ALGORITMO PARA OTIMIZAÇÃO DE ROTA TURÍSTICA EM *INSELBERGS* UTILIZANDO MODELOS DIGITAIS DE ELEVAÇÃO

Gabriel de Lima Ferreira [1], *Maira Rodrigues Villamagna* [2], *Denize Monteiro dos Anjos* [3], *Marcos Cordeiro Mamede Filho*[4], e *Iuri Kauã Simão de Oliveira*[5]

[1]IFPB, gabriel.ferreira.1@academico.ifpb.edu.br; [2]IFPB, maira.villamagna@ifpb.edu.br; [3]IFPB, denize.anjos@ifpb.edu.br; [4]IFPB, marcos.mamede@academico.ifpb.edu.br; e [5]IFPB, oliveira.iuri@academico.ifpb.edu.br

RESUMO

Este artigo descreve o desenvolvimento de um algoritmo para otimizar a exploração turística de *inselbergs* na zona sul de Patos-PB, utilizando modelos digitais de elevação (*DEM*) e técnicas de interpolação matemática. O sistema faz uso das bibliotecas *Python scipy.spatial, matplotlib* e *numpy* para gerar superfícies tridimensionais a partir de coordenadas geográficas, permitindo a visualização precisa do relevo local. A partir desses dados, o algoritmo identifica rotas com menor inclinação, facilitando trajetos mais seguros e acessíveis para turistas. O método envolve a triangulação de *Delaunay* para a criação de uma malha de pontos e superfícies, além de possibilitar a visualização interativa da topografia. O artigo discute a implementação detalhada do código, a eficiência dos resultados e as implicações para o turismo ecológico, com base em dados reais da região. A abordagem propõe uma solução robusta para o planejamento de rotas turísticas, otimizando o equilíbrio entre segurança e esforço físico.

Palavras-chave **—** interpolação, *inselberg*, triangulação de *Delaunay*, *Python*, turismo.

ABSTRACT

This paper describes the development of an algorithm to optimize tourist exploration of inselbergs in the southern region of Patos-PB, using digital elevation models (DEM) and mathematical interpolation techniques. The system leverages Python libraries such as scipy.spatial, matplotlib, and numpy to generate three-dimensional surfaces from geographic coordinates, allowing precise visualization of the local terrain. Based on these data, the algorithm identifies routes with the least slope, enabling safer and more accessible paths for tourists. The method involves Delaunay triangulation to create a mesh of points and surfaces, as well as providing an interactive visualization of the topography. The paper discusses the detailed implementation of the code, the efficiency of the results, and the implications for ecological tourism, based on real data from the region. This approach offers a robust solution for tourist route planning, optimizing the balance between safety and physical effort.

Key words **—** *interpolation, inselberg, Delaunay triangulation, Python, tourism*

1. INTRODUÇÃO

Com a homologação do Parque Nacional da Serra do Teixeira, na Paraíba, em junho de 2023, surgiu uma nova oportunidade para o desenvolvimento do turismo ecológico na região, reconhecida por sua rica biodiversidade e paisagens deslumbrantes. Dentro desse cenário, a cidade de Patos, localizada na mesorregião do sertão paraibano, destaca-se com seus *inselbergs*, formações geológicas isoladas que compõem a paisagem. Esses *inselbergs*, que significam "montanhas-ilhas", são blocos de rocha que resistiram à erosão ao longo do tempo, criando elevações solitárias e peculiares, abrigando flora e fauna distintas em comparação às áreas circundantes[1].

Essas formações possuem grande valor turístico e ecológico, mas devido à inclinação pronunciada de seu relevo, apresentam desafios para exploração segura e acessível. Dentre os *inselbergs* presentes na região de Patos, o Espinho Branco se destaca como uma formação proeminente. Situado no município, o Espinho Branco é uma elevação que atrai interesse tanto pela sua beleza natural quanto pelas oportunidades de atividades ao ar livre, como trilhas e escaladas. No entanto, a navegação por essa área exige rotas planejadas que considerem o relevo irregular e a segurança dos visitantes.

Este trabalho apresenta o desenvolvimento de um algoritmo que utiliza bibliotecas *Python*, como *scipy.spatial* e *matplotlib*, para gerar rotas turísticas otimizadas baseadas na menor inclinação entre as curvas de nível. A triangulação de *Delaunay* e o diagrama de *Voronoi* são aplicados para calcular o gradiente do terreno, destacando rotas que garantem acessibilidade e segurança aos turistas. O estudo inclui o Espinho Branco como área de análise, demonstrando o potencial do projeto em utilizar tecnologia avançada para promover o turismo sustentável e responsável na região de Patos.

Além de otimizar a exploração turística, o projeto contribui para a preservação ambiental, reduzindo o impacto nas áreas sensíveis, ao mesmo tempo que oferece uma experiência enriquecedora aos visitantes. A implementação do algoritmo equilibra o desenvolvimento econômico com a conservação dos recursos naturais e culturais da Paraíba.

2. MATERIAL E MÉTODOS

Este estudo desenvolveu e implementou um algoritmo em *Python* para calcular o gradiente com base nas curvas de nível dos *inselbergs* na região sul de Patos-PB, com o objetivo de traçar a melhor rota de visitação turística. A metodologia incluiu a coleta, processamento e análise de dados topográficos, além do desenvolvimento e validação do algoritmo proposto. A seguir, são detalhadas as etapas e os materiais utilizados.

Os dados topográficos da região de Patos-PB foram obtidos no site da *United States Geological Survey* (USGS), através da opção *Digital Elevation*. Especificamente, foi utilizado o modelo *SRTM 1 ARC-Second Global*, que fornece dados de elevação com resolução de aproximadamente 30 metros. Esse modelo foi selecionado devido à sua abrangência global e precisão adequada para a análise de relevo irregular, como os *inselbergs*. A Figura 1 ilustra o processo de aquisição dos dados topográficos.

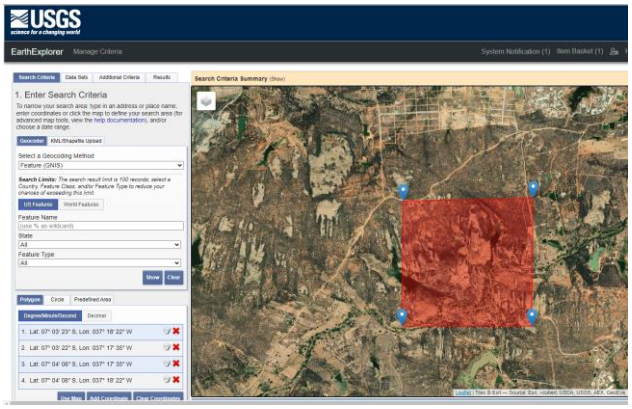


Figura 1. Aquisição dos dados topográficos via USGS

A partir dos modelos digitais de elevação (DEM), foram geradas curvas de nível representando diferentes altitudes da área estudada, conforme mostrado na Figura 2

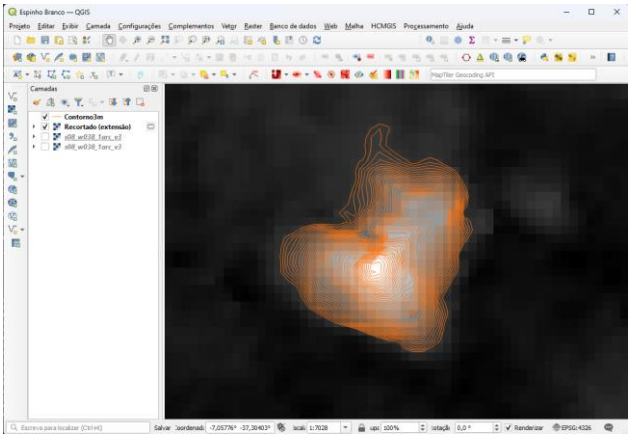
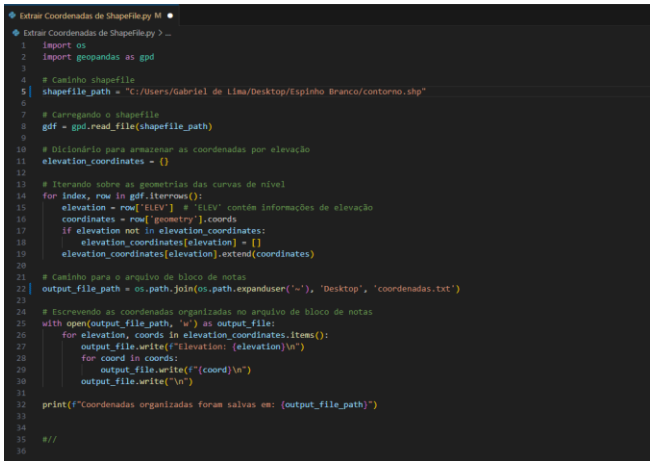


Figura 2. Obtenção das curvas de nível a partir de recorte por camada vetorial e aplicação de contornos no software QGIS.

Após obter os dados de elevação, o software QGIS foi utilizado para processamento e análise geoespacial. Um arquivo KML delimitou a área de interesse, focando nos inselbergs da região de Patos-PB, especialmente no Espinho Branco. As curvas de nível foram exportadas no formato *Shapefile*, um formato amplamente utilizado em sistemas de informação geográfica (SIG) para armazenar a localização geográfica e os atributos das entidades espaciais [2].

Para organizar as curvas de nível e facilitar análises subsequentes, desenvolveu-se um algoritmo em *Python*. O código demonstrou o processo de extração de elevações e coordenadas geográficas do *Shapefile*, salvando-as em um arquivo de texto estruturado, conforme ilustrado na Figura 3.



```python
import os
import geopandas as gpd

# Caminho shapefile
shapefile_path = "C:/Users/Gabriel de Lima/Desktop/Espinho Branco/contorno.shp"

# Carregando o shapefile
gdf = gpd.read_file(shapefile_path)

# Dicionário para armazenar as coordenadas por elevação
elevation_coordinates = {}

# Iterando sobre as geometrias das curvas de nível
for index, row in gdf.iterrows():
    elevation = row['ELEV']  # 'ELEV' contém informações de elevação
    coordinates = row['geometry'].coords
    if elevation not in elevation_coordinates:
        elevation_coordinates[elevation] = []
    elevation_coordinates[elevation].extend(coordinates)

# Caminho para o arquivo de bloco de notas
output_file_path = os.path.join(os.path.expanduser('~'), 'Desktop', 'coordenadas.txt')

# Escrevendo as coordenadas organizadas no arquivo de bloco de notas
with open(output_file_path, 'w') as output_file:
    for elevation, coords in elevation_coordinates.items():
        output_file.write(f"Elevation: {elevation}\n")
        for coord in coords:
            output_file.write(f"{coord}\n")
        output_file.write("\n")

print(f"Coordenadas organizadas foram salvas em: {output_file_path}")
```

Figura 3. Algoritmo desenvolvido para extrair os conjuntos de coordenadas e elevações de arquivo Shapefile.

O núcleo do estudo foi o desenvolvimento de um algoritmo em *Python* que integra técnicas de geoprocessamento e modelos matemáticos para calcular rotas turísticas otimizadas. As bibliotecas utilizadas incluem:
scipy.spatial: Utilizada para realizar a triangulação de *Delaunay*, essencial para a criação de uma malha triangular que representa a superfície do terreno.
matplotlib: Responsável pela geração de visualizações tridimensionais (3D) das curvas de nível e das rotas calculadas.
numpy: Empregada para manipulação eficiente de arrays de coordenadas geográficas e para cálculos matemáticos necessários na determinação do gradiente.
mpl_toolkits.mplot3d: Complementa o *matplotlib* na criação de gráficos 3D detalhados.

Aplicada aos pontos geográficos para gerar uma malha triangular que modela a topografia do terreno, este método garante que a triangulação seja otimizada, evitando a sobreposição de triângulos e mantendo a precisão na representação do relevo [3].
Realizar a determinação da inclinação do terreno entre as curvas de nível, identificando as rotas com menor gradiente é crucial para garantir que as rotas propostas sejam acessíveis

e seguras para os turistas. A Figura 4 apresenta a aplicação dessas técnicas por meio do código em *Python*.

```python
import os
import matplotlib.pyplot as plt
import numpy as np
from scipy.spatial import Voronoi, cKDTree
from scipy.interpolate import griddata
from mpl_toolkits.mplot3d.art3d import Poly3DCollection
from pykml import parser
from lxml import etree

# Função para importar e processar o KML usando pykml
def import_kml(kml_file_path):
    with open(kml_file_path, 'rt', encoding='utf-8') as file:
        doc = parser.parse(file)
        root = doc.getroot()

    ns = {'kml': 'http://www.opengis.net/kml/2.2'}
    coordinates = []

    # Função auxiliar para explorar recursivamente os elementos KML
    def extract_coords_from_element(element):
        # Verificar se o namespace está correto e se os elementos estão presentes
        placemarks = element.findall('.//kml:Placemark', namespaces=ns)
        if not placemarks:
            raise ValueError("Nenhum 'Placemark' encontrado no KML. Verifique a estrutura do arquivo KML.")

        for placemark in placemarks:
            line_string = placemark.find('.//kml:LineString', namespaces=ns)
            if line_string is not None:
                coords_text = line_string.find('.//kml:coordinates', namespaces=ns).text.strip()
                for coord in coords_text.split():
                    lon, lat, _ = map(float, coord.split(','))
                    coordinates.append((lon, lat))

    extract_coords_from_element(root)
    return np.array(coordinates)

# Função para encontrar a elevação mais próxima para cada ponto KML
def find_nearest_elevations(kml_coordinates, geo_coordinates):
```

Figura 4. Algoritmo final com as funções que descritas.

Após a geração das rotas otimizadas, o algoritmo também realiza cálculos de área das regiões a serem exploradas durante as visitas turísticas. Estes cálculos são realizados por meio de integrais duplas, utilizando conceitos de cálculo integral e geoprocessamento, o que permite uma análise abrangente e precisa das áreas de interesse [4].

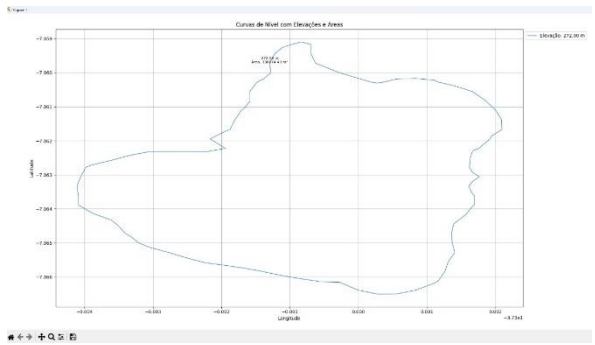


Figura 5. Plotagem da área calculada das curvas de nível

A eficácia do algoritmo foi validada por meio de dados reais da região do Espinho Branco. As rotas calculadas foram comparadas com trajetos convencionais afim de verificar a melhora na acessibilidade e segurança proporcionadas pelo algoritmo. Os resultados demonstram que as rotas otimizadas apresentam menor inclinação e maior facilidade de navegação, confirmando a eficiência do método proposto [5].

Entre as ferramentas e tecnologias utilizadas elencamos como principais: QGIS: Utilizado para o processamento inicial dos dados geoespaciais, geração das curvas de nível e análise espacial das áreas de interesse. *Python*: Linguagem de programação escolhida pela sua flexibilidade e pelas bibliotecas especializadas em geoprocessamento e visualização de dados.

3. RESULTADOS

O algoritmo desenvolvido conseguiu gerar rotas turísticas otimizadas com base nas curvas de nível dos *inselbergs* da região de Patos-PB, especificamente no Espinho Branco. As imagens geradas mostram tanto os dados topográficos quanto as curvas de nível detalhadas do relevo. A visualização tridimensional das superfícies permite observar o trajeto planejado em função da inclinação do terreno, conforme as Figuras 6 e 7 abaixo.

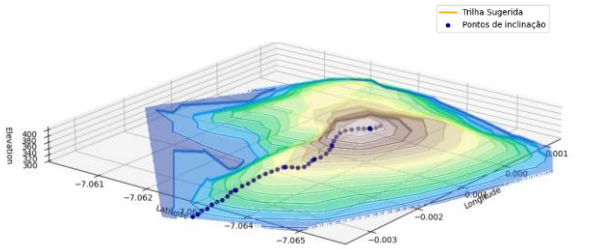


Figura 6. Plotagem dos dados extraídos e cálculo da melhor rota com menor inclinação.

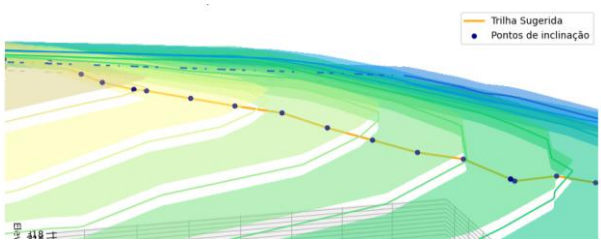


Figura 7. Visualização lateral do caminho criado pelo algoritmo.

A partir dos dados processados no QGIS, o algoritmo produziu mapas com curvas de nível e superfícies preenchidas, que indicam as elevações e o caminho sugerido para a exploração do inselberg. A rota gerada é claramente visível nas representações gráficas, onde foram identificados os pontos com menor inclinação, facilitando a navegação pelo terreno. A triangulação de *Delaunay* aplicada mostra a divisão do terreno em triângulos para garantir que o trajeto siga o relevo de maneira otimizada. Além disso, a visualização tridimensional permite comparar o trajeto com o relevo real, destacando áreas de maior elevação e trechos mais suaves para os turistas.

Os resultados indicam que o algoritmo é capaz de não apenas sugerir rotas com menor inclinação, mas também oferecer uma visão detalhada da topografia local, ajudando na análise da acessibilidade e segurança dos trajetos.

4. DISCUSSÃO

Os resultados obtidos demonstram que a aplicação de técnicas de triangulação de *Delaunay* e interpolação de curvas de nível, associadas a modelos digitais de elevação (DEM), são eficazes para a otimização de rotas em terrenos acidentados, como os *inselbergs*. A região do Espinho

Branco, por exemplo, se beneficiou da análise detalhada do relevo, permitindo a criação de rotas acessíveis e seguras, sem comprometer a integridade ecológica do local.

A precisão dos dados topográficos obtidos do USGS e o processamento realizado no QGIS foram fundamentais para garantir a qualidade das curvas de nível e a exatidão das rotas traçadas. Ao utilizar um algoritmo capaz de calcular o gradiente do terreno, o software identificou os caminhos com menor inclinação, proporcionando uma experiência mais confortável para os turistas, especialmente em um ambiente de relevo tão acidentado como o dos *inselbergs*.

Um aspecto interessante do projeto é a integração visual proporcionada pelas visualizações tridimensionais geradas pelo algoritmo. A combinação das superfícies preenchidas com as linhas de contorno das curvas de nível e a triangulação do terreno oferece uma percepção clara do trajeto em relação ao relevo, o que facilita a tomada de decisões para exploração turística. Isso corrobora com a ideia de que o turismo sustentável em regiões montanhosas pode ser otimizado por ferramentas tecnológicas de geoprocessamento [6].

No entanto, para áreas com elevações mais abruptas ou de difícil acesso, pode ser interessante considerar a implementação de técnicas mais sofisticadas de interpolação, como o uso de redes triangulares irregulares (TIN), que podem refinar ainda mais as rotas em áreas com variações de altitude mais significativas.

5. CONCLUSÕES

O algoritmo desenvolvido mostrou-se eficaz na otimização de rotas turísticas em regiões de relevo acidentado, como o Espinho Branco, na zona sul de Patos-PB. Ao utilizar dados topográficos detalhados e processados através de ferramentas como o QGIS e bibliotecas *Python* especializadas, o sistema foi capaz de gerar trajetos que levam em consideração a menor inclinação, otimizando a acessibilidade e segurança dos visitantes.

A triangulação de *Delaunay*, aliada à interpolação das curvas de nível, forneceu uma base robusta para a visualização tridimensional do relevo, permitindo a análise clara das melhores rotas para a exploração turística da região. Os resultados indicam que o uso de tecnologias de geoprocessamento, associado a um planejamento eficiente, pode contribuir para o desenvolvimento do turismo sustentável, oferecendo ao mesmo tempo uma experiência positiva para os visitantes e promovendo a preservação ambiental.

Futuramente, melhorias no algoritmo podem incluir a implementação de rotas alternativas baseadas em diferentes critérios, como o nível de dificuldade ou a preferência por pontos de interesse específicos. Além disso, a expansão para outras áreas turísticas de potencial geográfico e cultural pode ampliar ainda mais o impacto do projeto para a região de Patos-PB.

8. REFERÊNCIAS

[1] C. Cunha, F. França, E. Melo, and J. M. Goncalves. Complexo de sítios de pinturas rupestres da Pedra Grande na região dos *inselbergs* de Itatim, Bahia, Brasil: Estado atual e perspectivas de preservação de uma área arqueológica em alto risco de degradação, In. XV World Congress of Union Internationale des Sciences préhistoriques et protohistoriques, v. 47, pp. 101-107, Lisboa, 2006.

[2] I. J. De Oliveira. A cartografia aplicada ao planejamento do turismo, Boletim Goiano de Geografia, v. 25, n. 1, pp. 29-46, 2005.

[3] A. Sangreman, G. Freitas, and R. R. Costa. Modelagem de terrenos naturais através de malhas de triângulos, Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 1, n. 1, 2013.

[4] J. Stewart. Cálculo. 8th ed. São Paulo: Cengage Learning, 2016. 1408 p.

[5] M. Marques, M. C. De Paula, and M. H. Campos. Análise de variáveis espaço-temporais durante a locomoção em esteira, com diferentes gradientes, Educação Física em Revista, v. 6, n. 3, 2012.

[6] F. Y. Hanai. Desenvolvimento sustentável e sustentabilidade do turismo: conceitos, reflexões e perspectivas, Revista Brasileira de Gestão e Desenvolvimento Regional, v. 8, n. 1, 2012.