

UNIVERSIDADE FEDERAL DA BAHIA
INSTITUTO DE MATEMÁTICA

Paulo Henrique Sousa Cardoso

MÓDULO DE AEROFOTOGRAMETRIA PARA DRONES.

*Trabalho apresentado ao DEPARTAMENTO DE CIÊNCIA
DA COMPUTAÇÃO do INSTITUTO DE MATEMÁTICA
da UNIVERSIDADE FEDERAL DA BAHIA como requisito
parcial para obtenção do grau de Bacharel em SISTEMAS
DE INFORMAÇÃO.*

Orientador: Prof. Dr. Flávio Morais de Assis Silva

Salvador
23 de fevereiro de 2018

Ficha catalográfica.

Cardoso, Paulo Henrique

Módulo de aerofotogrametria para drones./ Paulo Henrique Sousa Cardoso– Salvador, 23 de fevereiro de 2018.

23p.: il.

Orientador: Prof. Dr. Flávio Morais de Assis Silva.
Monografia (graduação)– UNIVERSIDADE FEDERAL DA BAHIA, INSTITUTO DE MATEMÁTICA, 23 de fevereiro de 2018.

TOPICOS PARA FICHA CATALOGRAFICA.

I. Assis, Flávio Morais. II. UNIVERSIDADE FEDERAL DA BAHIA. INSTITUTO DE MATEMÁTICA. III Título.

NUMERO CDD

TERMO DE APROVAÇÃO

PAULO HENRIQUE SOUSA CARDOSO

MÓDULO DE AEROFOTOGRAMETRIA PARA DRONES.

Este Trabalho de Graduação foi julgado adequado à obtenção do título de Bacharel em Ciência da Computação e aprovada em sua forma final pelo Departamento de Ciência da Computação da Universidade Federal da Bahia.

Salvador, 23 fevereiro 2018

Prof. Dr.
Universidade Federal da Bahia

Prof. Dr.
Universidade Federal da Bahia

Prof. Dr.
Universidade Federal da Bahia

A todos que me ajudaram durante essa longa jornada

AGRADECIMENTOS

Education isn't something you can finish.
— ISAAC ASIMOV (Bill Moyer's World of Ideas, 1989)

RESUMO

Com a popularização dos drones nos ambientes industriais e residenciais, criou-se uma grande necessidade de desenvolver aplicações que pudessem utilizar dessa nova tecnologia, a fim de facilitar a realização de tarefas que antes somente era possível através do uso de aeronaves de grande porte como os helicópteros, aviões dentre outros meios de alto custo, como por exemplo satélites. A viabilidade dos drones pode facilitar a execução de uma determinada tarefa, porém, ainda traz a responsabilidade da pilotagem para o usuário, o que pode, ocasionalmente, resultar em acidentes ou em uma procedimento mal executado. Com isso aumentou-se a necessidade de automatizar o voo dessas aeronaves e o desenvolvimento de novas soluções em software para permitir uma utilização dos drones de uma maneira mais simples e eficaz. Entretanto o desenvolver de tal tecnologia é um trabalho árduo e que pode ser dividido em diversas etapas, desde o controle da aeronave até o processamento dos dados que foram recebidos. Este trabalho traz o processo de desenvolvimento de uma solução para o mapeamento de terrenos, através de fotos aéreas capturadas por meio de um ou mais drones. Essa solução é composta desde a definição das coordenadas geográficas, onde os drones devem se posicionar para efetuar a captura das imagens, até a representação gráfica através da composição dessas imagens.

Palavras-chave: Universidade Federal da Bahia, drone, voo autônomo, ar.drone, aerofotogrametria, quadricoptero, mecatrônica.

ABSTRACT

With the popularization of drones in industrial and residential environments, a great need was created to develop applications that could use this new technology, in order to facilitate the accomplishment of tasks that were previously possible only through the use of large aircraft such as helicopters, airplanes among other high-cost means, such as satellites. The viability of the drones can facilitate the execution of a certain task, but it still brings the responsibility of the piloting to the user, which can occasionally result in accidents or a poorly executed procedure. This increased the need to automate the flight of these aircraft and the development of new software solutions to enable drones to be used more simply and effectively. However, the development of such technology is hard work and can be divided into several stages, from the control of the aircraft to the processing of the data that has been received. This work brings the process of developing a solution for the mapping of terrains, through aerial photos captured through one or more drones. This solution is composed from the definition of geographic coordinates, where the drones must position themselves to capture the images, to the graphic representation through the composition of these images.

Keywords: Universidade Federal da Bahia, drone, autonomous flight, ar drone, aerophotogrammetry, quadcopter, mechatronics.

SUMÁRIO

Capítulo 1—Introdução	1
1.1 Objetivo do Projeto	1
1.2 Problema a ser resolvido	2
Capítulo 2—Arquitetura do Projeto	3
2.1 Componentes de hardware	3
2.1.1 O drone	3
2.1.2 Raspberry Pi 3	4
2.1.3 Módulo GPS	5
2.2 Software embarcado	6
2.2.1 Ps-Drone	6
2.2.2 CommandTower API	6
2.2.3 GPSD	7
Capítulo 3—Solução de aerofotogrametria utilizando drones	8
3.1 Algoritmo para gerar coordenadas dada uma área retangular	8
3.2 Servidor	10
3.2.1 Banco de dados	10
3.3 Processamento de imagens	11
3.3.1 OpenDroneMap	11
3.3.2 Ortorretificação das imagens	11
3.3.3 Georeferenciamento das imagens	13
3.4 Torre de comando	14
Capítulo 4—Trabalhos Correlatos	16
4.1 Pix4D	16
4.2 DroneDeploy	17
Capítulo 5—Tecnologias utilizadas	19
5.1 NodeJs	19
5.2 Docker	19
5.2.1 Docker compose	19
5.3 Sequelize	20
5.4 Git	20

SUMÁRIO

x

Capítulo 6—Avaliação

21

Capítulo 7—Conclusão

22

LISTA DE FIGURAS

2.1	ArDrone 2.0 com o <i>case outdoor</i>	3
2.2	Raspberry Pi 3 model B	5
2.3	Módulo GPS GY-NEO6MV2	6
2.4	Interface de texto da aplicação GPSD	7
3.1	Google Drawing tools	9
3.2	Resultado do processamento do algoritmo	10
3.3	Interface de acompanhamento do processamento das imagens	12
3.4	Tela para download dos resultados	12
3.5	Resultado da ortorretificação de 18 imagens	13
3.6	Tela de definição da missão	14
3.7	Tela após missão finalizada	15
4.1	Tela de definição de plano de voo	17
4.2	Definição de área para sensoriamento no DroneDeploy App	18

LISTA DE TABELAS

Capítulo

1

Contextualiza esta monografia, trazendo o histórico, a motivação para o trabalho e a organização dos capítulos.

INTRODUÇÃO

Os drones estão ficando cada vez mais eficientes e de baixo custo. O grande potencial que os drones têm de efetuar tarefas, que podem ser perigosas ou de difícil execução, trazem investimentos nessa área que são cada vez maiores.

Além disso a aplicabilidade destes drones na área militar movimentou ainda mais a busca pelo desenvolvimento de novas tecnologias, que permitissem o controle remoto e a automação destes drones em diversos tipos de tarefas, uma delas é o processo de reconhecimento territorial.

Na área acadêmica existem vários trabalhos destinados as aeronaves não tripuladas. A exemplo disto, o processo de criação mecânica de um drone hexacoptero em ([DEMOLINARI, 2016](#)), e o desenvolvimento de algoritmos embarcados em drones de asa fixa, que são capazes de identificar e desviar de obstáculos em um voo de até 32 Km/h em ([BARRY, 2015](#)).

O desenvolvimento deste trabalho traz a descrição e implementação de um dos módulos para um sistema distribuído de voos autônomos. Este módulo é capaz de gerar os pontos onde o drone deve efetuar a captura de imagens e a partir dessas imagens é criada uma única representação aérea de alta definição.

Foi implementado também uma aplicação REST (Representational State Transfer) que intermedia o fluxo dos dados entre o drone e a interface do usuário. Além disso, este trabalho faz uso de um kit de ferramentas de código aberto para o processamento de imagens aéreas do drone chamado OpenDroneMap.

1.1 OBJETIVO DO PROJETO

O objetivo do projeto final é o desenvolvimento de um sistema genérico de voo para drones onde nesse sistema será possível atribuir e acompanhar missões definidas pelo usuário, essas missões podem ser atribuídas a uma ou mais aeronaves o que não afetará o cumprimento da missão, ou seja, os drones devem ser capazes de se locomover em sinergia, identificando e desviando de objetos assim como corrigindo a rota se caso ocorra alguma interferência externa como correntes de ar.

1.2 PROBLEMA A SER RESOLVIDO

A fim de atingir viabilizar o objetivo final do projeto, foi inicialmente desenvolvido um algoritmo de planejamento de voo para drones que utilizam um sensor GPS (*geographic positioning system*). Este algoritmo recebe como entrada as coordenadas geográficas de destino. Durante o voo o algoritmo verifica se o drone está na direção correta, caso não esteja, o algoritmo recalcula as variáveis de distância e angulação que resultam em comandos de correção do movimento. A implementação e definição de todos os procedimentos deste algoritmo são encontrados em (MARTINS, 2017).

Apesar da implementação do algoritmo, descrito no parágrafo anterior, atender uma série de requisitos, que são importantes para o objetivo final do projeto, algumas limitações impediram a utilização de apenas este algoritmo, para o desenvolvimento deste trabalho.

Uma das limitações encontradas foi a ausência da definição da altitude, na qual o drone atua durante o voo até o destino. A altitude nesta solução é completamente fixa não existindo a possibilidade de alteração durante o voo. Outra limitação foi a ausência da definição de vários pontos geográficos restringindo-se a apenas a um único ponto de destino.

Na fase atual do projeto foi desenvolvido uma nova versão do módulo de planejamento de voo. Essa nova versão descrita em (FÁBIO, 2018) trazem implementações a fim de sanar algumas das limitações descritas anteriormente como: a definição de vários pontos de destino e altitude que o drone deve estar em cada um dos pontos de destino. A nova versão do módulo de planejamento recebe como parâmetro uma lista de coordenadas geográficas (latitude/longitude) seguido da altitude em metros, onde o drone deve estar ao chegar aquele ponto.

Neste trabalho é descrito um conjunto de soluções que viabilizam a definição e o acompanhamento de missões que tem como objetivo inicial a aerofotogrametria. Dentro desse conjunto de soluções foi implementada uma interface, onde um usuário pode acessar através de um navegador *web*. Através dessa interface o usuário pode definir uma área de mapeamento e a partir dessa área são gerados, através de um algoritmo, os pontos de destino para o drone designado.

Os dados gerados pela aplicação *web* são enviados a um servidor, por meio de requisições HTTP (*Hypertext Transfer Protocol*) e esses dados são acessados pelo drone da mesma forma. Este servidor permite a persistência dos dados e a validação dos dados de entrada.

O processamento das imagens que são capturadas pelo drone é feita por meio de uma biblioteca de código aberto. Esta biblioteca recebe uma série de imagens devidamente referenciadas com as respectivas latitudes e longitudes e, a partir disso é feito um procedimento para juntar as imagens, reduzir as distorções e criar uma representação em 2D, processo chamado de ortorretificação.

Capítulo

2

Este capítulo descreve todos os componentes utilizados.

ARQUITETURA DO PROJETO

2.1 COMPONENTES DE HARDWARE

2.1.1 O drone

O drone utilizado no projeto é o AR Drone 2.0, desenvolvido pela a empresa Parrot. Este drone foi escolhido por ser um dispositivo de baixo custo financeiro e entregar o necessário para a execução de tarefas triviais como, ir de um ponto A a um ponto B.

O drone possui estrutura para voo *indoor* e *outdoor*, a diferença entre essas duas estruturas é que a resistência aerodinâmica da estrutura *outdoor* é menor, pois a *indoor* é direcionada à segurança dos componentes.

Figura 2.1 ArDrone 2.0 com o *case outdoor*.



O ArDrone vem equipado de fábrica com alguns dispositivos descritos na lista a seguir:

- Câmera de video frontal HD (*High Definition*) 720p 30fps (*frames per second*)

- Câmera vertical QVGA (*Quarter Video Graphics Array*) 60fps utilizado para mensurar a velocidade
- Sensor de ultrassom vertical utilizado para mensurar a altitude
- Giroscópio de 3 eixos com a precisão de 2000°/segundos
- Acelerômetro de 3 eixos com a precisão de aproximadamente 50 mg
- Magnetômetro com 6 graus de precisão
- Sensor de pressão com uma precisão de aproximadamente 10 Pa (Pressão atmosférica)

O ArDrone foi desenvolvido para ser um dispositivo controlado por smartphones ou tablets. Para viabilizar isso o drone possui o seu próprio módulo embarcado, responsável por receber comandos e transformá-los em movimento. O hardware responsável por esse processamento é composto por:

- Processador 1GHz 32 bit ARM Cortex A8 com 800MHz video DSP TMS320DMC64x
- 1GB DDR2 RAM 200MHz
- USB 2.0
- Módulo Wi-Fi
- Sistema operacional Linux 2.6.32

A alimentação do ArDrone é feita por meio de uma bateria de 2000mah(*milliampere hour*) que permite uma autonomia de 15 a 20 minutos de voo.

Apesar do drone possuir um hardware capaz de embarcar *software* os recursos são limitados, impossibilitando a adição de novos sensores ou de um processamento mais robusto. Por causa disso foi adicionado a estrutura do drone um *Raspberry Pi 3*.

2.1.2 Raspberry Pi 3

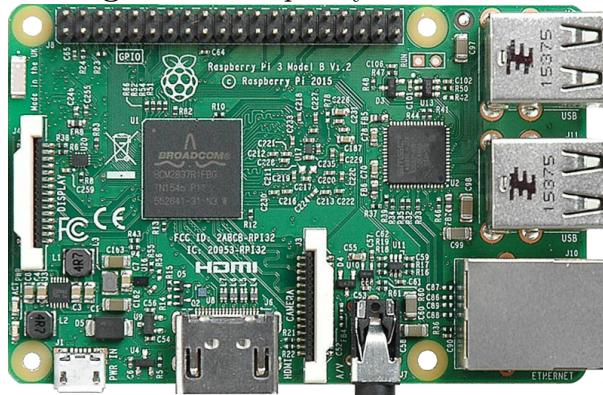
O Raspberry Pi é um computador completo de tamanho reduzido, criado para projetos educacionais de baixo custo. Esse dispositivo possui apenas 41.2g (gramas) de peso total e dimensões similares a de uma cartão de crédito.

A versão utilizada no projeto é o Raspberry Pi 3 model B, esta versão vem equipada com vários recursos relevantes para o projeto dentre elas:

- Processador Broadcom BCM2837 64bit ARMv8 Cortex-A53 Quad-Core
- Clock 1.2 GHz
- Memória RAM: 1GB

- Adaptador Wifi 802.11n integrado
- Bluetooth 4.1 BLE (*Bluetooth low energy*) integrado
- Conector de vídeo HDMI
- 4 portas USB 2.0
- Conector Ethernet
- Interface para câmera (CSI)
- Interface para display (DSI)
- Slot para cartão microSD
- Conector de áudio e vídeo
- GPIO de 40 pinos
- Dimensões: 85 x 56 x 17mm

Figura 2.2 Raspberry Pi 3 model B



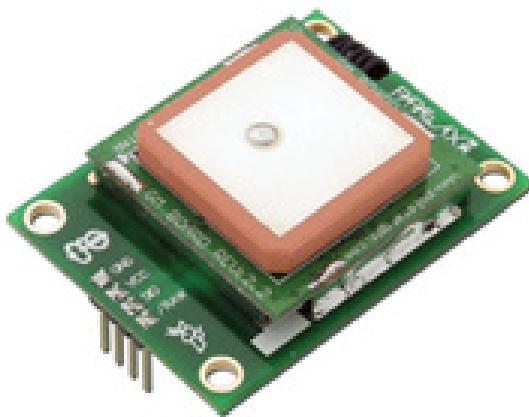
2.1.3 Módulo GPS

O GPS utilizado no projeto é o GPS GY-NEO6MV2. Por ser um módulo que possui compatibilidade com o Raspberry Pi 3, podendo transmitir dados e ser alimentado através da porta serial.

- Alimentação: 2.7 à 5V DC
- Corrente de operação: 45 mA (Miliampere)
- Comunicação serial

- Antena embutida
- Precisão: 5m (Metros)
- Dimensões: 35 x 25 x 25mm (Milímetros)

Figura 2.3 Módulo GPS GY-NEO6MV2



2.2 SOFTWARE EMBARCADO

2.2.1 Ps-Drone

O PS-Drone é uma API completa, escrita em Python, para o AR.Drone 2.0. Esta API fornece uma série de métodos que correspondem a movimentação do drone e a utilização dos seus componentes de vídeo. Este *software* é embarcado no Raspberry Pi 3 e utilizado no algoritmo de controle descrito em (FÁBIO, 2018).

2.2.2 CommandTower API

A CommandTower API foi desenvolvida para facilitar a utilização dos recursos fornecidos pelo servidor, responsável por registrar uma nova missão. Esta API foi criada a fim de remover a responsabilidade do desenvolvedor de implementar requisições as rotas com os métodos HTTP's corretos para acessar os recursos do servidor.

- `getAPIStatus()` - Retorna o status atual do servidor.
- `newDrone(droneName)` - Registra um novo drone no servidor.
- `getDroneCoords(droneName)` - Recebe todas as coordenadas da missão atual do drone.

- `updateMissionStatus(status, missionId)` - Atualiza o status da missão.

Esta API foi desenvolvida em Python e é utilizada no algoritmo de controle descrito em (FÁBIO, 2018), para a troca de informações entre o drone e o servidor.

2.2.3 GPSD

O GPSD é a aplicação utilizada para inicializar e coletar informações do módulo de GPS. Essa aplicação é começa após a inicialização do Raspberry Pi 3 e fica em *background* fazendo coleta de informações como:

- Latitude e longitude
- Velocidade
- Quantidade de satélites disponíveis

Quando em *foreground* a aplicação exibe uma interface de texto, utilizada para validar informações sobre o GPS durante os testes.

Figura 2.4 Interface de texto da aplicação GPSD

```

Time: 2014-12-26T22:01:14.000Z Lat: 40 43' 34.176" N Lon: 74 00' 17.291" W
Cooked PVT

GPGGA GPGSA GPRMC GPVTG GPGSV
Sentences

Ch PRN Az El S/N
0 22 194 77 22
1 14 323 66 25
2 18 148 41 23
3 25 125 38 24
4 31 225 34 36
5 12 74 34 38
6 51 225 31 32
7 4 296 23 23
8 24 49 22 32
9 32 313 16 26
10 1 323 9 21
11 11 302 8 0
GSV

Time: 220114.000
Latitude: 4043.5696 N
Longitude: 07400.2882 W
Speed: 0.25
Course: 145.29
Status: A FAA: D
MagVar: 0
RMC

Time: 220114.000
Latitude: 4043.5696
Longitude: 07400.2882
Altitude: 33.5
Quality: 2 Sats: 09
HDOP: 0.91
Geoid: -34.2
GGA

Mode: A 3
Sats: 22 12 31 24 32 25 4 1
DOP: H=0.91 V=0.86 P=1.26
GSA

UTC: RMS:
MAJ: MIN:
ORI: LAT:
LON: ALT:
GST

```

SOLUÇÃO DE AEROFOTOGRAFETRIA UTILIZANDO DRONES

Com o desafio de desenvolver uma solução de aerofotogrametria utilizando drones, o primeiro desafio foi definir o posicionamento geográfico adequado para que o drone efetuasse a captura das imagens. Além disso a altura que o drone está em relação ao solo influencia diretamente na qualidade da imagem e consequentemente na qualidade da imagem ortorretificada.

Diante disto foram listados alguns itens essenciais para o desenvolvimento de uma solução adequada.

1. Determinar as coordenadas (latitude/longitude) de atuação do drone, bem como a altura em relação ao solo.
2. Associar as coordenadas a um drone.
3. Enviar as coordenadas para o drone.
4. Receber as imagens captadas pelo drone e transformar em uma única imagem ortorretificada.

Para solucionar os requisitos listados no parágrafo anterior, foram desenvolvidas uma serie de aplicações que buscam sanar cada item.

3.1 ALGORITMO PARA GERAR COORDENADAS DADA UMA ÁREA RETANGULAR

Para a geração de pontos foi desenvolvido um algoritmo em *javascript*, capaz de atender o primeiro requisito citado no tópico anterior. Para que a visualização do resultado do algoritmo, fosse mais interativa ao usuário final, foi adicionada a ferramenta GoogleDrawingTools que possibilita criar interações com um mapa, através de formas geométricas como retângulos e polígonos.

Drawing tools



Figura 3.1 Google Drawing tools

Através da API fornecida por essa ferramenta, foi possível identificar as coordenadas geográficas das arestas, sendo elas, respectivamente, da esquerda para direita: noroeste, nordeste, sudoeste e sudeste. Para desenvolver um grid nessa área, inicialmente demarcada, são consideradas três variáveis sendo elas:

1. A distância horizontal.
2. A distância vertical.
3. A área da foto.

A distância horizontal e vertical, são alcançadas através da aplicação da fórmula de Haversine, que basicamente, permite o cálculo de distância entre duas coordenadas geográficas. Sendo assim a distância horizontal é definida como a distância entre o ponto nordeste e o ponto noroeste e a distância vertical como a distância entre o ponto nordeste ao ponto sudeste.

A área da foto foi definida inicialmente, por padrão, como 2 metros quadrados pois a qualidade da imagem é prejudicada com áreas maiores. Após a verificação dessas três variáveis é feita a divisão entre, a distância horizontal e a área da foto, assim como a divisão entre, a distância vertical e a área da foto. Essas duas divisões resultam em números que definem a quantidade de pontos horizontais e verticais que serão gerados. Para a definição dos pontos centrais é feito um cálculo que consiste basicamente em:

$$\text{latitude do ponto atual} + (\text{área da foto} * \text{latitude em metros})/2$$

O mesmo procedimento é feito na horizontal utilizando a longitude, esses cálculos geram novas combinações de latitude e longitude que são os pontos onde o drone deve estar posicionado para a captura das imagens.



Figura 3.2 Resultado do processamento do algoritmo

3.2 SERVIDOR

Para intermediar a comunicação entre o drone e a interface do usuário, foi desenvolvido um servidor. Este servidor é responsável por persistir os dados sobre a missão e disponibilizar esses dados quando requisitado.

A comunicação com o servidor é feita através do protocolo HTTP (*Hypertext Transfer Protocol*) protocolo de comunicação descrito por ([BERNERS-LEE, 1996](#)) como um protocolo com a leveza e velocidade necessária para sistemas de informação de hipermídia distribuídos e colaborativos.

O modelo arquitetural adotado para a aplicação denominado como REST (Representational State Transfer) inicialmente descrito por ([FIELDING, 2000](#)) como um estilo híbrido, derivado de outros modelos de arquitetura, combinado com restrições adicionais que definem uma interface de conexão uniforme. A principal abstração do modelo REST é a definição de recursos, onde qualquer informação pode ser denominada como um recurso, como explicado em ([FIELDING, 2000](#)), qualquer conceito que possa ser alvo de referência de hipertexto está presente na definição de recurso.

A implementação do servidor foi feita utilizando a linguagem de programação *javascript* por meio da tecnologia *Nodejs* descrita no topico **XX**

3.2.1 Banco de dados

O armazenamento das informações associadas a missão e a sua execução, são armazenadas em um banco de dados relacional MySQL. A escolha por um banco de dados relacional foi necessária devido as regras de relacionamento e associações definidas para o escopo atual do projeto.

Além disso, este tipo de banco de dados permite a implementação de transações, onde um conjunto de procedimentos são executados de maneira atômica e se caso ocorra alguma interrupção na execução, todos os procedimentos descritos naquela transação são desfeitos.

3.3 PROCESSAMENTO DE IMAGENS

3.3.1 OpenDroneMap

O OpenDroneMap como descrito em ([TOFFANI, 2017a](#)), é um conjunto de ferramentas de código aberto para o processamento de imagens aéreas. Essa aplicação facilita a criação de:

- Representação aérea 3D através da técnica de point cloud.
- Modelos de superfície digital.
- Modelos de superfície digital texturizados.
- Representação 2D através de ortorretificação de imagens.

A utilização do OpenDroneMap foi através de uma REST API feita para facilitar o acesso aos seus recursos. O projeto Node-OpenDroneMap é uma aplicação Open Source que cria uma interface de interação sobre o OpenDroneMap onde é possível interagir através de requisições HTTP ou por meio de um navegador *web*.

O Node-OpenDroneMap é um projeto *cross-platform* o que torna acessível pelos sistemas operacionais mais comuns. O processo de configuração é simples, basta executar o comando: "`docker run -p 3000:3000 opendronemap/node-opendronemap`".

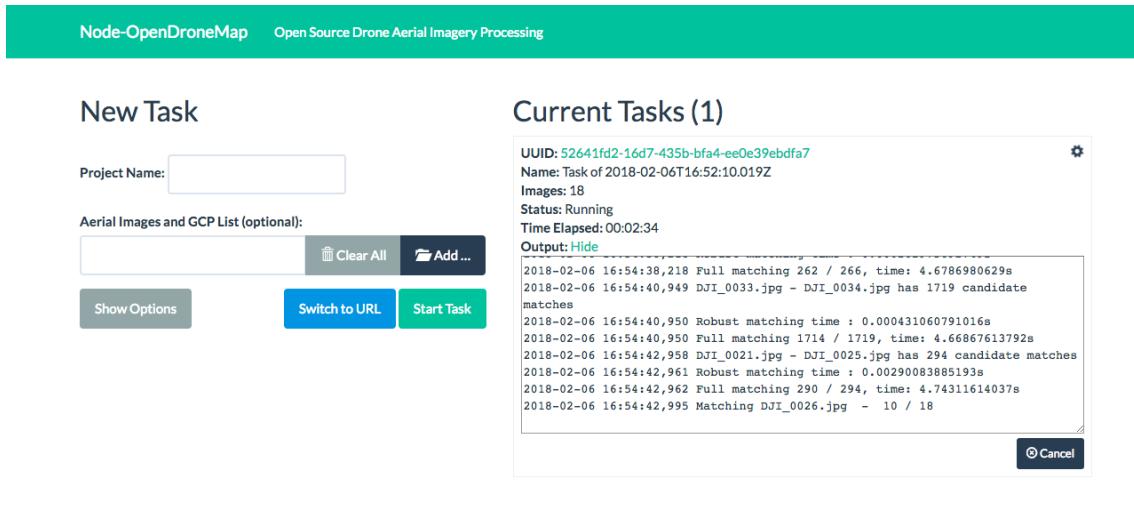
E assim será feito o download de todas as dependências necessárias para ter o serviço funcionando corretamente. A documentação completa da interface REST pode ser encontrada em ([TOFFANI, 2017b](#)). Após a captura das imagens feitas pelo drone, o usuário poderá então acessar a interface gráfica da ferramenta, através de um *browser* e efetuar o upload das imagens, nesse processo é feita uma verificação se as imagens são do tipo correto e possuem coordenadas associadas a elas. caso todas as validações sejam atendidas então o processo de ortorretificação é iniciado e o usuário poderá acompanhar o andamento pela própria plataforma.

Ao finalizar o processamento das imagens a aplicação indica ao usuário que por sua vez pode fazer o download do resultado do processamento das imagens através do botão "Download Orthophoto" como pode ser observado na [3.4](#).

3.3.2 Ortorretificação das imagens

O processo de ortorretificação das imagens consiste na projeção dessas imagens em um plano ortogonal das imagens aéreas utilizadas. Esse procedimento é muito comum para a composição de imagens de satélite.

Da mesma forma que, as fotos aéreas captadas pelo drone, as imagens de satélite também podem sofrer com as distorções por causa da inclinação da câmera e da variação de altitude. O processo para a eliminação dessas distorções, com a finalidade de reduzir esses erros, é chamado de ortorretificação, sendo a imagem resultante passa a ser denominada ortoimagem.



Links: [API Docs](#) | [WebODM](#)
This software is released under the terms of the [GPLv3 License](#). See [node-opendronemap](#) on Github for more information.

Figura 3.3 Interface de acompanhamento do processamento das imagens

Current Tasks (1)



Figura 3.4 Tela para download dos resultados

A ortorretificação baseia-se em um modelo matemático, o qual realiza a correspondência entre o espaço-imagem e o espaço-objeto, estabelecendo as correções das distorções que ocorrem na formação da imagem ([GEOPIXEL, 2017](#))

Para que o processo de ortorretificação ocorra, é necessário associar imagens a um conjunto de coordenadas, através de metadados agregados ao próprio arquivo de imagem.

Utilizando o dataset de imagens aéreas de um parque da cidade americana New York, disponível em ([ODMCOMMUNITY, 2017](#)), com um conjunto de 18 imagens de alta resolução, georreferenciadas é possível visualizar o resultado criado pelo OpenDroneMap na figura 3.5.



Figura 3.5 Resultado da ortorretificação de 18 imagens

3.3.3 Georeferenciamento das imagens

O processo de georeferenciamento das imagens consiste em adicionar metadados a um arquivo de imagem contendo informações referentes ao local e a posição em que as imagens foram capturadas.

Em drones mais avançados esse procedimento é feito automaticamente quando uma foto aérea é capturada. A exemplo disso, o drone *Phantom 4* da empresa DJI faz o georeferenciamento de todas as imagens capturadas pela câmera principal ([DJI, 2017](#)).

O ArDrone 2.0 não possui GPS nas suas especificações, impossibilitando o processo de georeferenciamento seja feito pelo próprio drone. Com isso foi necessário desenvolver um *script* em python capaz de adicionar a imagem os metadados necessários. Para que esse processo seja efetuado foi adicionado ao *Raspberry Pi 3* uma ferramenta chamada *ExifTool*.

O *ExifTool* é um *software* de código aberto capaz de ler, escrever e manipular metadados em diversos tipos audio, imagem, video e PDF(*Portable Document Format*) ([EXIFTOOL, 2017](#)).

Ao término da missão, o *script* de georeferenciamento é inicializado e começa a asso-

ciando imagens as coordenadas e altitudes, que foram recebidas do servidor no inicio da missão. No final da execução todas as imagens estão corretamente referenciadas.

3.4 TORRE DE COMANDO

Com o propósito de gerenciar os drones e acompanhar o andamento das missões foi desenvolvida a aplicação chamada torre de comando. Esta aplicação, na fase atual do projeto, possui a definição de missões para aerofotogrametria, porém outros conjuntos de soluções podem ser agregados a essa aplicação.

O algoritmo para gerar coordendas dada uma área retangular, descrito na seção 3.1, foi acoplado a aplicação de forma que um usuário possa definir os pontos de uma nova missão por meio do algoritmo.

Além disso o usuário faz define o nome da missão e o drone responsável para efetuar a missão.

Torre de comando

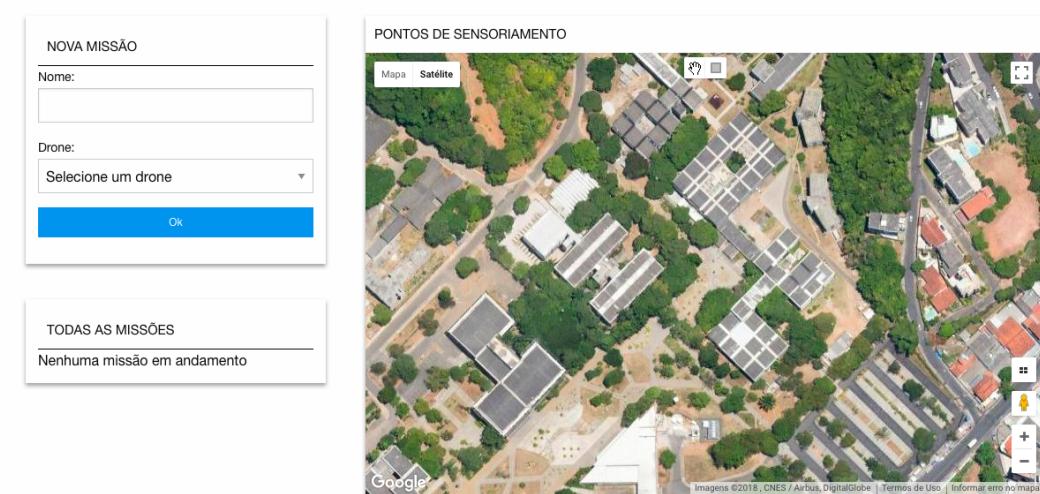


Figura 3.6 Tela de definição da missão

Após a definição dos pontos a aplicação faz uma requisição HTTP ao servidor, descrito na seção 3.2, registrando uma nova missão. Essa nova missão registrada fica com o status pendente no servidor até que o drone faça uma requisição solicitando uma missão. Depois de finalizar a missão o drone atualiza o status da missão no servidor indicando a conclusão da missão.

Quando finalizada a missão é exibido para o usuário uma nova opção chamada "Processar Imagens", como pode ser observado na figura 3.7, que direciona o usuário para a ferramenta de processamento de imagens, exibida na figura 3.3.

Torre de comando



Figura 3.7 Tela após missão finalizada

Capítulo

4

Este capítulo lista alguns trabalhos relacionados

TRABALHOS CORRELATOS

Algumas empresas que atuam no mercado de drones já desenvolvem soluções similares a solução apresentada neste trabalho, entretanto, possuem limitações e custos que podem inviabilizar o processo de aerofotogrametria para quem possui um drone de baixo custo. Foram escolhidas duas empresas que desenvolvem soluções que possuem funcionalidades e aplicações relacionadas a este trabalho: Pix4D e DroneDeploy.

4.1 PIX4D

Criada em 2011 a empresa Pix4D nasceu com o objetivo de criar mapas georeferenciados e modelos a partir de imagens de drones. As aplicações de software dessa empresa possui um aplicativo móvel chamado de Pix4DCapture e uma aplicação para desktop chamada de Pix4D Desktop. Ambas apresentam o mesmo objetivo final, o sensoriamento de áreas previamente definidas pelo o usuário, para isso as aplicações resumem a interação do usuário em 5 passos principais: selecionar um drone, selecionar o tipo da missão, ajustar plano de voo e parâmetros, iniciar voo e checar resultados.

Atualmente os softwares da Pix4D podem ser utilizados em 14 diferentes drones das marcas DJI ou Parrot, duas das maiores fabricantes de drones no mercado, dando suporte também a drones de asa fixa.

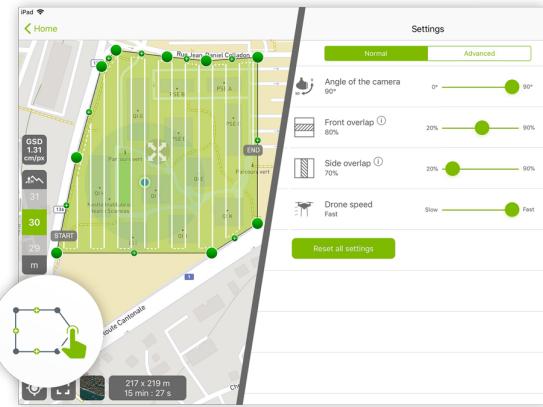


Figura 4.1 Tela de definição de plano de voo

4.2 DRONEDEPLOY

A DroneDeploy provê ao usuário um conjunto de soluções de software para que um usuário ou uma empresa possa gerenciar os dados do processo de mapeamento de uma determinada área. Um dos objetivos desta empresa é tornar o processo de voo, seguro, rápido e fácil para quem for manusear, a fim de tranquilizar o usuário quanto ao voo do drone e manter-lo focado nos dados resultantes. Esta empresa possui diversos softwares direcionados a cenários diferentes, atualmente a DroneDeploy disponibiliza duas aplicações sendo elas: Mobile app e a enterprise. A aplicação mobile é gratuita e permite que qualquer usuário, que possua um drone fabricado pela a empresa DJI, possa criar mapas 2D ou modelos 3D de uma região demarcada, além de possibilitar que o usuário assuma o controle do drone durante o voo, caso o mesmo julgue necessário.

Já a aplicação enterprise permite gerenciar projetos, mapas e modelos 3D através de times colaborativos e os dados gerados pelos drones podem ser acessados por vários usuários, em qualquer lugar, por qualquer computador que possua conexão com a internet, além disso, esta aplicação disponibiliza um dashboard para o administrador poder acompanhar as atividades de um time, como por exemplo as áreas que foram mapeadas e os resultados gerados.

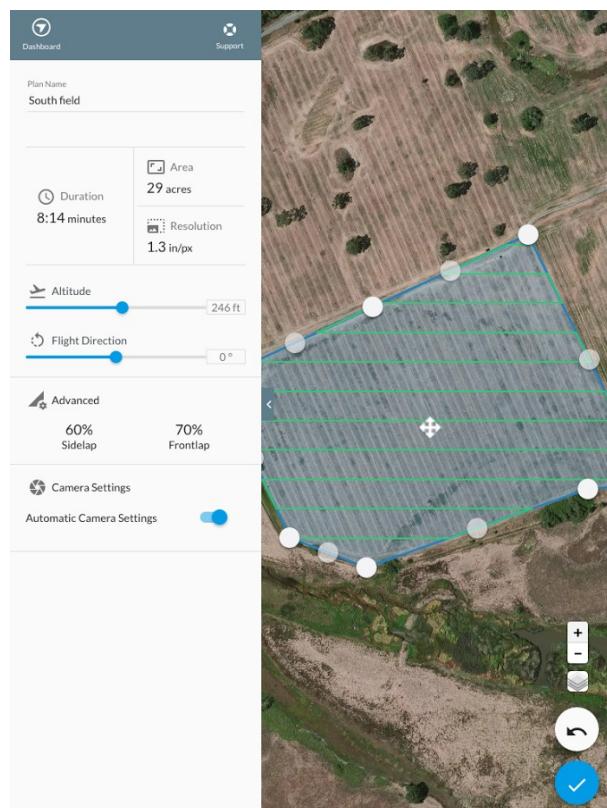


Figura 4.2 Definição de área para sensoriamento no DroneDeploy App

Capítulo

5

TECNOLOGIAS UTILIZADAS

5.1 NODEJS

O NodeJs é um interpretador de código javascript muito usado para o desenvolvimento de aplicações server-side por possuir muitas propriedades que aumentam o desempenho e a eficácia do servidor onde ele for executado.

Uma das principais características dessa tecnologia é que utiliza programação assíncrona, isto quer dizer que, a ordem das chamadas dos métodos nem sempre será a ordem de execução.

5.2 DOCKER

A ideia de ter um servidor responsável por receber dados de uma aplicação e disponibilizá-la para o drone, possibilita que esse servidor seja disponibilizado como um webservice em qualquer serviço de hospedagem, para facilitar o processo de instalação das dependências e automatizar procedimentos iniciais, foi adicionado uma ferramenta que permite a virtualização de ambientes chamada Docker.

O Docker é uma plataforma desenvolvida com a linguagem de programação Go que facilita a administração e criação de ambientes virtuais.

O grande diferencial dessa plataforma de virtualização é que os seus containeres, como são chamados seus ambientes, possuem apenas os recursos necessários para que uma determinada aplicação possa entrar em funcionamento, ao contrário das plataformas de virtualização mais comuns, que fazem uma cópia de todo um sistema operacional, o docker apenas utiliza alguns módulos essenciais do kernel.

5.2.1 Docker compose

O Docker compose é uma ferramenta que auxilia no gerenciamento e definição de aplicações que possuem muitos containers e permite a criação e o inicio desses containers através de um único comando. O arquivo de configuração do docker compose pode ser encontrado no diretório raiz da aplicação como o nome docker-compose.yml que possui todas as informações necessárias para a configuração destes containers.

5.3 SEQUELIZE

Sequelize é uma ORM (object-relation mapping) de código aberto para ambientes NodeJs, feito para viabilizar a conexão e facilitar as operações em banco de dados MySQL. Através desta ferramenta foi possível gerar migrações, definir modelos, gerenciar associações entre modelos e criar operações de transação na aplicação descrita na seção 3.2.

5.4 GIT

Para o controle de versão de todas as aplicações foi utilizado o Git um software livre multiplataforma que permite o controle de versão distribuído desde pequenos projetos aos mais extensos, o Git possui diversas formas de interação o usuário pode interagir por meio de linha de comandos ou através de interfaces gráficas.

Capítulo

6

AVALIAÇÃO

Capítulo

7

CONCLUSÃO

REFERÊNCIAS BIBLIOGRÁFICAS

- BARRY, A. J. Pushbroom stereo for high-speed navigation in cluttered environments. 2015. Disponível em: <<http://hdl.handle.net/1721.1/101080>>.
- BERNERS-LEE, T. Hypertext transfer protocol. 1996. Disponível em: <<http://www.ietf.org/rfc/rfc1945.txt>>.
- DEMOLINARI, H. C.
- PROJETO DE CONSTRUÇÃO UM DRONE HEXACÓPTERO* — UNIVERSIDADE FEDERAL FLUMINENSE, 2016. Disponível em: <[http://www.repositorio.uff.br/jspui/bitstream/1/1792/1/PROJETODECONSTRUÇAOUMDRONEHEXACOPTERO\(SEMAssinaturas\).pdf](http://www.repositorio.uff.br/jspui/bitstream/1/1792/1/PROJETODECONSTRUÇAOUMDRONEHEXACOPTERO(SEMAssinaturas).pdf)>.
- DJI. *dji*. 2017. Acessado em Dezembro de 2017. Disponível em: <<https://www.dji.com/phantom-4>>.
- EXIFTOOL. *Exiftool*. 2017. Acessado em Dezembro de 2017. Disponível em: <http://owl.phy.queensu.ca/~phil/exiftool/exiftool_pod.pdf>.
- FIELDING, R. T. *Hypertext Transfer Protocol*. 2000. Disponível em: <<https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>>.
- GEOPIXEL. *OpenDroneMap*. 2017. Acessado em Novembro de 2017. Disponível em: <<http://www.geopx.com.br/servicos/2/processamento-digital-de-imagens>>.
- MARTINS, M. A. M.
- Sistema de Controle de Rota para Voo Autônomo de Drones Baseado em Dados de GPS* — UNIVERSIDADE FEDERAL DA BAHIA, 2017.
- ODMCOMMUNITY. *OpenDroneMap*. 2017. Acessado em Novembro de 2017. Disponível em: <<http://community.opendronemap.org/>>.
- TOFFANI, P. *OpenDroneMap*. 2017. Acessado em Outubro de 2017. Disponível em: <<https://github.com/OpenDroneMap>>.
- TOFFANI, P. *OpenDroneMap*. 2017. Acessado em Outubro de 2017. Disponível em: <<https://github.com/OpenDroneMap/node-OpenDroneMap/blob/master/docs/index.adoc>>.