**Methodology and Procedures Document**

**Title: Boulder Detection and Attribute Extraction**

**Script Overview**

The provided Python script is designed to perform the following tasks:

1. **Objective**: The script's main objective is to detect and extract attributes of boulders within a given geographic area represented by polygon layers and associate them with their corresponding depths from a bathymetric raster dataset.
2. **Inputs**:
   * Input Polygon Layer: A shapefile containing polygon features representing boulders' locations.
   * Input Raster Dataset: A bathymetric raster dataset representing the depth of the water bodies.
   * Block Name: A user-defined identifier for the geographic block (e.g., 'B02').
3. **Outputs**:
   * Target List Shapefile: A shapefile containing point features representing detected boulders along with their attributes.

**Script Workflow**

The script follows these main steps to achieve its objective:

1. **Import Required Libraries**: The script begins by importing the necessary Python libraries:
   * os: For handling file operations.
   * geopandas: For working with geospatial data and shapefiles.
   * rasterio: For reading and processing raster datasets.
   * pandas: For data manipulation.
   * shapely.geometry.Point: For working with geometric points.
2. **Define Input and Output Paths**: The script defines file paths for the input polygon layer, input raster dataset, and the output target list shapefile. Additionally, it sets a user-defined block name.
3. **Read Input Data**: The script reads the input polygon layer (representing boulders) into a GeoDataFrame.
4. **Open the Bathymetric Raster**: It opens the bathymetric raster using the rasterio library and proceeds to extract attributes for each boulder polygon within the loop.
5. **Attribute Extraction Loop**: For each polygon in the input polygon layer, the script performs the following operations:
   * Calculates the centroid of the polygon.
   * Retrieves the coordinates (Easting, Northing) of the centroid.
   * Samples the bathymetric raster to obtain the depth at the centroid's location.
   * Calculates the length, width, and height (absolute depth) of the boulder.
   * Generates a unique identifier (Target ID) for each boulder.
   * Creates a dictionary containing boulder attributes.
6. **Create Target GeoDataFrame**: The script creates a GeoDataFrame from the extracted attributes and creates point geometries from the Easting and Northing coordinates.
7. **Write Output Shapefile**: If the output shapefile does not exist, the script writes the target GeoDataFrame to the output shapefile. If the output file already exists, it appends the new data to the existing shapefile.
8. **Completion Message**: A message indicating the successful completion of the task is printed.

**Testing and Usage Instructions**

**Prerequisites**

Before running the script, ensure that you have the following prerequisites installed:

* Python 3.x
* Required Python libraries (geopandas, rasterio, pandas)

**Usage**

1. **Prepare Input Data**:
   * Ensure that you have the input polygon layer (shapefile) and the bathymetric raster dataset ready.
   * Modify the input\_polygon\_layer\_path, input\_raster\_path, and block\_name variables in the script to specify the correct file paths and block name.
2. **Run the Script**:
   * Execute the script in a Python environment. You can use a Python IDE.
3. **Review Output**:
   * After the script completes successfully, you will find the output target list shapefile (target\_list.shp) in the specified output directory.
4. **Additional Operations**:
   * You can customize the script further to include additional attributes or modify the attribute names according to your requirements.
5. **Error Handling**:
   * Ensure that the input data files exist at the specified paths, and the required libraries are correctly installed to avoid any errors.
6. **Integration**:
   * You can integrate this script into a larger geospatial analysis workflow or data processing pipeline as needed.

This document provides an overview of the methodology, procedures, and usage instructions for the provided Python script. Follow the instructions to utilize the script for boulder detection and attribute extraction in your geospatial data analysis tasks.