

## ***QPlanner: Módulo para Planejamento de Voo no Software QGIS***

**Frederyco Augusto Pereira Elleres<sup>1</sup>, Carlos Rodrigo Tanajura Caldeira<sup>1</sup>, Mayara Ortega Caldeira<sup>1</sup>, Alan José Salomão Graça<sup>2</sup>**

<sup>1</sup>Instituto Ciberespacial – Universidade Federal Rural da Amazônia (UFRA)  
Belém – PA – Brasil

<sup>2</sup>Departamento de Engenharia Cartográfica – Universidade do Estado do Rio de Janeiro (UERJ)  
Rio de Janeiro – RJ – Brasil.

fredelleres@gmail.com, {carlos.caldeira, mayara.caldeira}@ufra.edu.br, alan.salomao@eng.uerj.br

**Abstract.** *If the photography is to satisfactorily serve its intended purposes, the photographic mission must be carefully planned and faithfully executed according to the “flight plan.” The objective of this paper was to develop a flight planning plugin for GIS QGIS. QGIS and Python documentation were used to develop the plugin through code implementation, where we used Python-based programming techniques, called PyQGIS. From this, there is an experimental flight planning plugin produced by QGIS, where it requests the user the essential parameters for the calculations in an automated way, obtaining as products the ground surface area covered by a block of photographs (polygon), the point clouds, where each point represents the central coordinate of the photograph taken during flight and the results of calculations of the variables required for flight planning. Given the results obtained by the plugin, the products and the time taken to generate them were compared with the mechanically performed operations. Concluding the great advantage of the plugin over the tutorial procedure, helping to save time and minimize the risk of mistakes.*

**Resumo.** *O planejamento de voo é uma das etapas primárias, no que diz respeito aos produtos fotogramétricos, fazendo com que o mesmo, deva ser bem planejado e executado. Dessa forma, o objetivo desse artigo, consistiu em desenvolver um módulo de planejamento de voo utilizando a plataforma de Sistema de Informação Geográfica QGIS. Foram usados manuais, métodos e bibliotecas do QGIS e Python, para desenvolver o complemento por meio da implementação de códigos, onde utilizou-se de técnicas de programação, com base na linguagem Python, chamada de PyQGIS. A partir disso, dispõe-se de um módulo experimental de planejamento de voo projetado para o QGIS, onde o mesmo solicita ao usuário os parâmetros essenciais para os cálculos de forma automatizada, obtendo como produtos a poligonal da área de sobrevo, a nuvem de pontos, em que, cada ponto representa a coordenada central da fotografia capturada durante o voo e os resultados dos cálculos das variáveis necessárias para o planejamento de voo. Diante dos resultados obtidos pelo complemento, comparou-se, os produtos e o tempo levado para gerá-los, com as operações feitas mecanicamente. Concluindo-se a grande*

*vantagem do complemento sobre o procedimento manual, contribuindo para economizar tempo e minimizar o risco de erros grosseiros.*

## 1. Introdução

A principal tarefa do levantamento aerofotogramétrico é o registro tridimensional de recursos naturais e artificiais no solo [Eisenbeiss e Sauerbier, 2011]. Para o sucesso de um levantamento aerofotogramétrico é necessário um bom planejamento de voo, determinando as ferramentas a serem utilizadas e o orçamento que precisará ter para subsidiar o aerolevantamento. Com isso, existe a necessidade de automatizar o processamento do plano de voo para obter dados precisos sobre a quantidade de fotos a serem tiradas e a área a ser mapeada. À medida que um maior número de usuários não-especialistas têm realizado voos fotogramétricos com RPAs (*remotely piloted aircraft*), há uma popularização do uso de APIs (*Application Programming Interface*) como por exemplo *Pix4D Capture* (<https://www.pix4d.com/product/pix4dcapture>), *DJI Go* (<https://www.dji.com/br/goapp>), *Drone Deploy* (<https://support.dronedeploy.com>), entre outros destinados ao planejamento da missão e cobertura fotogramétrica.

O progresso da Fotogrametria, nos últimos anos, deve-se em grande parte às contribuições da tecnologia da informação, uma vez que cálculos matemáticos da fotogrametria analítica podem ser programados com resultados adequados, principalmente em termos de economia de tempo e maior precisão nos trabalhos fotogramétricos [Amorim et al., 2006]. Em decorrência desses avanços tecnológicos, a maioria dos processamentos fotogramétricos se dão através de imagens digitais e recursos de visão computacional, onde fotografias podem ser compreendidas como um conjunto regular de *pixels*, os quais são descritos por sua geometria e radiometria, mas também podem ser especificados por outros elementos, como o ângulo determinado por um pixel e a distância focal [Berveglieri, 2014].

Nesse contexto da Fotogrametria Digital para a extração de feições terrestres, o planejamento de voo é uma etapa crucial da missão para a orientação das imagens aéreas e a geração subsequente dos produtos fotogramétricos. A maioria dos planos de voo inclui um conjunto de especificações detalhadas que descrevem os materiais, equipamentos e procedimentos a serem usados no projeto [Wolf et al., 2014]. No universo dos serviços de planejamento de voo, *softwares* proprietários, mesmo com pagamento por licenças de uso, ainda assumem uma posição privilegiada na preferência dos usuários devido a grande maioria dos *softwares* livres não apresentarem uma plataforma amigável e nem manuais detalhados para a entrada de *scripting* no aplicativo [Paixão Junior e Tavares Junior, 2019]. Desenvolver um complemento para planejamento de voo em uma plataforma FOSS (*Free and Open Source Software*) integrando o *software QGIS* a linguagem *Python*, com uso do *PyQGIS* no *plugin Builder* [Sherman, 2016], foi uma alternativa proposta para facilitar a conexão entre o *scripting* de plano de voo e os *códigos* internos do *QGIS 3.4*.

O presente trabalho traz consigo a criação da ferramenta *QPlanner*, um módulo de planejamento de voo gratuito, para atuar como um complemento plataforma livre *QGIS*. Entre suas atribuições estão a otimização do trabalho do usuário em calcular e gerar as informações para o levantamento fotogramétrico, contribuindo para o aprimoramento do *QGIS*, aumentando suas funcionalidades e finalidades de integração SIG/Fotogrametria.

## 2. Metodologia

A fim de permitir a aquisição de imagens de forma automatizada e precisa para o processamento fotogramétrico, elementos como a trajetória de voo aeronave deve ser calculada com antecedência, utilizando-se para isso parâmetros do projeto, como o tipo de objeto a ser documentado, a sobreposição das imagens, o sensor da câmera, especificações da aeronave bem como sua autonomia de tempo, e restrições de voo (devido a requisitos de segurança), são parâmetros relevantes de um projeto fotogramétrico [Eisenbeiss e Sauerbier, 2011]. Para a elaboração do módulo, adotou-se os parâmetros propostos por Eisenbeiss (2018), Figueiredo e Figueiredo (2018) e Wolf et al. (2014) pertinentes ao planejamento de voo como: Altura de voo; Distância focal; Dimensões do pixel e do quadro da câmara; Superposição lateral e longitudinal; Velocidade da aeronave; Coordenadas da área de sobrevoo; e Projeção. Apresenta-se ao final um diretório de saída com os resultados calculados.

A interface operacional do módulo é editável, e foi elaborada na extensão *Qt Designer* do *QGIS* a partir de um arquivo de interface pré-definido no *plugin Builder* [Sherman, 2016]. Para a atualização da interface em cada modificação feita no *Qt Designer*, utilizou-se outro *plugin*, denominado de *Reloader* [Sherman, 2016]. Em seguida, inseriu-se as bibliotecas *PyQgis* necessárias e recomendadas pelo Sherman (2016), tais como: *PyQt.QtCore*, *PyQt.QtGui*, *PyQt.QtWidgets*, *utils* e a *core*. Além de uma biblioteca *python* denominada *numpy* ([https://qgis.org/pt\\_BR/site/](https://qgis.org/pt_BR/site/)). Dessa maneira o arquivo escrito em *python* na extensão do *Builder* contém todas as variáveis, formulações e comandos que o *QGIS* necessita para interpretar e gerar os produtos do levantamento fotogramétrico. Com essas bibliotecas é possível executar os processos para geração do planejamento de voo, seguindo a esquematização do fluxo de trabalho como mostra a Figura 1.

Com as bibliotecas inseridas, continuou-se com o desenvolvimento do *scripting*, em que, nessa etapa fez-se a adaptação das equações para determinar cada variável necessária para o plano de voo como mostra a Tabela 1.

**Tabela 1. Discriminação das variáveis e suas respectivas equações**

<i>Nº</i>	<i>Variáveis</i>	<i>Equações</i>
1	Denominador da escala	$den_{escala} = \frac{H_v}{f}$
2	GSD	$GSD = den_{escala} * d$
3	Escala da foto	$EF = \frac{f}{H_v}$
4	Lado longitudinal	$GL = den_{escala} * qX$
5	Lado transversal	$GT = den_{escala} * qY$
6	Aerobase	$B = GL * \left(1 - \frac{PE}{100}\right) * 100$
7	Avanço de base	$Av = \left(1 - \frac{PE}{100}\right)$
8	Distância entre faixas	$W = GL * \left(1 - \frac{PS}{100}\right)$

9	Distância longitudinal	$L = \sqrt{(Longitude\ 1 - Longitude\ 2)^2}$
10	Distância transversal	$Q = \sqrt{(Latitude\ 1 - Latitude\ 2)^2}$
11	Área total	$AT = L * Q$
12	Quantidade de modelos por faixa	$Nm = int\left(\frac{L}{B} + 1\right)$
13	Quantidade de fotos por faixa	$Nf = Nm + 1$
14	Quantidade de faixas por bloco	$Ns = int\left(\frac{Q}{W} + 1\right)$
15	Área incremental	$Anm = B * W$
16	Modelo estereoscópico	$Am = (GL - B) * GT$
17	Total de fotografias	$Nt = int\left(\frac{At}{Anm}\right)$
18	Total de fotografias capturadas	$Ntf = Nf * Ns$
19	Intervalo entre exposição	$I_e = \frac{B}{v}$

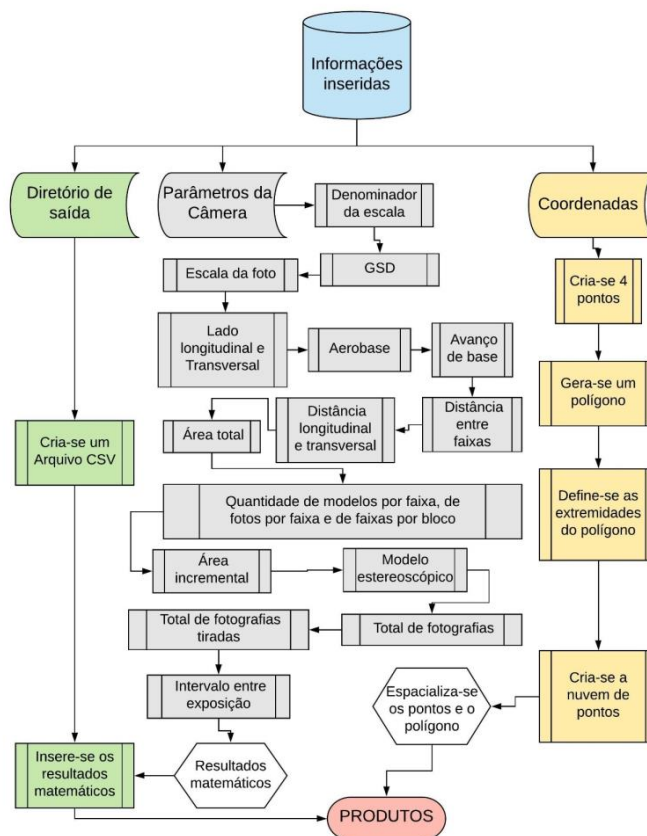


Figura 1. Fluxograma dos processos que módulo executa.

O passo seguinte foi a programação de um método que possibilita criar e espacializar uma geometria de forma automática. Primeiramente definiu-se a geometria da camada do polígono a ser gerada, a partir de um provedor para os pontos fornecidos pelo usuário, ou seja, o código reconheceu as coordenadas e cria mais duas, formando assim um polígono quadrilátero dentro do *QGIS*, correspondente à área que será sobrevoada. Dessa forma adiciona-se uma nova camada ao painel de *layers* do *QGIS*.

Para o desenvolvimento da nuvem de pontos, representando as coordenadas do centroide de cada fotografia, utilizou-se às equações: 14, correspondente a aerobase [Wolf et al., 2014] para representar o espaçamento dado em X e 16 para o espaçamento em Y, equivalente a distância entre faixas. Em seguida, no *scripting* inseriu-se uma variável para obter a extensão da camada da poligonal, e uma nova camada de pontos vetoriais. Com a nuvem de pontos construída, dentro dos limites da área que será sobrevoada, a próxima etapa, compõe a adição dessas coordenadas da camada de pontos e, assim, uma atualização da extensão da camada foi necessária para que ao final a camada ao painel do *QGIS* pudesse ser inserida. Do *scripting* para o diretório de saída dos arquivos calculados, o produto é gerado no formato *Comma-separated values* (.csv), e assim, seleciona-se os dados calculados e determina-se uma nomenclatura para cada variável, com o intuito de facilitar a inserção das informações no CSV.

### 3. Resultados Preliminares

A interface operacional criada a partir da extensão *Qt Designer*, possibilita ao usuário inserir os parâmetros da câmera, onde o tamanho do pixel em X e Y está em micrômetros, a altura do voo em metros, tamanho do quadro (x, y) em milímetros e a velocidade da aeronave está em metros por segundo. Outras informações inseridas corresponderam com as coordenadas do quadro que será sobrevoado, onde essas devem ser inseridas no formato UTM. Além de ser solicitado a projeção que os arquivos serão gerados e o diretório de saída, com isso, facilitando os cálculos que o *scripting* realizará, deste modo, a interface pode ser vista na Figura 2.

Figura 2. Interface operacional do módulo *QPlanner* para o *QGIS*.

O tempo de execução entre os cálculos realizados pelo módulo e manualmente foi de  $\pm 5$  segundos e  $\pm 20$  minutos, respectivamente, ou seja, a utilização do

complemento, além da praticidade, tornou os cálculos 240 vezes mais rápido, com o processo sendo executado pela equipe desse projeto nos testes preliminares.

#### 4. Considerações Finais

O presente trabalho conseguiu desenvolver um *scripting* para o planejamento de voo, dentro da plataforma do *software QGIS*, com uma interface de fácil manipulação e intuitiva, onde um usuário só precisa inserir as informações solicitadas. Dessa forma, seus produtos serão gerados de forma automática. Além disso, a instalação do módulo se torna bem simples, sendo baixado diretamente nos complementos do *software*.

Em suas funcionalidades o módulo consegue determinar as coordenadas do centro perspectivo de cada fotografia que será tomada durante o voo, e, com essas coordenadas gera-se uma nuvem de pontos para representar a localização de cada fotografia. Deste modo, fazendo com que o usuário consiga tomar decisões antes do voo, devido a geração de resultados matemáticos do plano de voo de forma rápida e precisa, além de poder com esses dados, gerar uma carta de auxílio para o voo.

Com isso, o módulo gerado, necessitará de adaptações e melhorias para facilitar o seu uso. Porém, nota-se que o objetivo do trabalho foi alcançado, gerado o plano de voo e exportando suas informações, o ambiente *QGIS*. Assim, essas informações geradas, são essenciais para a montagem de uma carta de voo, que em softwares comerciais são omitidas.

#### 5. References

- Amorim, A., Tommaselli, A. M. G. and Silva, I. (2006). Use of hybrid Stereopairs in Map Revision. In *Geodésia On-Line*, v. 10, p. 1–13. UFSC.
- Berveglieri, A. (2014). “Localização automática de pontos de controle em imagens aéreas baseada em cenas terrestres verticais”. Tese de Doutorado, UNESP Presidente Prudente, Programa de Pós-Graduação em Ciências Cartográficas, 148 f.
- Eisenbeiss, H. (2008). The autonomous mini helicopter: a powerful platform for mobile mapping. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, v. 37, p. 977–983. ISPRS.
- Eisenbeiss, H. e Sauerbier, M. (2011). Investigation of UAV Systems and Flight Modes for Photogrammetric Applications. *The Photogrammetric Record*, v. 26, n. 136, p. 400–421.
- Figueiredo, E. O. e Figueiredo, S. M. de M. (2018). Planos de voo semiautônomos para fotogrametria com aeronaves remotamente pilotadas de classe 3. *Circular Técnica*, v. 75, p. 1-56. Embrapa.
- Paixão Junior, J. G. C. e Tavares Junior, J. J. R. (2019) Perfilagem Multifonte de Borda com Mapeador de Texturas em *Python*: Baía de Icó-mandantes, Reservatório de Itaparica – PE. In: *Anais Do XIX Simpósio Brasileiro De Sensoriamento Remoto*, Instituto Nacional de Pesquisas Espaciais – INPE.
- Sherman, G. (2016). *The PyQGIS Programmer’s Guide*. Locate Press LLC, 1<sup>st</sup> edition.
- Wolf, P. R., Dewitt, B. A. e Wilkinson, B. E. (2014). *Elements of Photogrammetry with Applications in GIS*, McGraw-Hill, 4<sup>th</sup> edition.