

SISTEMA SEMI-AUTOMÁTICO DE CORREÇÃO ATMOSFÉRICA PARA MULTI-SENSORES ORBITAIS

Felipe Menino Carlos¹, Vitor de Souza Martins², Cláudio Clemente Faria Barbosa³

¹ Instituto Nacional de Pesquisas Espaciais (Bolsista no LabISA), felipe.carlos@fatec.sp.gov.br, ² Iowa State University, vitorstmartins@gmail.com, ³ Instituto Nacional de Pesquisas Espaciais, claudio.barbosa@inpe.br

RESUMO

A acurácia da correção atmosférica para o monitoramento de sistemas aquáticos é extremamente importante, por estes ambientes apresentarem reflectância espectral normalmente baixa. Alguns estudos vem mostrando que a aplicação do modelo 6SV de correção atmosférica, apresenta bons resultados para estes ambientes. Porém a usabilidade e automatização deste modelo são baixas, por necessitar de recursos não presentes nos metadados das imagens, além da baixa intuitividade na utilização. Com o objetivo de facilitar a aplicação deste modelo e de acelerar o procedimento de correção atmosférica em imagens dos sensores Sentinel-2/MSI, Landsat-8/OLI e Sentinel-3/OLCI desenvolveu-se um sistema na linguagem de programação Python, junto a alguns pacotes auxiliares. Os resultados obtidos através do programa foram validados por pesquisadores e alunos de pós-graduação do Laboratório de Instrumentação de Sistemas Aquáticos (<http://www.dpi.inpe.br/labisa/>).

Palavras-chave – Python, 6SV, Correção atmosférica.

ABSTRACT

The accuracy of the atmospheric correction for the monitoring of aquatic systems is extremely important because these environments have normally low spectral reflectance. Some studies have shown that the application of the 6SV model of atmospheric correction present good results for these environments. However, the usability and automation of this model are low, because it requires resources not present in the metadata of the images, besides the low intuitiveness in the use. In order to facilitate the application of this model and to accelerate the procedure of atmospheric correction in images of the Sentinel-2 / MSI, Landsat-8 / OLI, and Sentinel-3 / OLCI sensors, a system was developed in the Python programming language with some auxiliary packages. The results obtained through the program were validated by researchers and graduate students of the Laboratory of Instrumentation of Aquatic Systems (<http://www.dpi.inpe.br/labisa/>).

Key words – Python, 6SV, Atmospheric correction.

1. INTRODUÇÃO

O sensoriamento remoto de ambientes aquáticos vem se expandido com a disponibilidade de novos sensores orbitais [1], como *Operational Land Imager* (OLI) a bordo do Landsat 8, do *Multispectral Instrument* (MSI) a bordo SENTINEL-2A/B e o *Ocean and Land Colour Instrument* (OLCI) a bordo do Sentinel-3. Dentro as

diversas áreas, as aplicações em ambientes aquáticos são comumente focadas em estimativa de constituintes (sólidos em suspensão, pigmentos fotossintetizantes), ou na avaliação da dinâmica sazonal dos corpos d'água, como mapeamento e extensão de lagos. No entanto, os estudos quantitativos impõem um desafio no pré-processamento das imagens para obtenção de medidas radiométricas, como a reflectância de superfície. Esse processamento é conhecido como correção atmosférica, e visa remover os efeitos atenuantes causados pelos constituintes atmosféricos (aerossóis e gases) durante a propagação da radiação solar [2]. Como a água apresenta uma reflectância espectral tipicamente baixa, a acurácia nessa etapa se torna ainda mais relevante para aplicações como modelagem dos constituintes na água [3].

Os procedimentos de correção atmosférica variam entre modelos empíricos [4] ou físicos [5], e as suas limitações dependem das características atmosféricas, dados auxiliares e dos alvos em estudo. [6] realizou a avaliação de três modelos de correção atmosférica, em imagens Landsat-8/OLI da Amazônia, e conclui que, dentre os avaliados, os melhores resultados foram obtidos através do modelo 6SV (*Second Simulation of a Satellite Signal in the Solar Spectrum Vector version*). No entanto, mesmo diante dos bons resultados, a aplicabilidade deste modelo é limitada pela necessidade de extrair os parâmetros atmosféricos de outros sensores orbitais, como os produtos do *Moderate Resolution Imaging Spectroradiometer* (MODIS) level-2.

Diante dos desafios de um processamento contínuo desses dados orbitais para o monitoramento sistemático de ambientes aquáticos, um sistema único para a correção atmosférica de imagens de diferentes sensores orbitais se torna uma demanda de usuários e pesquisadores dessa área, como o Laboratório de Instrumentação de Sistemas (LabISA). Neste contexto, o objetivo deste trabalho foi desenvolver um sistema, denominado de AtmosPy, para automatizar as etapas da correção atmosférica. Esse sistema é baseado no estudo desenvolvido por [6], e possui aplicação para os sensores OLI, MSI e OLCI. Vale ressaltar também que o AtmosPy, é um sistema modular que permite adaptação da correção atmosférica para outros sensores ópticos.

2. MATERIAIS E MÉTODOS

2.1. Materiais

O desenvolvimento do AtmosPy, foi realizado em Python, uma linguagem de programação de código aberto e sem custo. Vale ressaltar que o Python é uma linguagem de uso geral para a qual vem sendo desenvolvido e disponibilizado, pela comunidade de usuários, pacotes para as mais diversas áreas, incluindo a científica [7]. Para as imagens dos sensores

MSI e OLI utilizou-se o EarthExplorer, serviço da *United States Geological Survey* (USGS), e para imagens do sensor OLCI, o *Open Access Hub* da *European Space Agency* (ESA) foi utilizado. Cada sensor disponibiliza as imagens em um formato específico, sendo para o MSI, OLI e OLCI os formatos, JPEG2000, TIF e netCDF respectivamente. Para manipular estes tipos de dados, diferentes pacotes foram utilizados na implementação do AtmosPy. Para imagens dos sensores MSI e OLI, o pacote *Geospatial Data Abstraction Library* (GDAL) foi utilizado, por este prover suporte à leitura, escrita e manipulação de arquivos *raster* em Python, o que o torna aplicável a diversas atividades de sensoriamento remoto, como demonstrado em [8]. Para os dados do OLCI, utilizou-se a biblioteca *xarray*, que já contém estruturas de dados que facilitam a abstração em código Python do formato netCDF.

Como o modelo de transferência radiativa 6SV [5], utilizado nas correções atmosféricas no AtmosPy, foi implementado na linguagem Fortran, o pacote Py6S [9], desenvolvido em Python, foi utilizado pelo AtmosPy para auxiliar a extração de parâmetros do modelo 6SV. O 6SV necessita de informações atmosféricas como carga óptica dos aerossóis, ozônio e vapor d'água, para determinar os coeficientes de correção. Como estas informações não estão disponíveis no metadado das imagens, elas foram obtidas a partir de produtos de outros sensores, como os disponibilizados pelo MODIS. Esta etapa de obtenção das informações atmosféricas foi automatizada utilizando o código disponibilizado por Murphy [10], que por sua vez utiliza o Google Earth Engine (GEE), uma plataforma global de análise de grandes conjuntos de dados geoespaciais [11].

Para a interação com o usuário, o AtmosPy utiliza o pacote de desenvolvimento de interfaces gráficas, PyQt, este que utiliza a biblioteca Qt. Junto a este pacote, a ferramenta de criação de mapas interativos, OpenLayers, foi utilizada para facilitar a busca de geolocalizações, necessárias na pesquisa de características atmosféricas. Em casos onde há diversas imagens a serem corrigidas, o que pode demandar muito tempo computacional, pode-se utilizar o processador gráfico do computador (GPU - *Graphical Processing Unit*) para reduzir o tempo de processamento, conforme demonstrado por [12]. No AtmosPy, utilizou-se o pacote Numba, que fornece formas de comunicação com hardwares que possuem tecnologia CUDA, para acelerar o processamento das imagens através de GPU. A arquitetura *Compute Unified Device Architecture* (CUDA), criada pela NVIDIA, vem sendo aplicada em diversos trabalhos de sensoriamento remoto como em [13, 14], principalmente pelo aumento na velocidade dos processos.

2.2. Métodos

A metodologia utilizada para o desenvolvimento do AtmosPy foi dividida nos seguintes passos: (i) Modelagem, (ii) Implementação, (iii) Validação e correções.

Na etapa de modelagem, o objetivo principal foi identificar e desenvolver formas de facilitar e agilizar as etapas de correção atmosférica descritas e desenvolvidas por [6]. Para isto, fez-se uma avaliação de etapas passíveis de automatização e de onde se poderia melhorar os

processamentos. Desta forma identificou-se as seguintes necessidades:

- Automatização do processo de busca de parâmetros atmosféricos;
- Aumento da velocidade na aplicação da correção atmosférica;
- Forma única para a aplicação da correção, independente do sensor.

Após esta identificação, a especificação de funcionamento do sistema (Figura 1) foi realizada, para permitir que, a etapa de implementação fosse direta a resolução dos problemas encontrados.

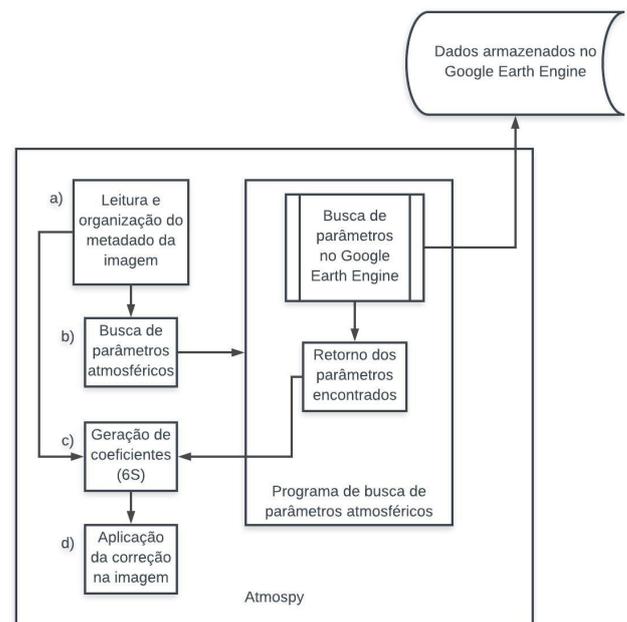


Figura 1: Funcionamento do sistema

Como apresentado na Figura 1, no módulo (a), o usuário define o local em disco onde a imagem está e a sua geolocalização. A partir destas informações o AtmosPy extrai do metadado da imagem a data de imageamento e os ângulos solar e do sensor e realiza a busca automatizada dos parâmetros atmosféricos pelo módulo (b). Após a aquisição das informações atmosféricas o modelo 6SV (módulo c) determina os parâmetros de correção a serem aplicados na imagem. Na última etapa, o módulo (d), que é parte do 6SV, utilizando os parâmetros gerados no módulo (c), aplica a correção em cada uma das bandas espectrais da imagem, e salva em disco as imagens corrigidas.

Os testes e validações foram realizados em conjunto com pesquisadores e alunos de pós-graduação do LabISA. Como resultados destes testes de utilização, funcionalidades extras foram adicionadas ao AtmosPy, como a busca de geolocalização em um mapa interativo.

3. RESULTADOS

A Figura 2 mostra a interface de usuário do AtmosPy, que é dividida em duas abas: Coeficiente e Correção.

A aba Coeficiente é onde o usuário fornece ao AtmosPy as informações iniciais para a busca automática dos dados atmosféricos que serão utilizados pelo 6SV para determinação dos parâmetros de correção das imagens, conforme descrito na seção 2.2 (métodos). Se o usuário eventualmente tiver os parâmetros de correção determinados por outro sistema ou em uma sessão anterior, estes parâmetros podem ser utilizados sem realizar a busca automática. Uma das informações de entrada é a localização da cena que se vai corrigir. Para facilitar a entrada desta informação, o AtmosPy possibilita a abertura de um mapa interativo, conforme o mostrado na Figura 3. A segunda aba, mostrada na parte inferior da Figura 2, é onde efetivamente se aplica a correção atmosférica. Caso o usuário esteja fazendo a correção na mesma sessão em que realizou a busca automática dos dados atmosféricos, as informações iniciais fornecidas na aba Coeficientes poderão ser usada nesta interface. É também na aba Correção que o usuário pode escolher o uso da GPU para otimizar o tempo de processamento.



Figura 2: Tela principal do AtmosPy

Há ainda disponível através do menu Início, uma opção para acessar a toda documentação do sistema, desde formas de uso, até os diagramas de especificação.

4. DISCUSSÃO

Todo o esforço no desenvolvimento do AtmosPy foi voltado no sentido de facilitar o processo de correção atmosférica utilizando o modelo 6SV. Entretanto, vale ressaltar, que para uso do AtmosPy, o usuário precisa instalar o pacote 6SV. Esta instalação, que pode apresentar alguma dificuldade para o usuário, não foi tratada neste trabalho, entretanto, junto a documentação do AtmosPy, encontra-se os passos a serem seguidos para esta instalação.

Há também o fato que, ao utilizar a GPU, o processo de correção torna-se mais rápido, e consome muito menos da memória principal da máquina onde está sendo executado,

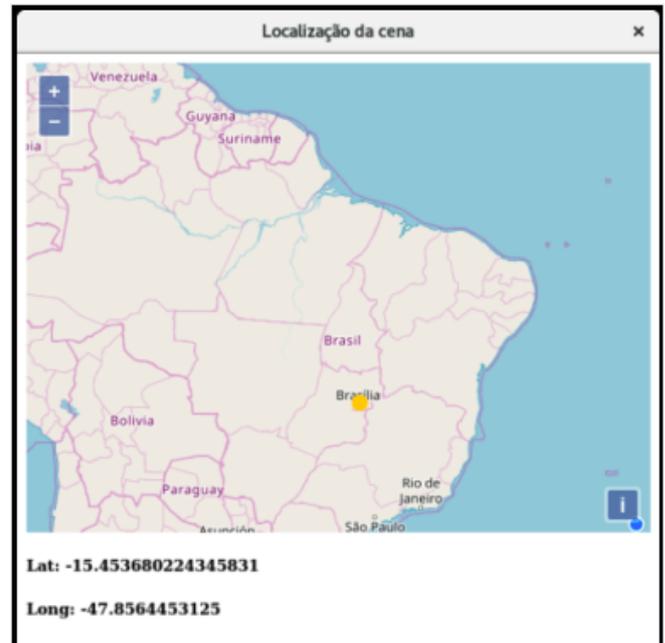


Figura 3: Mapa interativo do AtmosPy

conforme ilustrado na Figura 4, com o exemplo de uma correção sendo aplicada a imagens do sensor MSI.

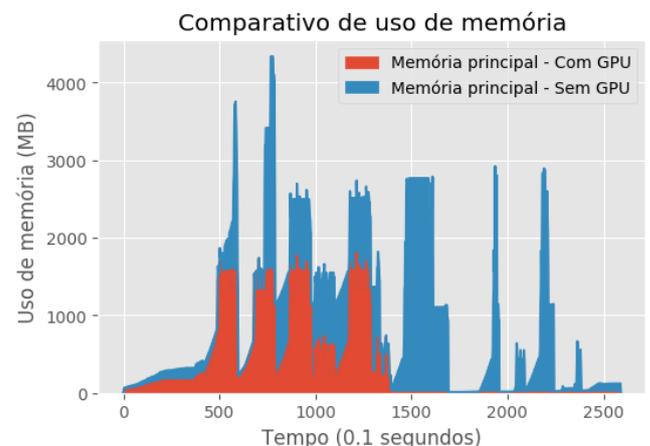


Figura 4: Uso de memória principal: CPU X GPU

Além disto, a interface de usuário (Figura 2), não requer nenhum passo especial para nenhum dos sensores corrigidos pelo AtmosPy, sendo necessário apenas a alteração da opção do sensor a ser corrigido, disponível também na janela principal.

5. CONCLUSÕES

O desenvolvimento do sistema AtmosPy, complementado por sugestões e testes feitos por diferentes usuários permitiu obter uma melhora sensível tanto na facilidade de uso quanto no tempo de processamento de correções atmosféricas utilizando o modelo 6SV. O AtmosPy também permitiu uma agilização no processo de correção de uma sequencia temporal de imagens.

6. REFERÊNCIAS

- [1] PALMER, S. C. J.; KUTSER, T.; HUNTER, P. D. Remote sensing of inland waters: Challenges, progress and future directions. *Remote Sensing of Environment*, v. 157, p. 1–8, 2015. ISSN 0034-4257.
- [2] KAUFMAN, Y. J. et al. Operational remote sensing of tropospheric aerosol over land from EOS moderate resolution imaging spectroradiometer. *Journal of Geophysical Research: Atmospheres*, v. 102, n. D14, p. 17051–17067.
- [3] VANHELLEMONT, Q.; RUDDICK, K. Advantages of high quality SWIR bands for ocean colour processing: Examples from Landsat-8. *Remote Sensing of Environment*, v. 161, p. 89–106, 2015. ISSN 0034-4257.
- [4] CHAVEZ, P. S. An improved dark-object subtraction technique for atmospheric scattering correction of multispectral data. *Remote Sensing of Environment*, v. 24, n. 3, p. 459–479, 1988. ISSN 0034-4257.
- [5] VERMOTE, E. F. et al. Second simulation of the satellite signal in the solar spectrum, 6s: an overview. *IEEE Transactions on Geoscience and Remote Sensing*, v. 35, n. 3, p. 675–686, May 1997. ISSN 0196-2892.
- [6] Martins, Vitor Souza, DE CARVALHO, L.A.S, BARBOSA, CLAUDIO, NOVO, E. Avaliação da acurácia em produtos OLI/Landsat 8 em lagos amazônicos: Correção Atmosférica. *SBSR*, v. 18, 2017.
- [7] HAENEL, V.; GOUILLART, E.; VAROQUAUX, G. Python Scientific lecture notes. *Languages*, v. 2013, n. euroscipy 2013, p. 1–139, 2010.
- [8] APPEL, M. et al. Open and scalable analytics of large Earth observation datasets: From scenes to multidimensional arrays using SciDB and GDAL. *ISPRS Journal of Photogrammetry and Remote Sensing*, v. 138, p. 47–56, 2018. ISSN 0924-2716.
- [9] WILSON, R. T. Py6S: A Python interface to the 6S radiative transfer model. *Computers and Geosciences*, v. 51, p. 166–171, 2013. ISSN 00983004.
- [10] MURPHY, S. *Atmospheric correction of Sentinel 2 imagery in Google Earth Engine using Py6S*. [S.l.], 2016. Disponível em: <<https://github.com/samsammurphy/gee-atmcorr-S2>>.
- [11] GORELICK, N. et al. Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, Elsevier, 2017.
- [12] SANTOS, B. P.; DOMINGUEZ, D. S.; ORELLANA, E. V. Aplicando Processamento Paralelo com GPU ao Problema do Fractal de Mandelbrot. p. 1–10, 2011.
- [13] LOPEZ-FANDINO, J. et al. CUDA multiclass change detection for remote sensing hyperspectral images using extended morphological profiles. *Proceedings of the 2017 IEEE 9th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2017*, v. 1, n. September, 2017.
- [14] SI, X.; ZHENG, H. High performance remote sensing image processing using cuda. In: *2010 Third International Symposium on Electronic Commerce and Security*. [S.l.: s.n.], 2010. p. 121–125.