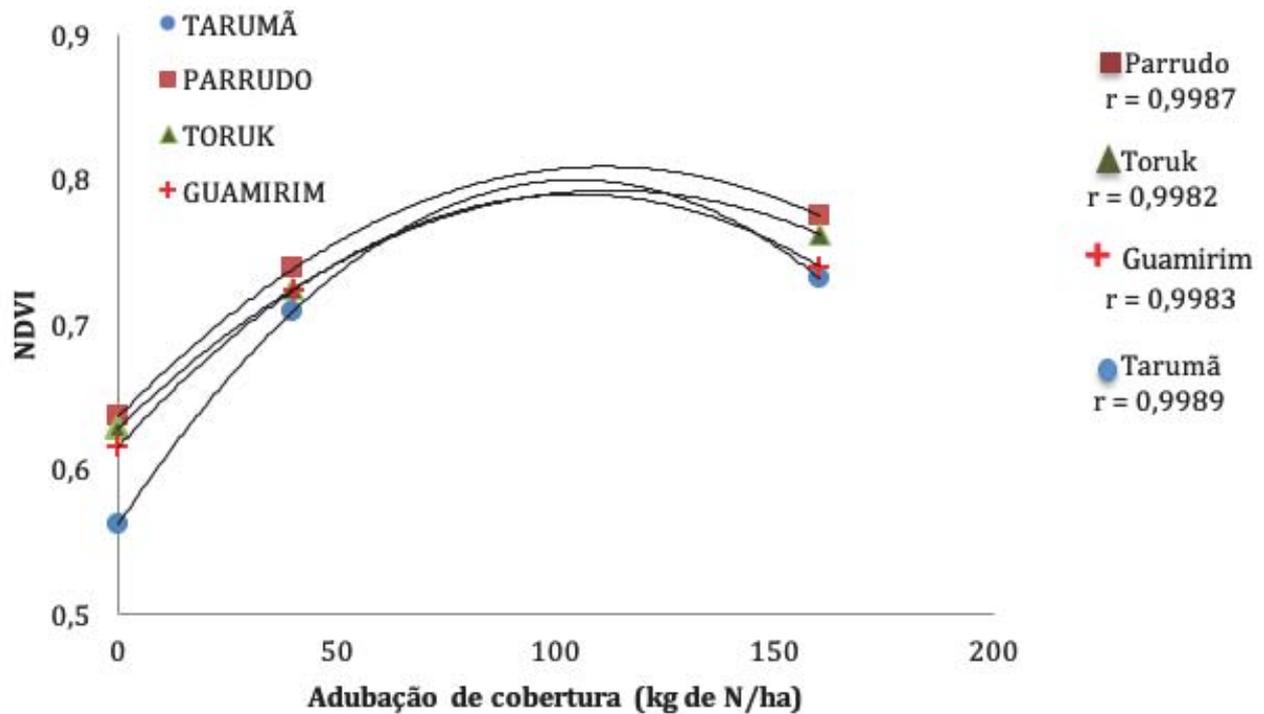


Figura 51. Gráfico de correlação entre diferentes doses de N e NDVI das cultivares Parudo, Toruk, Guamirim e Tarumã

5.5 CONCLUSÕES

As leituras do índice de vegetação por diferença normalizada (NDVI) ao longo do ciclo das diferentes cultivares de trigo, mostraram-se eficientes em identificar variações de biomassa promovidas por variações na disponibilidade de nitrogênio.

A utilização de Drones e câmeras multiespectrais na coleta de imagens trazem um grande benefício temporal, ao contrário de satélites, onde as imagens são disponibilizadas em horários diferentes ao necessário para obter-se um bom resultado. Aliado a este fato, a utilização de uma plataforma que automatize alguns processos neste trabalho tornaram-se importantes para ganhar mais agilidade na obtenção destes dados, além de simplificar este processo que pode vir a ser complexo para pessoas com pouco contato com os equipamentos.

6. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

O aplicativo desenvolvido no Capítulo 3 pode receber algumas melhorias futuramente. O primeiro ponto a ser discutido como trabalho futuro é a utilização da API do GoogleMaps no lugar do MapKit da Apple. Isso deve-se principalmente pelas limitações encontradas na utilização do mesmo para trabalhar com marcadores no mapa, sendo que o procedimento para movimentar automaticamente pelo mapa foi feito de forma manual, ao contrário do GoogleMaps, aonde já existem funcionalidades para tal. Outro ponto a ser abordado é a utilização de mapas offline, este disponibilizado pela API do GoogleMaps torna-se importante para a utilização do aplicativo em áreas aonde não há nenhuma conexão com internet para acessar o mapa na tela.

O aplicativo atualmente é compatível com dispositivos da marca DJI que suportam a utilização do seu SDK. A partir da versão 4.0 do SDK houve uma separação de funcionalidades da SDK para diferentes dispositivos, sendo uma das principais a funcionalidade de disparo automático da câmera no modo de voo autônomo. Para dispositivos como Phantom 3 series, usado para este trabalho, o disparo da câmera é realizado em função de um *timer* definido que considera a velocidade de voo e sua altura. Este cálculo deve ser implementado pelo programador. Na versão Phantom 4 ou superior, Matrice 100, Matrice 600, Mavic Pro ou superior, o disparo automático da câmera pode ser definido pela distância percorrida, o que para utilização no mapeamento pode tornar-se mais preciso em relação ao relativismo do *timer* definido por um calculo de altura e velocidade.

Outro ponto a ser discutido é a utilização de uma banco de dados como o Google Firebase para armazenamento de rotas criadas pelo usuário, o que pode ser útil para mapeamento constante de uma área. Este por sua vez poderia poupar ao usuário a criação da mesma rota todas as vezes, além de trazer uma maior precisão e igualdade nos mapeamentos de uma mesma área em diferentes datas. Junto com esta funcionalidade também deverá ser implementado um sistema de usuários e login no aplicativo, o que para tal também poderá ser utilizado a funcionalidade Auth do Firebase.

Por fim, ainda é interessante desenvolver uma versão do aplicativo para a plataforma Android, visto que a SDK DJI também esta disponível para tal. Desta maneira o público a ser beneficiado com este aplicativo pode-se tornar ainda maior.

O software desenvolvido no Capítulo 4 deste trabalho funciona de maneira bem simplificada, executando em modo terminal. Como trabalho futuro, o desenvolvimento de uma interface gráfica pode deixá-lo mais simples para usuários sem contato com o terminal dos sistemas operacionais, visto que a utilização do mesmo não é comum para todos. Além da interface gráfica um último passo antes de exportar o mosaico gerado pode ser implementado. Este ultimo passo, que consiste em desenvolver um visualizador do mosaico gerado, onde o usuário poderá visualizar e fazer recortes no mosaico ou salvá-lo com diferentes formatos.

7. CONCLUSÕES

A utilização de softwares e aplicativos que de alguma forma possibilitem a redução de passos a serem executados em uma coleta de imagens mostrou-se eficaz para este trabalho. De uma forma geral, sem a utilização da plataforma, o trabalho e o tempo para mapear alguma área de interesse seria maior

Com as coletas realizadas na área da EMBRAPA TRIGO, o aplicativo mostrou-se capaz de capturar imagens de forma automática e simples, facilitando o trabalho de mapeamento do meio agrícola. Na parte funcional, onde se encaixa a exatidão no ponto de coleta de cada imagem, foram obtidos resultados satisfatórios, sendo que cada imagem teve uma média próxima a desejada de 70% de *overlapping* com sua vizinha, fator importante a ser considerado para a posterior mosaicação das imagens.

O software de mosaicação desenvolvido no capítulo 4 mostrou-se capaz de realizar o processamento das imagens anteriormente obtidas com o aplicativo Auto Capture. A utilização da técnica de ROI (do português, "retângulo de interesse") aprimorou o algoritmo já existente utilizado para a criação de mosaicos. Apesar de não ter interface gráfica o aplicativo é simples de ser utilizado, bastando seguir algumas etapas para obter-se o resultado esperado.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] RONG, W. et al. Mosaicing of microscope images based on surf. In: *2009 24th International Conference Image and Vision Computing New Zealand*. [S.l.: s.n.], 2009. p. 271–275. ISSN 2151-2191.
- [2] KOLONIA, P. When more is better. *Popular Photography*, v. 58, n. 1, p. 30–34, 1994.
- [3] LOCH, C. Importância do monitoramento global e integrado do planejamento municipal. In: *VI CONGRESSO BRASILEIRO DE SENSORIAMENTO REMOTO*. [S.l.: s.n.], 2001. v. 2, p. 523–530.
- [4] FERREIRA, A. M. R. et al. Utilização de aeronaves remotamente pilotadas para extração de mosaico georreferenciado multiespectral e modelo digital de elevação de altíssima resolução espacial. *Anais do Simpósio Brasileiro de Sensoriamento Remoto (SBSR)*, 2013.
- [5] ARTIOLI, F.; BELONI, T. Diagnóstico do perfil do usuário de drones no agronegócio brasileiro &61; profile diagnosis of users of drones in brazilian agribusiness. *Revista iPecege*, Directory of Open Access Journals, v. 2, n. 3, p. 40–56, 2016.
- [6] MANERA, J. F. et al. Aerial image acquisition and processing for remote sensing. *Journal of Computer Science & Technology*, v. 10, 2010.
- [7] ARTIOLI, F.; BELONI, T. Diagnóstico do perfil do usuário de Drones no Agronegócio Brasileiro. *Revista iPecege*, v. 2, n. 3, p. 40–56, 2016. ISSN 2359-5078. Disponível em: <<https://revista.ipecege.org.br/Revista/article/view/73>>.
- [8] ROSSETTI, L. Modelagem dinâmica espacial de mudanças no uso do solo urbano: contribuição metodológica. p. 132, 2011. Disponível em: <<http://base.repositorio.unesp.br/handle/11449/104295>>.
- [9] WOLF, P. R.; DEWITT, B. A. *Elements of Photogrammetry: with applications in GIS*. [S.l.]: McGraw-Hill New York, 2000. v. 3.
- [10] MUSEUM, M. *Boston, as the Eagle and the Wild Goose See It*. Disponível em: <<https://www.metmuseum.org/art/collection/search/283189>>. Acesso em: Out. 16, 2018.
- [11] UB, B. U. *Universidade de Barcelona*. Disponível em: <<http://www.ub.edu/geocrit/b3w-828.htm>>. Acesso em: Out. 16, 2018.
- [12] PHOTOSCAN, A. *Photogrammetric processing of digital images and generates 3D spatial data*. Disponível em: <<http://www.agisoft.com/>>. Acesso em: Out. 16, 2018.
- [13] MORÉS, N. et al. Metodologia para a Geração de Mosaicos Semicontrolados a partir de Imagens Orbitais Provenientes do Google Earth. p. 1–5, 2002.
- [14] PIX4D. *PIX4D Software*. Disponível em: <<https://pix4d.com/>>. Acesso em: Out. 16, 2018.

- [15] ICE, M. *Microsoft Image Composer Editor*. Disponível em: <<http://research.microsoft.com/en-us/um/redmond/projects/ice/>>. Acesso em: Out. 16, 2018.
- [16] ALEXANDRE, C.; OLIVEIRA, A. D. Mosaicagem de imagens para aplicações industriais.
- [17] CERBARO, V. A. et al. Plataforma de baixo custo para coleta de imagens ndvi. In: X CONGRESSO BRASILEIRO DE AGROINFORMÁTICA (SBIAGRO 2015). [S.l.]: UEPG.
- [18] BURA RAZVAN NESIU, A. K. J. *The Ultimate iOS 10, Xcode 8 Developer Book. Build 30 apps*. [S.l.]: Lulu.com, 2017. ISBN 978-1365711459.
- [19] REBOUÇAS, M. et al. An empirical study on the usage of the swift programming language. In: *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*. [S.l.: s.n.], 2016. v. 1, p. 634–638.
- [20] COCOAPODS. *Dependency management*. Disponível em: <<https://cocoapods.org/>>. Acesso em: Out. 16, 2018.
- [21] DJI, D. *System Developmet Kit DJI*. Disponível em: <<http://dev.dji.com/>>. Acesso em: Out. 16, 2018.
- [22] DRONE, D. 2014. (Number of Drones). Disponível em: <<http://dronecentral.com.br/>>. Acesso em: 30 Jul. 2016.
- [23] JOSE, B. et al. Multispectral ndvi aerial image system for vegetation analysis by using a consumer camera. In: IEEE. *Power, Electronics and Computing (ROPEC), 2014 IEEE International Autumn Meeting on*. [S.l.], 2014. p. 1–6.
- [24] GAZCÓN, N. F.; CHESÑEVAR, C. I.; CASTRO, S. M. Automatic vehicle identification for argentinean license plates using intelligent template matching. *Pattern Recognition Letters*, Elsevier, v. 33, n. 9, p. 1066–1074, 2012.
- [25] OPENCV. *Open Source Computer Vision*. Disponível em: <<http://opencv.org/>>. Acesso em: 29 Jul. 2018.
- [26] LEEMANS, V.; DUMONT, B.; DESTAIN, M.-F. Assessment of plant leaf area measurement by using stereo-vision. In: IEEE. *3D Imaging (IC3D), 2013 International Conference on*. [S.l.], 2013. p. 1–5.
- [27] HUANG, C.-M.; LIN, S.-W.; CHEN, J.-H. Efficient image stitching of continuous image sequence with image and seam selections. *Sensors Journal*, IEEE, v. 15, n. 10, p. 5910–5918, 2015.
- [28] KAILI, C.; MEILING, W. Image stitching algorithm research based on opencv. In: IEEE. *Control Conference (CCC), 2014 33rd Chinese*. [S.l.], 2014. p. 7292–7297.
- [29] TRUCCO, E.; VERRI, A. Motion. In: _____. *Introductory Techniques for 3-D Computer Vision*. [S.l.: s.n.], 1998. p. 178–194. ISBN 9780132611084.

- [30] JULIEN, Y. et al. Temporal analysis of normalized difference vegetation index (NDVI) and land surface temperature (LST) parameters to detect changes in the Iberian land cover between 1981 and 2001. Disponível em: <https://www.uv.es/ucg/articulos/2011/Publications_2011_3.pdf>.
- [31] ROYO C.; S. VILLEGAS, D. Field measurements of canopy spectra for biomass assessment of small-grain cereals. v. 3, 2011.
- [32] BACKES, K. S. Variações do índice de vegetação por diferença normalizada (ndvi) do sensor modis associadas a variáveis climáticas para o estado do rio grande do sul. 2010.
- [33] PONZONI F. J.; SHIMABUKURO, Y. E. Sensoriamento remoto no estudo da vegetação. 2007.
- [34] MOGES, S. M. e. a. Evaluation of green, red, and near infrared bands for predicting winter wheat biomass, nitrogen uptake, and final grain yield. 2015.
- [35] PERRY, E. M. e. a. Ndvi from active optical sensors as a measure of canopy cover and biomass. isprs-international archives of the photogrammetry, remote sensing and spatial information sciences. 2012.
- [36] SUGAWARA, L. M. Avaliação de modelo agrometeorológico e imagens noaa/avhrr no acompanhamento e estimativa de produtividade da soja no estado do paran . 2002.
- [37] FU, Y. e. a. Winter wheat biomass estimation based on spectral indices, band depth analysis and partial least squares regression using hyperspectral measurements. 2014.
- [38] PRIMICERIO, J. e. a. A flexible unmanned aerial vehicle for precision agriculture. precision agriculture. 2012.
- [39] TURNER D. LUCIEER, A. W. C. Development of an unmanned aerial vehicle (uav) for hyper resolution vineyard mapping based on visible, multispectral, and thermal imagery. 2011.
- [40] RISSINI, A. L. Ndvi, crescimento e produtividade de cultivares de trigo submetidas a doses de nitrog nio. 2011.
- [41] DWORAK, V. e. a. Strategy for the development of a smart ndvi camera system for outdoor plant detection and agricultural embedded systems. 2013.
- [42] ARAUS, J. L. et al. New Technologies for Phenotyping. In: *Phenomics*. Cham: Springer International Publishing, 2015. p. 1–14. Disponível em: <http://link.springer.com/10.1007/978-3-319-13677-6_1>.
- [43] TETRACAM. *Better Science Through Advanced Imaging*. Disponível em: <<http://www.tetracam.com/>>. Acesso em: Out. 16, 2018.
- [44] CANDIAGO, S.; REMONDINO, F. Evaluating Multispectral Images and Vegetation Indices for Precision Farming Applications from UAV Images. In: _____. [S.l.: s.n.]. v. 3, p. 4026–4047.

- [45] MICASENSE. *At MicaSense, we believe in delivering more than pretty pictures*. Disponível em: <<https://www.micasense.com/rededge-m>>. Acesso em: Out. 16, 2018.
- [46] SALGADO, M. P. G.; FORMAGGIO, A. R.; RUDORFF, B. F. T. Anais XV Simpósio Brasileiro de Sensoriamento Remoto - SBSR, Curitiba, PR, Brasil, 30 de abril a 05 de maio de 2011, INPE p.8880. *XV Simpósio Brasileiro de Sensoriamento Remoto - SBSR*, n. 2000, p. 8880–8887, 2011. ISSN 00223077.
- [47] ABRAHÃO, S. A. et al. Índices De Vegetação De Base Espectral Para Discriminar Doses De Nitrogênio Em Capim-Tanzânia. *Revista Brasileira de Zootecnia*, v. 38, n. 9, p. 1637–1644, 2009. ISSN 15163598.
- [48] BREDEMEIER, C.; VIAN, A. Classes de ndvi como ferramenta para recomendação de nitrogênio em milho. 2016.
- [49] CARVALHO, M.; NETO, M. Desenvolvendo Aplicações Multimídia e Ubíquas com Drones. n. November, 2016.
- [50] APPLE. *Start Development iOS Apps (Swift)*. Disponível em: <<https://developer.apple.com/library/content/navigation/>>. Acesso em: Out. 16, 2018.
- [51] SWIFT. *The powerful programming language that is also easy to learn*. Disponível em: <<http://developer.apple.com/swift/>>. Acesso em: Out. 16, 2018.
- [52] DJI. *DJI Developer*. Disponível em: <<https://dev.dji.com/>>. Acesso em: Out. 16, 2018.
- [53] FERREIRA, A. et al. Utilização de aeronaves remotamente pilotadas para extração de mosaico georreferenciado multiespectral e modelo digital de elevação de altíssima resolução espacial. *Simpósio Brasileiro de Sensoriamento Remoto*, v. 1, p. 9308–9315, 2013.
- [54] DJI. *The Powerfull Drones*. Disponível em: <<http://dji.com/>>. Acesso em: Out. 16, 2018.
- [55] TARALLO, A. D. S. et al. Construção automática de mosaico de imagens agrícolas aéreas sequenciais. *Agricultura de precisão: um novo olhar*, p. 106–109, 2011.
- [56] YANG, L. et al. A research of feature-based image mosaic algorithm. In: *2011 4th International Congress on Image and Signal Processing*. [S.l.: s.n.], 2011. v. 2, p. 846–849.
- [57] GHOSH, D. Quantitative evaluation of image mosaicing in multiple scene categories. *IEEE International Conference on Electro/Information Technology*, p. 1–6, 2012.
- [58] GHOSH KAABOUCHE, F. Robust spatial-domain based super-resolution mosaicing of cubesat video frames: Algorithm and evaluation. p. 68, 2014.
- [59] OPENCV. *Documentation Open Source Computer Vision*. Disponível em: <<https://docs.opencv.org/2.4/modules/stitching/doc/introduction.html>>. Acesso em: 29 Jul. 2018.

- [60] BROWN, M.; LOWE, D. Automatic Panoramic Stitching Using Invariant Features. *International Journal on Computer Vision (IJCV)*, v. 74, n. 1, p. 59–73, 2007. ISSN 09205691. Disponível em: <<http://link.springer.com/article/10.1007/s11263-006-0002-3>>.
- [61] SILVA S, B. M. Informações técnicas para trigo e triticales - Safra 2017. Embrapa Soja. *REUNIÃO DA COMISSÃO BRASILEIRA DE PESQUISA TRIGO E TRITICALE*, 2017.
- [62] ROSOLEM, C. A.; FOLONI, J. S. S.; De Oliveira, R. H. Dinâmica do nitrogênio no solo em razão da calagem e adubação nitrogenada, com palha na superfície. *Pesquisa Agropecuária Brasileira*, v. 38, n. 2, p. 301–309, 2003. ISSN 0100204X.
- [63] TURRA, M. A. *UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL FACULDADE DE AGRONOMIA PROGRAMA DE PÓS-GRADUAÇÃO EM FITOTECNIA MANEJO DO NITROGÊNIO EM CULTIVARES DE TRIGO BASEADO NO ÍNDICE DE VEGETAÇÃO POR DIFERENÇA NORMALIZADA (NDVI)*. [S.l.]. Disponível em: <<https://lume.ufrgs.br/bitstream/handle/10183/172090/001058376.pdf?sequence=1&isAllowed=y>>.
- [64] COLTRI, P. et al. UtilizaÇÃo de Índices de vegetaÇÃo para estimativas nÃo- destrutivas da biomassa, estoque e sequestro de carbono do cafeeiro arÁbica. 01 2009.
- [65] BLASCHKE, T. *Sensoriamento remoto e SIG avançados*. [S.l.]: Brochura, 2007. ISBN: 978-85-86238-57-4.
- [66] MUTANGA, O.; SKIDMORE, A. K. Hyperspectral band depth analysis for a better estimation of grass biomass (*Cenchrus ciliaris*) measured under controlled laboratory conditions. *International Journal of Applied Earth Observation and Geoinformation*, v. 5, n. 2, p. 87–96, 2004. ISSN 03032434.
- [67] HATFIELD, J. Spectral behavior of wheat yield variety trials. photogrammetric engineering and remote sensing. *Proceedings of the Brazilian Symposium on GeoInformatics*, p. 209–214, 1981.
- [68] LIU, W. T. H. Aplicações de sensoriamento remoto. p. 908, 2006.
- [69] POVH, J. e. a. Comportamento do ndvi obtido por sensor ótico ativo em cereais. p. 1075–1083, 2008.