Title: PyVF: A Python Program for Extracting Vertical Features from LiDAR-DEMs

Author: Shu Gao[1], Matthew V. Bilskie[2], Scott C. Hagen[1,3,4,5, †]

[1] Department of Civil and Environmental Engineering, Louisiana State University, Baton Rouge, LA 70803, USA

[2] School of Environmental, Civil, Agricultural, and Mechanical Engineering, University of Georgia, Athens, GA, 30602, USA

[3] Center for Coastal Resiliency, Louisiana State University, Baton Rouge, LA 70803, USA

[4] Center for Computation and Technology, Louisiana State University, Baton Rouge, LA 70803, USA

[5] Coastal Studies Institute, Louisiana State University, Baton Rouge, LA 70803, USA

[†] Deceased

Corresponding Author: Shu Gao

Email: sgao7@lsu.edu

Address: 1110 River Rd #100, Baton Rouge, LA 70802

Authorship

Shu Gao developed the algorithm, wrote code, and prepared the initial draft of the manuscript.

Matthew V. Bilskie aided conceptual algorithm development, wrote code, and revised the manuscript.

Scott C. Hagen (PI) conceived the initial idea, aided conceptual algorithm development, and revised the manuscript.

Abstract

Coastal and riverine flooding is one of the most common environmental hazards that affect billions of people worldwide. A coupled hydrologic and coastal storm surge simulation is required to develop an improved understanding of the individual and collective mechanisms that can cause flooding within watersheds. These simulations are dependent on an accurate digital elevation model (DEM); however, it is a challenge to include numerical model resolution as fine as contemporary DEMs due to the enormous computational cost. Therefore, significant vertical features (VFs) such as roadbeds, levees, railroads, and natural ridges must be identified and considered in developing the model representation of the DEM since the VFs can affect flow propagation. PyVF is an open-source program to extract significant VFs from a high-resolution, bare-earth, LiDAR-derived DEM automatically. This paper introduces the methods and shows the automated extraction of VFs for a coastal, urban, mountain and beach area.

Keywords: Vertical Features; Automated Extraction; Digital Elevation Model; Surface Hydrology; Compound Flooding; Storm Surge

Computer Code Availability

Software name: PyVF (Version 1.2)

Availability: All Python code and testing data for PyVF associated with the current submission is available through https://github.com/ShuGao7/PyVF.git or https://doi.org/10.5281/zenodo.4291027 under the GNU General Public License v3.0. Any updates will also be published on Github and Zenodo.

1. Introduction

Flooding in coastal regions can be caused by (i) riverine flooding from extreme rainfall runoff; (ii) coastal surges driven by tropical cyclones or strong onshore winds; or (iii) a compounding of both processes occurring simultaneous or in close succession (Bevacqua et al., 2019; Bilskie and Hagen, 2018; Santiago-Collazo et al., 2019; Zheng et al., 2014; Zheng et al., 2013). Accurate representation of the bathymetry (i.e., water depth), topography (i.e., land elevation), and inundation barriers (e.g., levees, raised roadways, and natural ridges) is fundamental to accurately simulating floods (Bilskie, 2012; Dube et al., 2010; Gallien et al., 2018; Westerink et al., 2008). However, a critical challenge for predictive flood modeling is the geometric complexities of the terrain (Gallien et al., 2014; Xie et al., 2019). Model performance in low-gradient coastal regions is particularly susceptible to inaccurate topographical representation within computational models since the land elevation variation can be as few centimeters (Colby and Dobson, 2010; Van de Sande et al., 2012).

A prerequisite to the numerical flood model is the generation of high-quality structured or unstructured meshes that permit an accurate representation of complex domain geometry. Finite element- and volume-based models typically employ unstructured triangular meshes that are capable of resolving complex coastal domains (Chen et al., 2003; Ham et al., 2005; Namin et al., 2004; Pain et al., 2005; Shen et al., 2006; Xie et al., 2019; Yoon and Kang, 2004). The unstructured triangular mesh allows users to refine the mesh in critical areas and use coarse resolution in less sensitive regions such as in deeper bathymetries while maintaining a given computational cost (Bern and Plassmann, 2000; Hagen et al., 2001; Kim et al., 2014; Marsh et al., 2018; McGuigan et al., 2015).

In recent years, airborne Light Detection And Ranging (LiDAR) technology has grown more precise, and high-resolution (< 10 m) data have become increasingly available for supporting multi-dimensional flood modeling research (Bates et al., 2003; NOAA, 2007). Although the increasing terrain data resolution may permit an improved description of the bare earth topography, the unstructured meshes are restricted to a minimum resolution to minimize computational cost and numerical instabilities (e.g., Courant-Friedrichs-Lewy condition) (Bilskie et al., 2015). Limiting the resolution results in smoothing out the elevation of natural barriers and anthropogenic features (e.g., levees and raised roadbeds), which can alter the path of simulated inundation and result in an inaccurate solution (Bilskie et al., 2015; Horritt and Bates, 2002; Kim et al., 2014; Sofia et al., 2014).

Purvis et al. (2008) recognized this shortcoming in model resolution and manually digitized significant terrain features from UK Ordnance Survey maps to include their peak elevation within the LISFLOOD-FP inundation model. Bunya et al. (2010) applied the federal levees defined by USACE-MVN surveys, and the road and railroad crown heights taken from Atlas lidar surveys into high-resolution ADCIRC hurricane models. They concluded that the model accuracy is dependent on the high-level grid resolution of the terrain surface. Their efforts demonstrated that including long and narrow raised features are critical to build accurate flood inundation models. However, there are limitations in these methods. These limitations include the amount of hand digitizing and editing, low accuracy in the horizontal placement of crest elevations, a high number of person-hours, and the potential for errors. In sum, these findings motivate the automated extraction of significantly raised linear features (i.e., vertical features) from high-resolution terrain data.

Vertical features (VF) are raised linear features such as roadbeds, railroads, levees, floodwalls, and natural features that can block flow, conduct flow and redirect flow. The wetting and drying of an inundation front may differ depending on the unstructured mesh model with or without vertical features (Bilskie et al., 2015). There have been many studies aimed at automatically extracting ridge features from high-resolution topographic data. Roberts (2004) applied a method to take the point where the maximum gradient in a region is steep enough as VF points from LiDAR point cloud data. Then, anomalous points were manually cleaned and employed as raised feature points. These raised feature points were incorporated into the coastal flooding analysis. Coggin (2008) developed an automatic method for extracting VFs from watershed boundaries generated from LiDAR-DEM data for inclusion in an ADCIRC finite element mesh with special parameters. Bilskie et al. (2015) followed and expanded the work of Coggin (2008) and Roberts (2004) to extract VFs and fix them as polylines in an unstructured mesh that was employed in the simulation of shallow water hydrodynamics for Hurricane Katrina. The research results of Coggin (2008) and Bilskie et al. (2015; 2020) show that simulations of flood extents and depths are more accurate when using an unstructured mesh that includes VFs.

VFs play an essential role in the simulation of storm surge and hydrological flow routing. In hydrology, watersheds typically define land units, which have boundaries delineated from topographic high points (Edwards et al., 2015). Zhang et al. (2016) obtained ridges by extracting the watersheds of the river network since the watersheds correspond to the ridges in many real scenarios. Most hydrological models (e.g., SWAT, AnnAGNPS, HSPF, GSSHA) delineate watershed boundaries and topographic characteristics of watersheds using DEMs (Parajuli and Ouyang, 2013). Wang et al. (2011) predicted the spatial patterns of water yield by a SWAT-Road model and a SWAT-NoRoad model. The conclusion is that hydrologic effects of raised roads are important for accurately simulating runoff within a low-relief watershed. Alzahrani (2017) manually added VFs into a HEC-RAS 2D model and kept water away from the "dry" side of a VF until the water surface elevation was higher than the VF's elevation. Griffiths (2010) represented VFs as embankment arcs that alter the overland surface flow characteristics of a watershed, along with grid cell edges or elevated grid cells to simulate overbank flow in the GSSHA model. Thus, it is well-documented in previous efforts that the inclusion of vertical terrain features is critical for accurate surface water flow in coastal and riverine floodplains.

Most techniques for extracting raised linear features from LiDAR data are concentrated on breaklines (Coggin, 2008). In surface modeling, breaklines are linear features used to represent a sudden or abrupt change in the terrain's smoothness and continuity (Abdullah, 2017) or an otherwise string of connected points that should be honored by the data triangulation. They are commonly extracted from Airborne Laser Scanner (ALS) point cloud data, digital orthophotos, or ground surveyed cross-sections requiring cumbersome manual work (Bodoque et al., 2016; Briese, 2004; Brugelmann, 2000; Wang et al., 2018; Yang et al., 2016). Breaklines are often located at the toe and shoulder of levees and highways rather than along the highest point of the protruding feature (where the VFs should be located). In unstructured mesh generation, vertical features are breaklines considered as polylines or series of edges connected by triangular elements. Vertical features are used as a reference to form breaklines in unstructured mesh generation to improve the terrain description. In contrast to breaklines in surface modeling, vertical features are extracted from DEMs rather than LiDAR point clouds. They have special requirements, including being tall enough to affect flood propagation, long enough to span the edges of at least one element, and having an appropriate spatial distribution for the horizontal scale of the unstructured grid to be designed (Bilskie et al., 2015; Coggin, 2008).

The work of Bilskie et al. (2015) and Coggin (2008) show the usefulness of the inclusion of VF for flood models. However, they do point to some shortcomings that should be overcome. First, their methods require over a dozen parameters that should be manuall adjusted based on the surrounding terrain. Bilskie et al. (2015) states that future work should focus on parameter section for varying landscapes and VF-extraction sensitity to hydrodynamic model resolution. In addition, since their VF extraction methods begin with watershed boundaries, the final VF lines reside along DEM cell edges rather down the centroid. We aim to address these shortcomings through a revised VF-extraction algorithm that minimizes the number of parameters while considering various topographic landscapes from mountains to coastal regions.

This prevous research led to developing an automated VF delineation algorithm method based on a LiDAR-derived DEM for inclusion in flood inundation models. Section 2 describes the capability of the developed algorithm and software, called PyVF, and how VFs are delineated. In section 3, examples of extracted VFs are presented for four study areas in different types of geography to illustrate the capacity of PyVF for a variety of terrains. Section 4 contains a discussion of the PyVF method, and section 5 summarizes the research and conclusions.

2. Methods

PyVF is written as a Python version 2.7 script to take advantage of ArcGIS functions through the Arcpy module. All of the geoprocessing functions of ArcGIS, such as data analysis, data conversion, and data management can be accessed through Python using Arcpy, which is a Python site package that integrates ArcGIS with Python(Esri, 2016). PyVF also includes Numpy (Walt et al., 2011) for data manipulation and the Python standard Math library (Lundh, 2001). The design of PyVF is divided into four main tasks: 1) batch processing sub-DEMs; 2) VF target recognition; 3) VF delineation, and 4) post-processing of potential VFs. PyVF produces VFs as a shapefile and two raster images with attributes, described in sections 2.3 and 2.4.

PyVF, like the method proposed in Coggin (2008) and Bilskie et al. (2015), aims to extract VFs that are high enough and long enough. Their method evaluates the relative elevation by comparing the height of a vertex on the watershed boundary line with the height at two perpendicular distances from the vertex. According to the height difference, each vertex is declared as “significant” or “continue”. If the “significant” vertex is below a ratio (e.g., 35%), the watershed line is eliminated. The extracted VFs from this previous method are a subset of the watershed boundaries. The path of the watershed boundary is along the edge of the DEM cells, and so are the extracted VFs. This can lead to large height errors when placing mesh nodes on narrow VFs and coarser DEMs. To reduce the height error in the model and provide more meaningful parameters, an iterative increasing size moving window method is employed to search for potential VFs cells by calculating the height trend of all DEM cells in eight directions. The extracted VFs with meaningful attributes are along the center of the cells.

The flowchart in Figure 1 provides a general description of the PyVF algorithm. The two inputs of the algorithm are a DEM and a target unstructured mesh element size (ES). The ES also can be replaced with a constant value. First, the DEM is split into sub-DEM tiles to efficiently utilize computer memory, useful for large domains with small cell sizes by batch processing. Next, two rasters, which have the value of r and dh, for all tiles are extracted through the target recognition (TR) method. Two thresholds of r and dh, min_r and min_{dh}, are used for reducing the VF cell candidates to avoid weak VF cells and extraneous cell noise. They are defined through the 1.5 x IQR (interquartile range) rule and explained in the following section. The DEM also serves as the input to watershed boundary delineation. The extracted watershed boundaries are considered ridges. The reduced VF cells that coincide with watershed boundaries are potential VF raster. Then, the potential VF cells continue to be deleted to create a linear feature of a single cell width by a thinning approach. This is in preparation to convert the linear raster to vertical feature polylines. Finally, post-processing is performed based on the constraints of the individual modeling study that will utilize the extracted VFs. For example, when vertical features are applied to an unstructured mesh model, the element size is required to determine the appropriate length of the final vertical feature polylines.

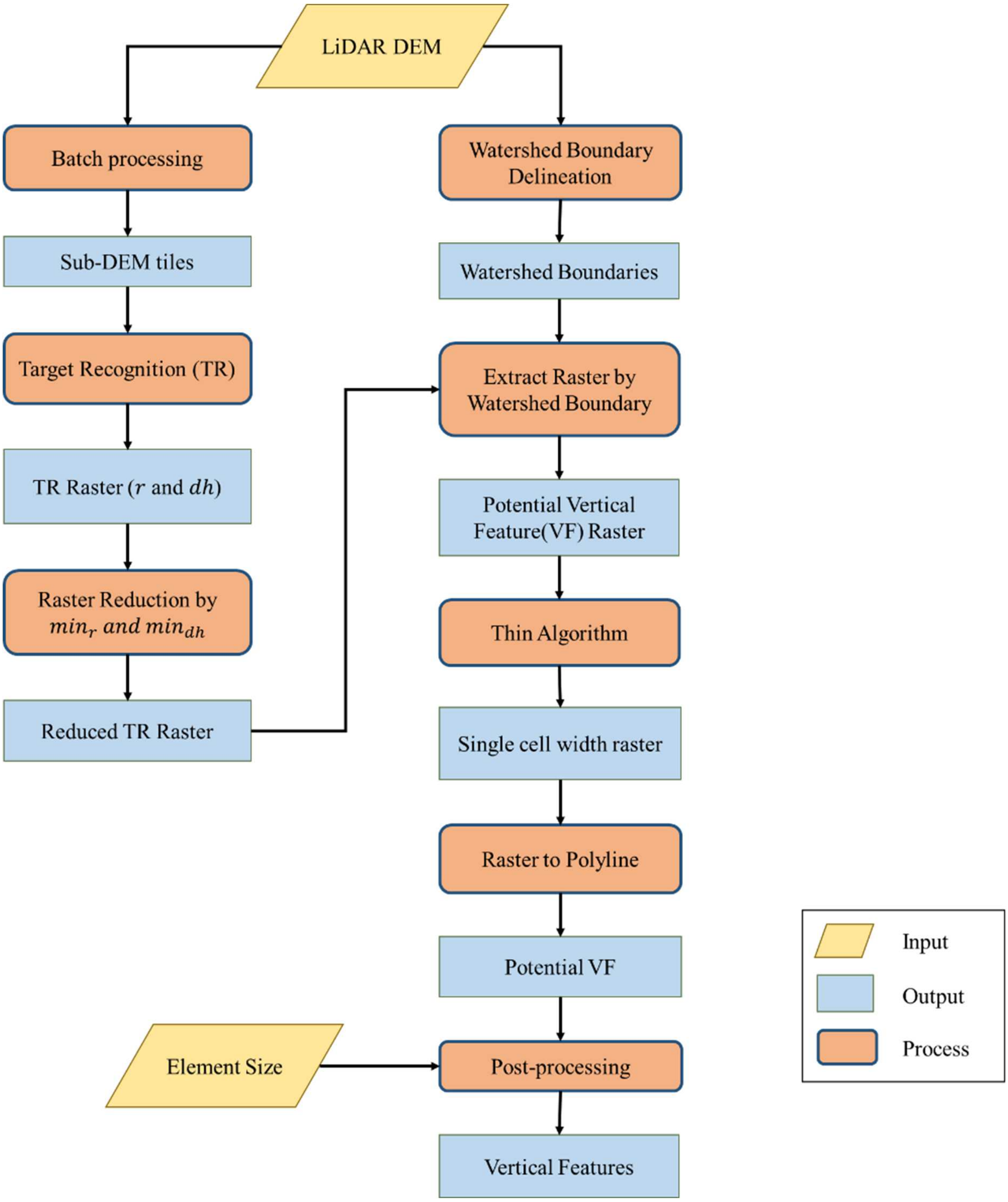

Figure 1. A flowchart outlining the vertical feature delineation procedure beginning with the LiDAR-DEM.

2.1 Batch processing

The high-resolution raster datasets across large domains result in large amounts of data and processing challenges due to the computer memory limitation. Batch processing is a common method for overcoming memory limitations. There are two methods provided for dividing the large raster into tiles. Tiles are read into memory, processed and written to disk one by one until the task is complete. The two methods have a common purpose - to split the DEM so that the data volume of each sub-DEM region can reside in memory and minimize discontinuities or gaps in the final mosaicked raster image.

The first batch processing approach is based on customized rectangular polygons shapefile. The grid polygons used in this research are a net of square polygons. The large region DEM is clipped into sub-DEM by the polygon shapefile. It should be noted that the side length of each tile in the polygon shapefile must be a multiple of the DEM cell size, otherwise, there will be gaps between each tile. The sub-DEM region is called the recognition area as a minimum unit for the following target recognition method.

Another DEM decomposition approach is dividing the DEM into many tiles from the upper-left corner according to the DEM coordinate and using coordinates i and j, with (0,0) denoting the upper-left corner of the DEM. The minimum unit (i.e., the recognition region) is a number of rectangular $i \times j$ tiles where i is the number of rows j is the number of columns. Each tile is processed in the target recognition algorithm individually (one by one). This method is better for manipulating the entire DEM region without creating a grid polygon.

The target recognition method in this research applies a moving window approach. Since the raster edge cells (e.g., the cells in the top and bottom rows and the left and right columns) do not have sufficient neighbors. For example, in Figure 2, the center of the moving window is on the cell at the left edge. The lack of data in the moving window will affect the calculation. If the edge cells of each sub-DEM cannot be calculated as non-edge cells, apparent discontinuities will result along the edges of each tile in the final mosaicked image. Hence, a buffer distance around the recognition area is determined to define an effective area. Effective areas, polygon tiles at a specified distance around recognition areas, are determined to solve this problem. The role of effective areas is that when the moving window traverses each recognition area, there is no null-value inside the window range. The following section will introduce the method of searching potential VF cells from each recognition area.

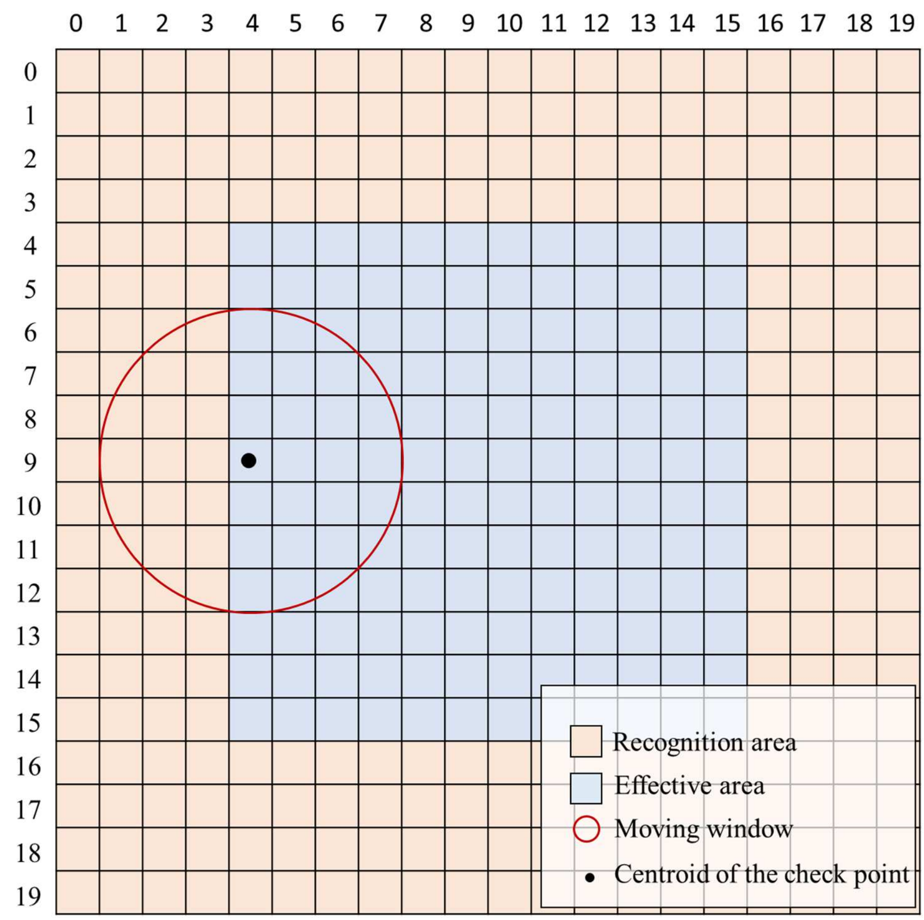

Figure 2. Recognition area and effective area defined for Batch processing

2.2 Target Recognition

The VF cell recognition method in this research adopts a circular moving window approach, which has an advantage in directional uniformity over a square moving window (Chang and Sinha, 2007; Chang et al., 1998; Koike et al., 1995). The centroid of the circular moving window is placed at the center of each cell. The circular moving window is divided into eight sectors representing eight directions (e.g., A1 is North, B1 is Northeast) (Figure 3 (a)). The target recognition method traverses each target DEM cell and aims to find the highest cells in at least one symmetry direction. The higher elevation cells are identified as potential VF candidates.

Two parameters: r and dh are calculated by the target recognition method. The first parameter r is the radius of the moving window. Moving windows can use a fixed-size or iteratively increasing size. Casas et al. (2012) proposed a method for assessing the structural integrity of levees. This method relies on the slope calculation based on a 3 by 3 moving window. However, small fixed moving windows are not always the best choice (Lin et al., 2013). In this research, an iteratively increasing circular moving window is applied. Hence, the circular moving window expands as the r value increases. The initial value of r is 1.5 times the cell size. The second parameter dh is the difference in height between the value of the checked cell with that of the lowest cell. For example, Figure 3 (b) shows an assumed cross profile of a North-South VF and the "Check cell" is located on the VF. To the west of "Check cell", the lowest cell can be found when r is equal to 3.5 cell size. The r in the west and east direction are represented by r_W and r_E. The dh in the west direction is dh_W. In the east of "Check cell", the lowest cell is found when r is equal to 5.5 cell size. The dh in the west direction is dh_E . A different value of r will lead to a different value of dh. This example illustrates that a variable window size is more appropriate for extracting VFs than a constant size since the width of the VFs is not fixed.

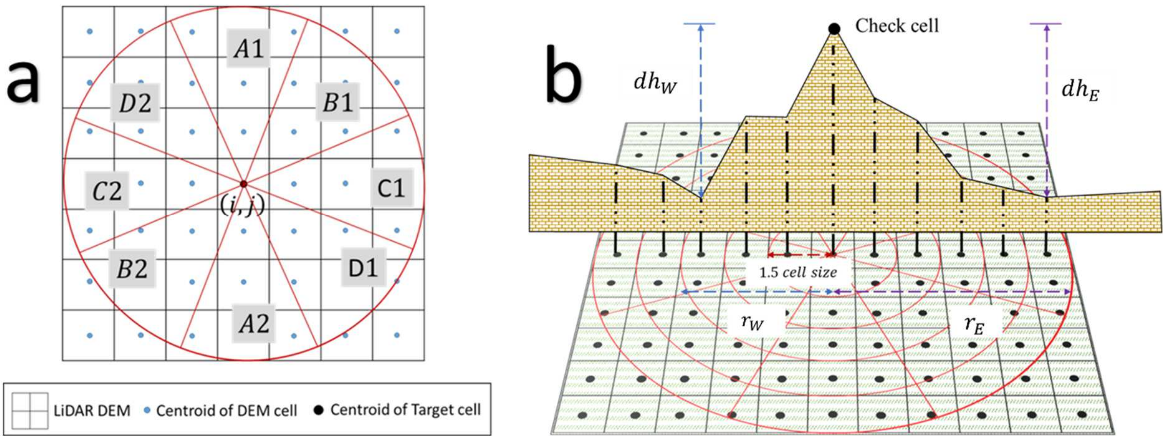

Figure 3. a) The circle is divided in to eight sectors to present eight directions of a target point to be examined; b) This is an assumed small-scale cross profile of terrain. The r of lowest elevation in each direction are different, which can show a variable window size is more appropriate for this research.

For example, Figure 4 shows a hypothetical large-scale cross-section spanning multiple raised features to illustrate the desired target cell location for this study. Because the actual terrain is very complex, there are many possible elevation relationships between the potential VF cells and their surrounding points on the same cross-section: (1) The transverse profile is approximately symmetric and vertex position is very clear (i.e., inversed V-shape) (Figure 4 (b)(d)); (2) The transverse profile is approximately symmetric and top is wide (i.e., inversed U-shape) (Figure 4 (c)); (3) The transverse profile is asymmetrical and vertex position is very clear (Figure 4 (a)); (4) The transverse profile is asymmetrical and the top is wide (Figure 4 (e)). The black dots are non-target points and hollow dots are target points in Figure 4. Although the transverse profiles have different forms, the common feature of the target cells is that they must be the highest point within a certain distance in a symmetric direction.

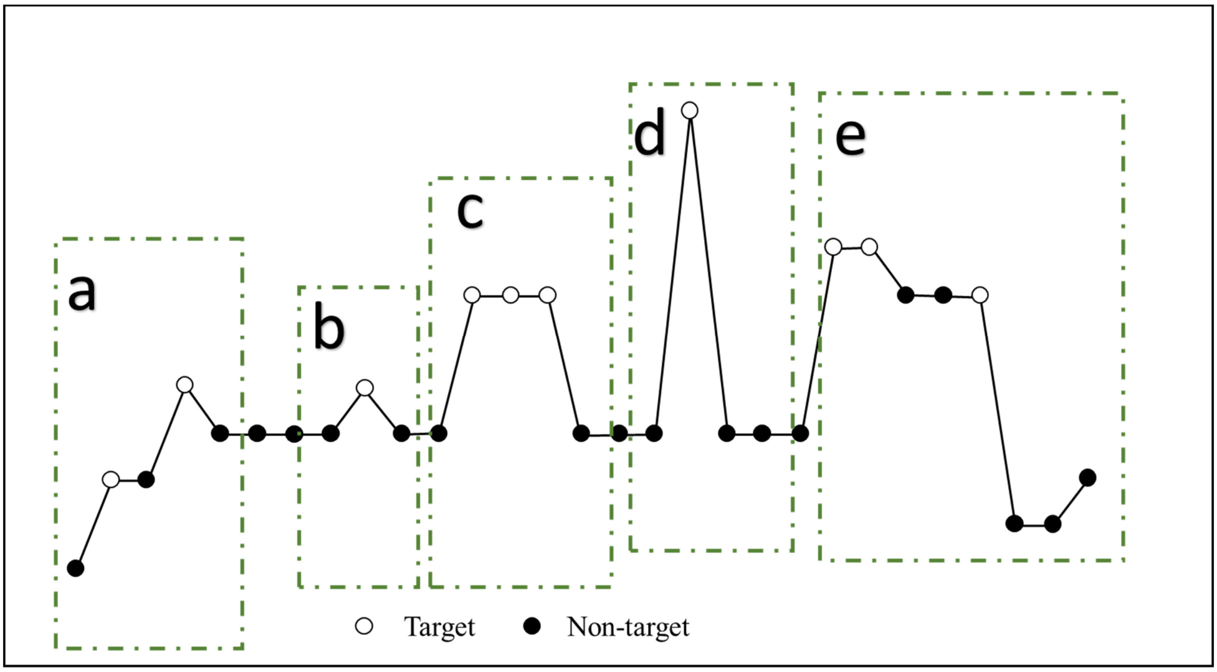

Figure 4. An example large-scale cross profile of terrain. Empty circles are potential VF points having a variety of elevation relationships with their neighbor points on the cross-section.

The flowchart for the VF recognition method in the E-W direction is presented in Figure 5. The variables and their descriptions are summarized in Table 1. First, the target recognition algorithm aims to find the highest cells in at least one symmetrical direction. This is determined by comparing the elevation of the checked point (H_0) with the elevation around the checked points (e.g.,HW_i , HE_i), that is dh_W and dh_E in the E-W direction. When i is larger than 1, the dh_W and dh_E is computed from H_0 minus mHW_i and mHE_i , respectively. When the elevation values in one of the symmetrical directions stop getting lower, that is $dmHW_i$ (i.e., $mHW_i - mHW_{i-1}$) or $dmHE_i$

(i.e., $mHE_i - mHE_{i-1}$) is less than 0 m, each direction must be checked separately to ensure that this situation in Figure 4 (a, e) is not overlooked. That is, in Figure 4 (a,e), starting from the check point, after a certain distance of elevation drop, the height remains stable, or even rises, and then continues to fall. If the calculation loop terminates when the height does not drop further, a smaller dh is obtained than the dh returned by continuing the calculation. The check point with small dh value is likely to be filtered out due to the insignificant height difference. A short ascent is allowed to ensure that the dh is closer to the actual situation and avoid deletion of important VF points. Taking west direction as an example, this process is achieved by setting the thresholds to two parameters t and $dmHW_i$ (i.e.,max_t and max_{dmH}). The $max_t$is employed to limit the distance of ascent so that the height does not continuously decrease with increasing r. The max_{dmH} limits the height of each ascent.

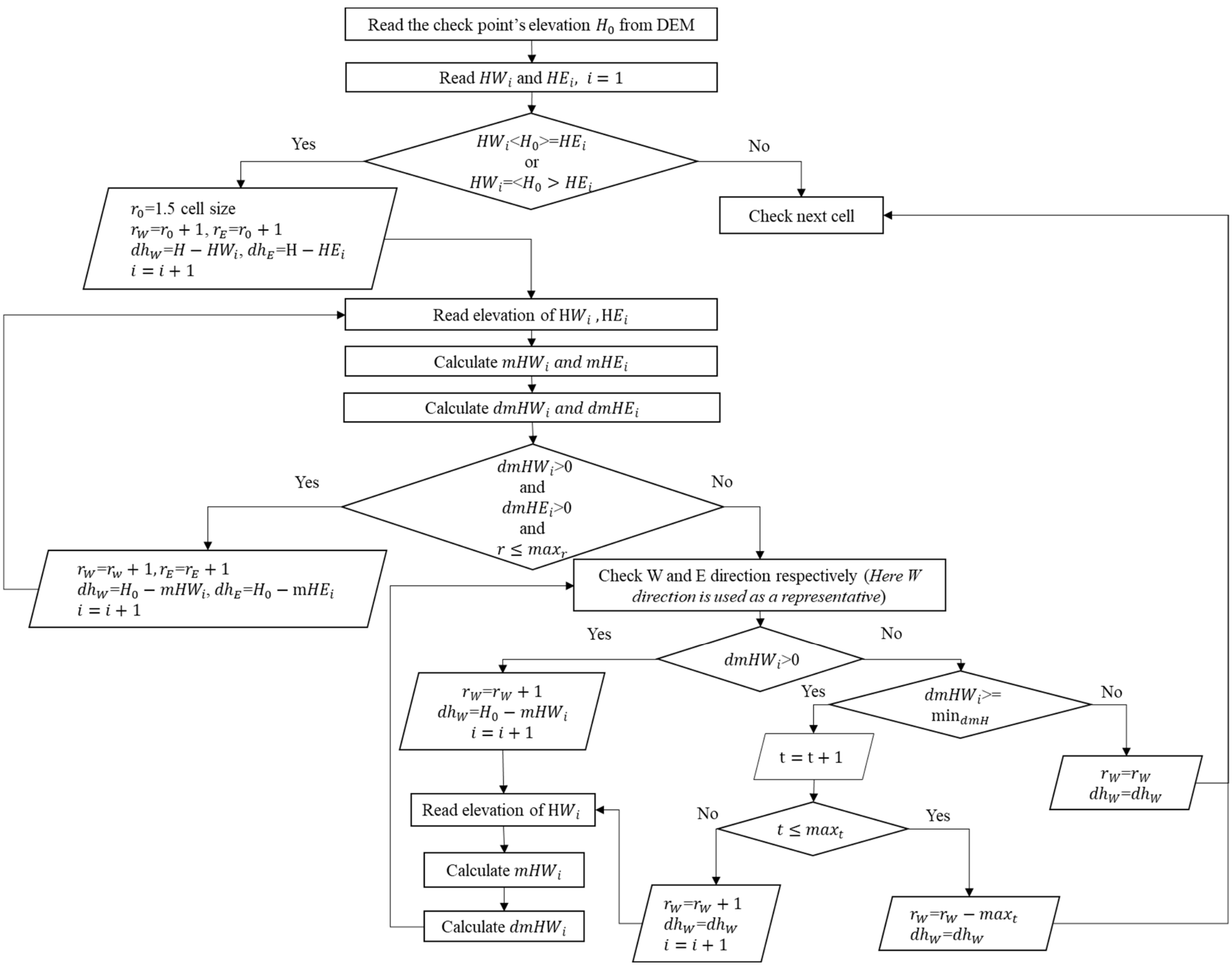

Figure 5. The flowchart for the main vertical feature recognition method.

Table 1. The variables in the flowchart for the target recognition algorithm in E-W direction.

Variable	Description
i	Increase the size of the moving window for the $(i-1)^{th}$ time
H_0	The elevation of checked cell
r_0	The initial value of r
max_r *	The maximum number of r.
r_W	The radius of moving window in West direction
r_E	The radius of moving window in East direction
dh_W	The elevation difference in West direction
dh_E	The elevation difference in East direction
HW_i	The value of the cells contained in the i^{th} ring in the West direction
HE_i	The value of the cells contained in the i^{th} ring in the East direction
mHW_i	The average value of the cells contained in the i^{th} ring in the West direction
mHE_i	The average value of the cells contained in the i^{th} ring in the East direction
$dmHW_i$	The difference between mHW_i and mHW_{i-1}
$dmHE_i$	The difference between mHE_i and mHE_{i-1}
max_{dmH} *	For every increase of 1 in r, the maximum height allowed to rise
t	A tolerance that allows stability or rise over a short distance
max_t *	The maximum of t. It can be defined by users.

* The value determined by users

The outputs of this portion of the framework are two target recognition raster files and an attribute table. The values of the two generated rasters are the maximum radius (r) and the maximum height difference (dh) of the eight directions. The attribute table includes r and dh for each direction.

For VF cells, r represents how wide and dh represents how high. Users must define two thresholds min_{dh} (i.e., the minimum dh) and min_r (i.e., the minimum r) to filter raster cells for their study. That is, all remaining VF cells have dh values greater than min_{dh} and r values greater than min_r. These VF cells are the input for the following target delineation method. The min_{dh} is determined by the 1.5 IQR rule and min_r is recommend to be 2.5 * cell size. The details of the two thresholds and the recommended values for different study areas will be provided in section 3.

2.3 Target Delineation

Next, the VF raster is converted to feature polylines. The potential VF cells identified from target recognition are more than the desired VF cells as this is the first phase in VF delineation. Raster cells of potential VF points that do not meet required criteria (i.e., not high enough) are removed based on values of r and dh. The remaining potential VF cells form wide and linear raster cells or even blocks of raster cells. Specific examples and values of *r* and *dh* are discussed in the following sections.

There is an assumption that significant barriers to surge propagation will be captured as watershed boundaries (Bilskie et al., 2015). Some programs (e.g., TauDEM, GDAL, ESRI's ArcHydro extension) were developed to delineate watershed boundaries from DEM (Kraemer and Panda, 2009; Tarboton, 2005). Through establishing flow direction, linking flow path, and calculating flow accumulation based on a DEM, cells with a flow accumulation value of zero generally correspond to watershed boundaries. Also, it is impossible to point out which of the two adjacent cells that share the watershed boundary is higher. Hence, the watershed boundaries are buffered by a distance of one cell size on both sides. However, the potential raster cells covered by the buffered watershed boundary is desired.

At this stage, the width of the linear potential VF raster is two cell sizes at a minimum. When generating VF polylines it is necessary to reduce the number of cells to create a linear feature of a single cell width. This is accomplished by using a thinning (i.e., skeletonization) approach (Davies and Plummer, 1981; Naccache and Shinghal, 1984; Zhan, 1993; Zhang and Suen, 1984). The thinned linear raster can then be converted to polyline by ArcGIS or similar GIS software. Therefore, the VF polylines reside along the centroid of raster cells and not on the edge.

2.4 Post-processing

Post-processing is required depending on the research objective. The main purpose of our VFs extraction method is to help guide the unstructured finite element mesh generation for a flood inundation model. Therefore, post-processing focuses on retaining sufficiently long VFs relative to the element size, including removing shorter VFs and bridging the gap between VFs. This is to facilitate the placement of nodes (and element edges) along the VF lines. For example, the ADvanced CIRCulation (ADCIRC) model employs unstructured finite element meshes that are widely used for predicting storm surge-generated coastal inundation across normally dry regions (Bhaskaran et al., 2014; Bilskie et al., 2016; Bilskie et al., 2014; Bunya et al., 2010; Dietrich et al., 2011; Gayathri et al., 2016; Luettich, 1992). If the length of the extracted VF is less than the length of the desired local element size, it is not possible to directly include it in the mesh. Therefore, VF lines with lengths less than the desired local mesh size must be removed. Additionally, there may be small gaps in the VFs that should be connected. Further details are described in the following section.

3. Applications of PyVF

This section highlights some applications of the PyVF methods. PyVF is employed to extract VFs from four distinct landforms: low-gradient coastal region, urban region, mountain region, and beach region. For all regions, the values of the parameters in target recognition and target delineation methods are recommended. Table 2 summarizes the description and values of the parameters in the four study areas. The application of batch processing is described in the mountain region. The cases requiring post-processing are illustrated in the low-gradient coastal region.

Table 2. Summary of PyVF parameters and respective values used in four distinct landforms

Variable	Description	Reference	Study Area			
			Low-gradient coastal	Urban	Mountain	Beach
min_{dh} (m)	Minimum height difference	1.5 IQR rule	0.2	1	50	1.5
min_r (m)	Minimum radius	2.5*cell size	7.5	7.5	25	25
T	Minimum number of cells that constitute a stream	Trial and error	5000	1000	10000	100

3.1 Low-gradient coastal region area

Low-gradient coastal areas have a higher probability of flooding from storm surges and prolonged torrential precipitation than regions with higher elevation gradient terrains. Moreover, especially in low-gradient regions, the potential of more destructive flooding from compound events is often higher than the occurrence of a single event (Bevacqua et al., 2019; Ikeuchi et al., 2017; IPCC, 2013; Moftakhari et al., 2017; Nicholls et al., 2007). This region is difficult to model due to the complicated flows of coastal storm surges, rainfall-runoff and fluvial flooding that can occur in combination (Bilskie and Hagen, 2018; Santiago-Collazo et al., 2019).

The low-gradient coastal study area displayed in Figure 6 is a part of the Lake Maurepas watershed in southeastern Louisiana. Figure 6 shows the aerial imagery draped over a 3-m resolution LiDAR derived topo-bathymetric DEM provided by U.S. Geological Survey (USGS). Some roadbeds and natural barriers that can alter the path of flood flow are shown in the figure. The elevation in this area ranges from -0.5 m to 11.7 m (NAVD88).

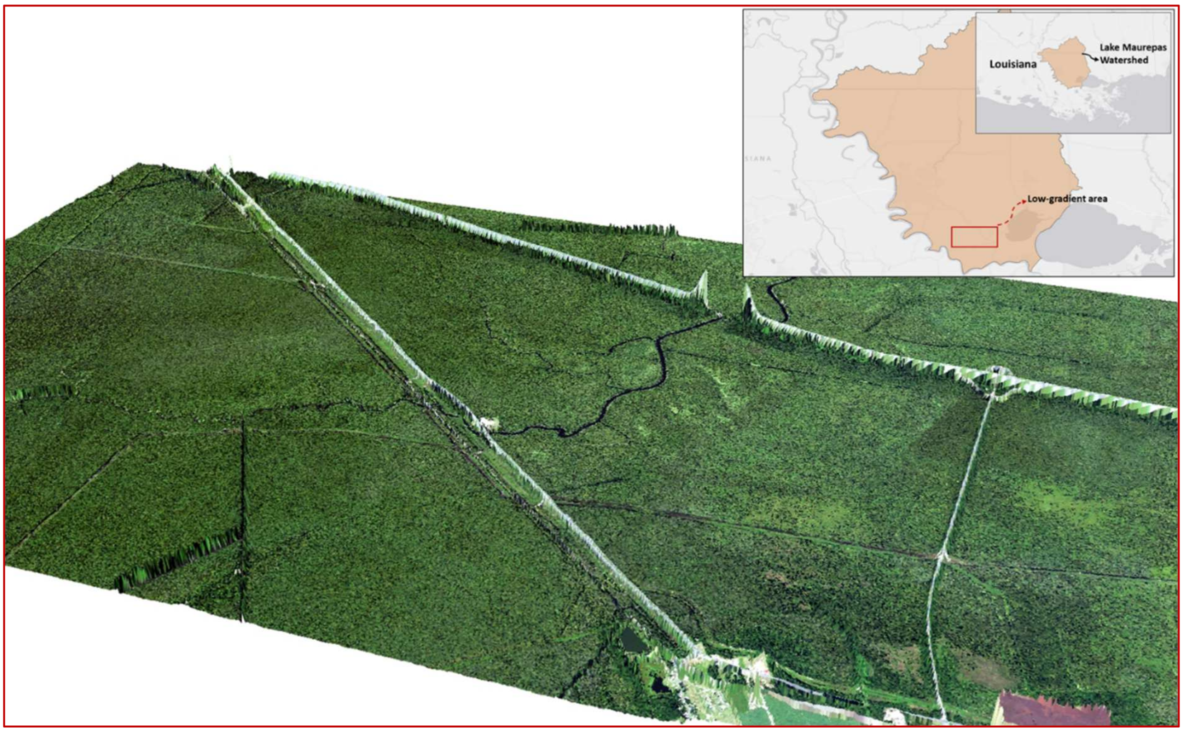

Figure 6. Aerial imagery draped over a 3-meter resolution topo-bathymetric digital elevation model in southeastern Louisiana.

The elevation difference dh, which is a result of the target recognition method, is shown in Figure 7 (a)(a'). The darker the color, the larger the elevation difference. The VFs of interest have a larger elevation difference. However, there is a large amount of VF cells with small elevation difference (i.e.,dh). These cells are not regarded as potential vertical features. Hence, min_{dh} is the major parameter to provide a threshold to identify the VF cells.

Some geomorphometric parameters such as slope, curvature, elevation residual and entropy are used in terrain analysis to extract vertical features (Hiller and Smith, 2008; Sofia et al., 2014; Tarolli et al., 2010). The min_{dh} has the similar core idea with elevation residual (ER), that is, to filter low-relief plains in local scale. The ER is calculated as following equation.

$$ER = E_{DEM} - \bar{E}_{mean\_r} \quad (1)$$

where $\bar{E}_{mean\_r}$ is the average elevation of cells within a moving window with a fixed size $mean\_r$, which is the average r of the entire study area, and E_{DEM} is the elevation of the cell in the center of the moving window.

The statistical IQR is used to define the threshold value of geomorphometric parameters (Hiller and Smith, 2008; Sofia et al., 2014). Therefore, the IQR is feasible to analyze dh. The VF cells can be regarded as the outliers of the entire DEM cells. The min_{dh} should satisfy the condition:

$$IQR = Q3 - Q1 \quad (2)$$

$$min_{dh} > Q3 + n * IQR \quad (3)$$

where Q1 is the first quartile, Q3 is the third quartile, and n is a parameter defined by users.

1.5 IQR (i.e., $n = 1.5$) is the commonly used rule to define outliers. In this region, the min_{dh} in 1.5 IQR rule is about 0.2 m (Figure 13). A value greater than 0.2 can be considered as a min_{dh} value, and the upper limit is recommended not to exceed the average value of outliers (~0.6 m). Since the width of VFs in this study is not regarded as an important indicator for VF extraction, the selection of minimum r is only used to delete those discrete local high cells. Hence, the value of minimum r is related to the resolution of DEM, which is generally 2.5 (i.e., (initial r) +1) times the cell size. The potential VF raster image using 0.2 m min_{dh} and 7.5 m min_r (i.e., 2.5 cell size*3m) is shown in Figure 7 (b) (b'). To avoid noise, the value of the parameter min_{dh} is selected as 0.5 m and min_r is 7.5 m (Figure 7 (c) (c')).

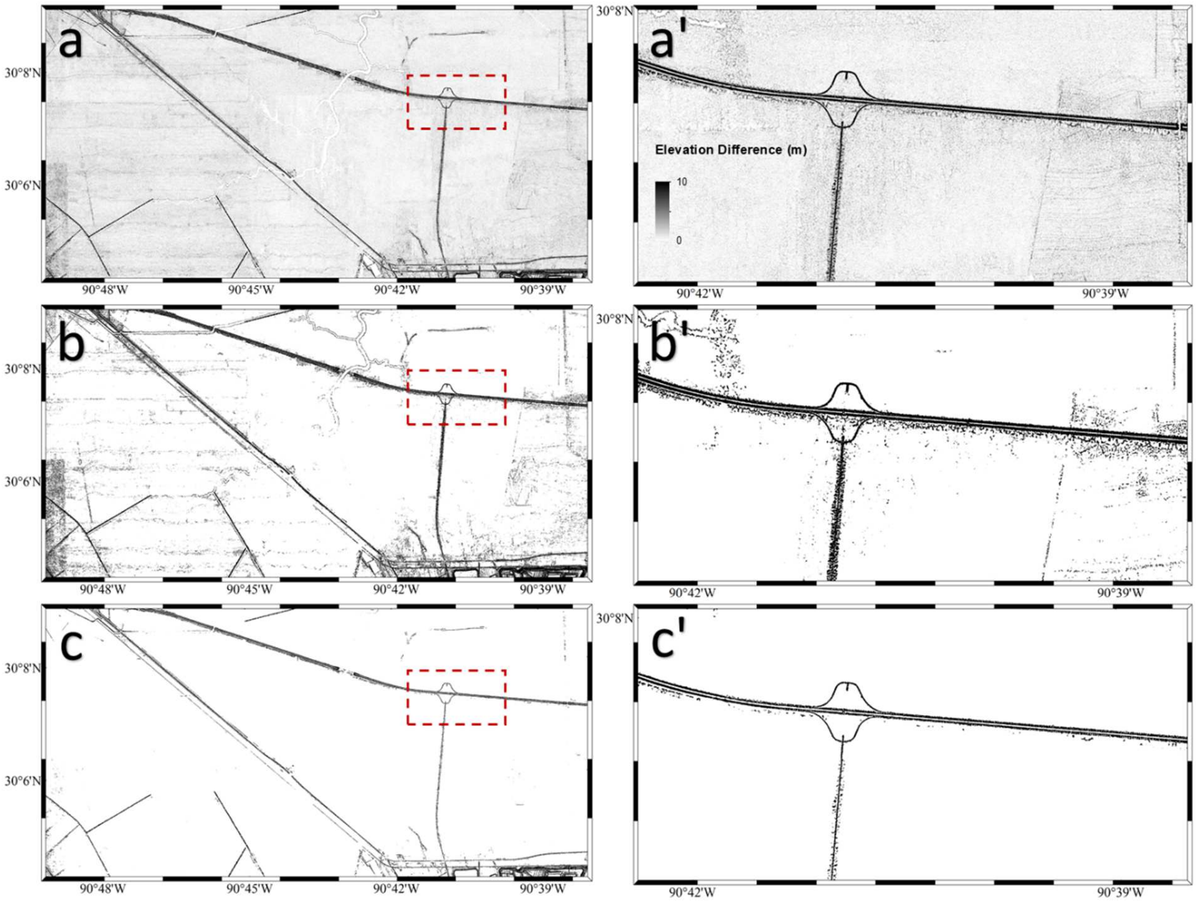

Figure 7. a) The raster image with the value of dh in low-gradient area. a') A zoom-in of the raster image a. b) The potential VF raster image with the value of $\min_{dh}$ is 0.2 m and the value of $\min_{r}$ is 2.5 cell size in low-gradient area. b') A zoom-in of the raster image b. c) The potential VF raster image with the value of $\min_{dh}$ is 0.5 m and the value of $\min_{r}$ is 2.5 cell size in low-gradient area. c') A zoom-in of the raster image c.

As previously mentioned, the extracted cells are wide and difficult to convert to lines. The potential VF cells are delineated with the aid of watershed boundaries (Figure 8). A threshold (T) that represents the minimum contributing cells in the drainage network needs to be selected. If the threshold is too small, the flow accumulation will be too short, resulting in more watersheds and short watershed boundaries. Conversely, if the threshold is too large, some important watershed boundaries will be omitted. The process of selecting the threshold is conducted through trial and error, while iterating on target recognition results. Four thresholds are chosen: 20000 cells, 10000 cells, 5000 cells and 2000 cells. The watershed boundaries generated by the accumulated flow threshold of 5,000 is determined suitable for the VF extraction. The delineated watershed boundaries are along the edge of the cells. There is no method to determine whether the watershed boundary lies to the right or left of the potential VF cells. So the watershed boundaries (polylines) are buffered by one cell size distance on both sides. The potential VF cells that overlap with the watershed boundary buffer (polygons) are extracted. Then a linear feature of a single cell width for generating polyline by a thinning approach. The polylines converted from the thinned raster cells are the initial set of VF lines (Figure 9).

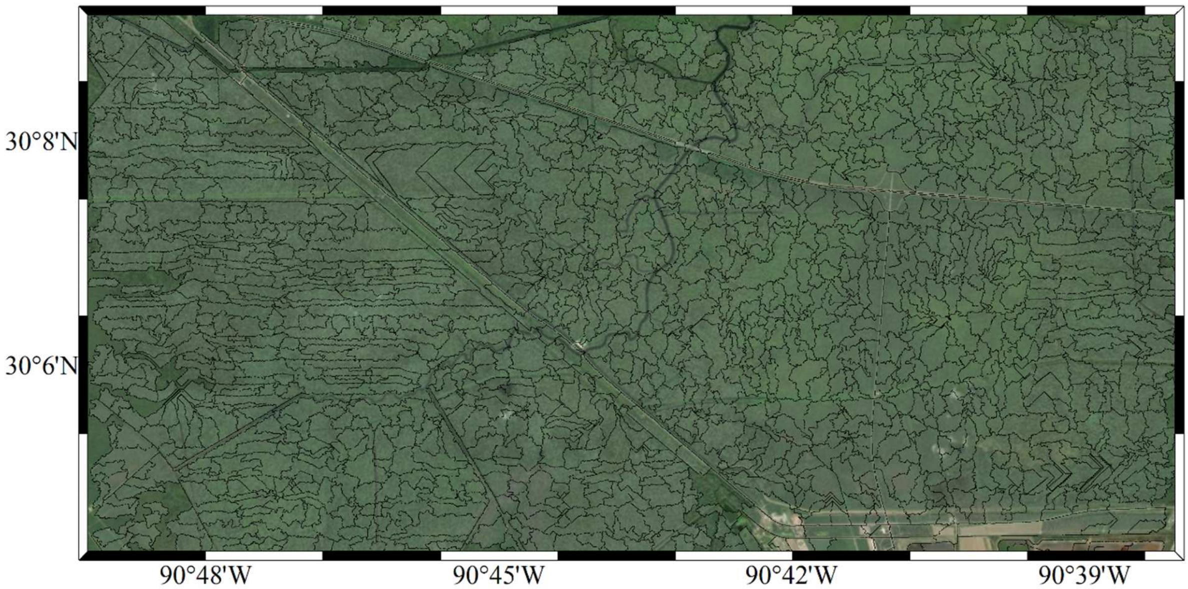

Figure 8. Watershed boundaries with a 5,000 accumulation flow threshold.

For inclusion in the mesh generation processes, the VFs should be further processed to remove VF lines shorter than the minimum element size and to connect small gaps. For example, the VFs shown in Figure 9 (a) are shorter than the surrounding desired mesh element size and, therefore, deleted. Additionally, there may be small gaps in the VFs that should be connected (Figure 9 (c)). Gaps should remain intact when a river flows through a VF (Figure 9 (d)). In addition, VF polylines that have a large bend at the end are not conducive to mesh generation. (Figure 9 (e)). Also, parallel VF lines that have a separation distance within a given element size should be compared to decide which should be kept (Figure 9 (b)). Closed loops caused by thinning are cleaned (Figure 9 (f)). As a result of the automated post-processing routines, a cleaner and more meaningful set of VF lines are produced for mesh generation (Figure 10 (a)). Furthermore, the location of the extracted

VFs is along the centroid of raster cells as opposed to watershed boundaries that are on the edges of raster cells (Figure 10 (b)).

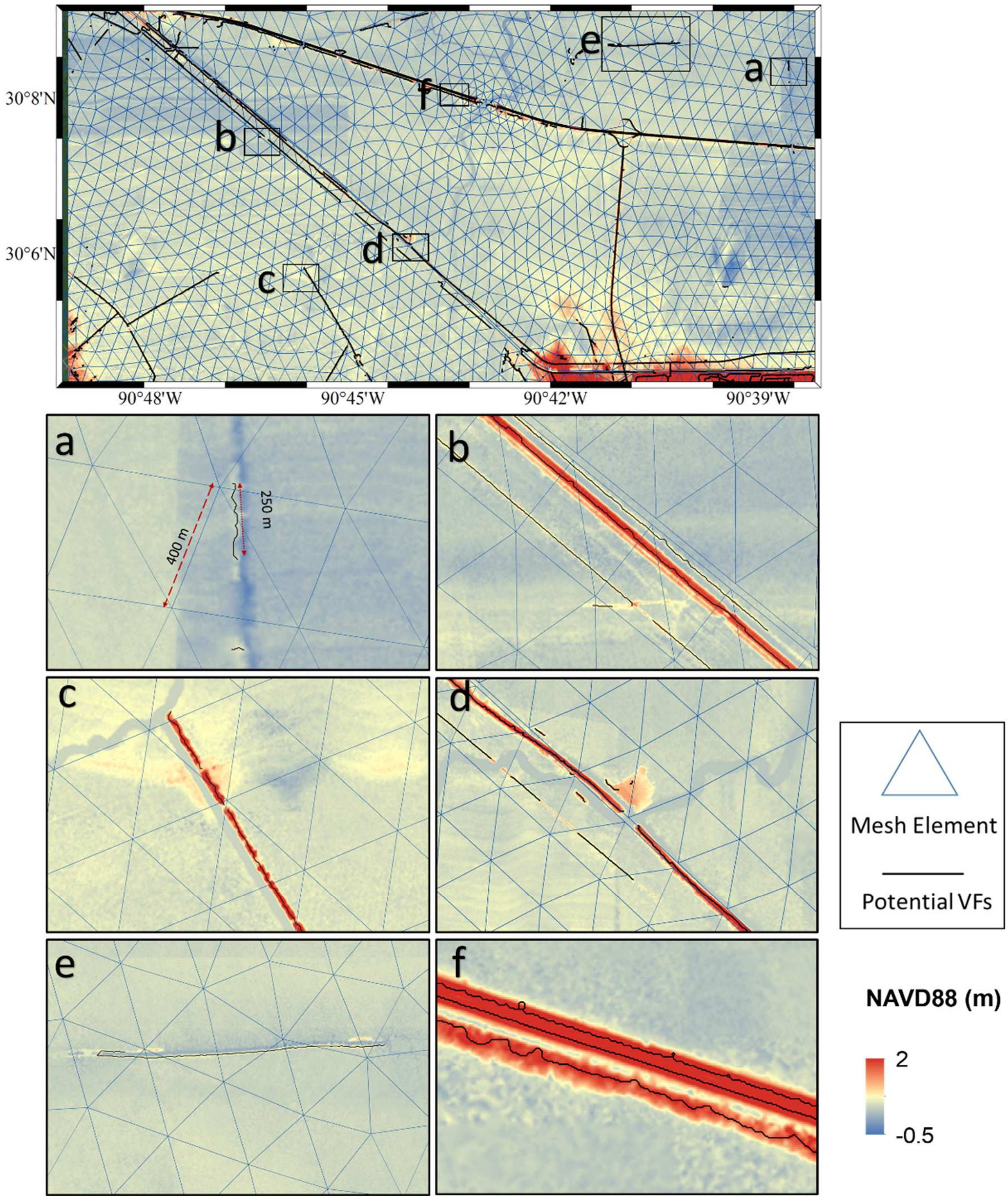

Figure 9. Examples of potential VFs requiring post-processing. (a) Lines shorter than element size; (b) Adjacent parallel lines; (c) small gaps between potential VFs; (d) small gaps with a river flowing through it; (e) a line with a large sinuosity at the end; (f) closed loop

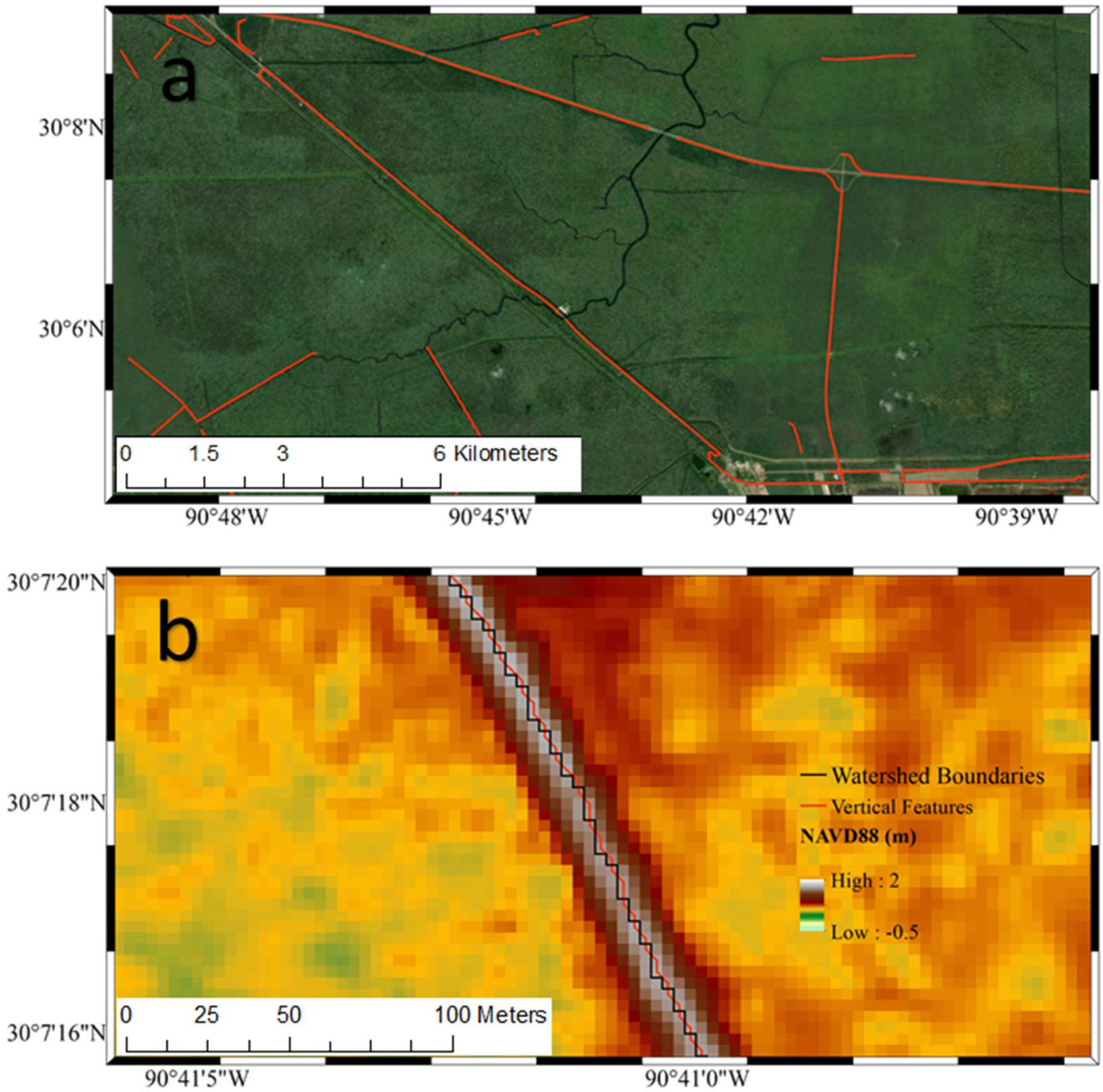

Figure 10. a) Vertical features after post-processing for low-gradient coastal area location. b) Comparison of a watershed boundary and a vertical feature

3.2 Urban Area

VF extraction using PyVF was also tested for an urban area – Port Allen and Baton Rouge, Louisiana (Figure 11 (a)). The Mississippi River passes through the study region. The USGS 3-m resolution LiDAR topo-bathymetric model was used as the source DEM Figure 11 (b). The elevation of Port Allen is substantially lower than that of Baton Rouge, however, the average elevation differences within each city are small. The 1.5 IQR of the dh is about 0.66 m (Figure 14). The Figure 11 (c) shows that there are many potential VF cells in the Mississippi River region.

Then the min_{dh} and min_r are set to 1 m and 7.5m. In other words, the value of recognized potential raster cells with an elevation greater than 1 m and the r is greater than 7.5 m (Figure 11 (d)).

The watershed boundaries are generated with an accumulated flow threshold of 1,000 (Figure 11 (e)). Through post-processing, the VFs derived from PyVF with lengths greater than 200 m are retained. The black dotted line shown in Figure 11 (f) (g) is the Mississippi River east and west bank levees obtained from Nation Levee Database. It is obvious that levees are extracted and there are many non-levee VFs that can impact flow path in this area, especially in the Port Allen area.

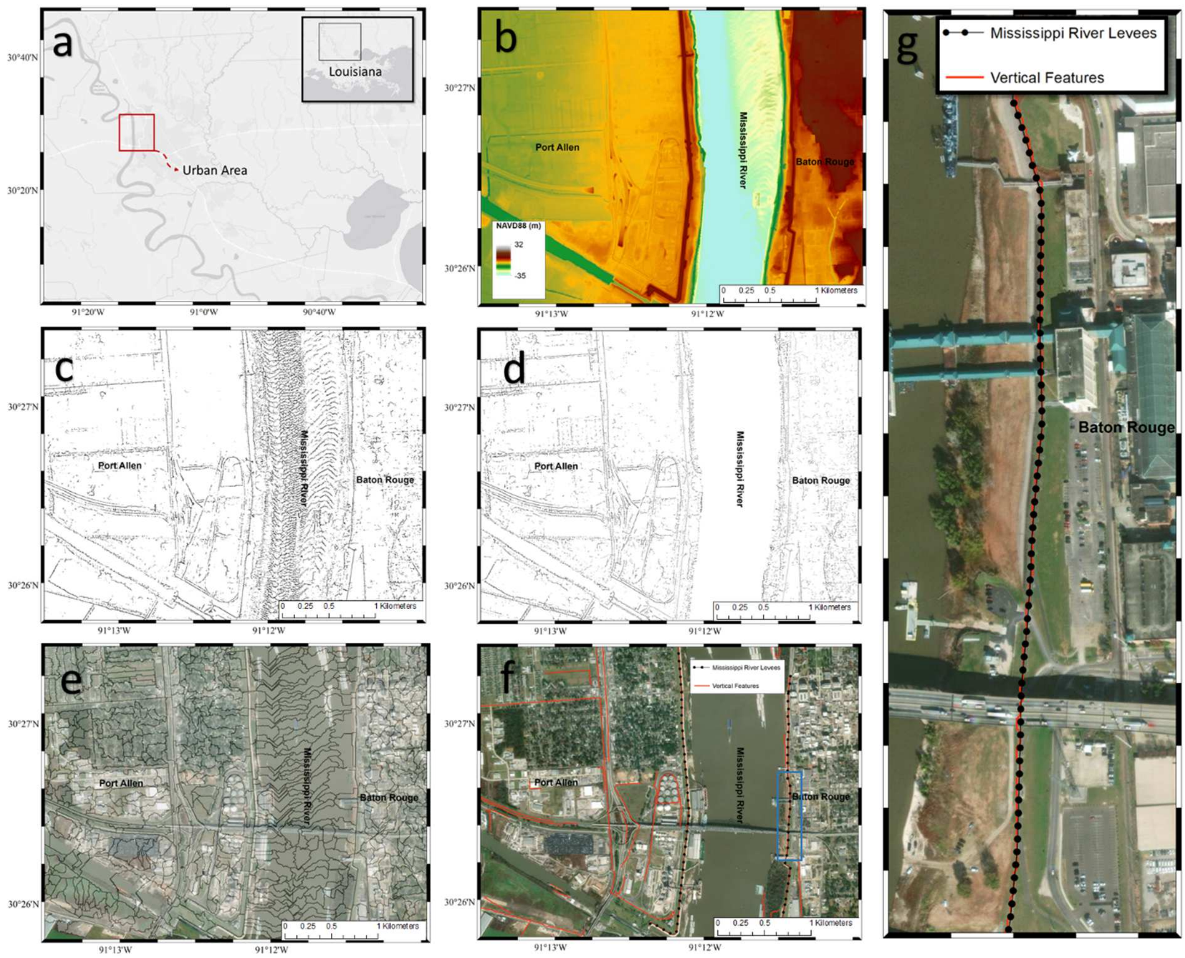

Figure 11. a) Location map of urban area in Louisiana. b) The DEM in urban area. c) The potential VF raster image with the value of $\min_{dh}$ is 0.66 m and the value of $\min_r$ is 7.5 m in urban area. d) The potential VF raster image with the value of $\min_{dh}$ is 1 m and the value of $\min_r$ is 7.5 m in

urban area. (e) Watershed boundaries with a 1000 accumulated flow threshold. (f) Vertical features greater than 200 m and Mississippi levees from Nation Levee Database. (g) A zoom-in of the blue box in (f) presenting the extracted VF and the Mississippi levee.

3.3 Mountain Area

The mountain study area is a region of north Georgia and is about 20 by 40 km Figure 12 (a). The 10 m DEM in this area was obtained from the Nation Elevation Dataset (NED) assembled by the USGS (Figure 12 (b)). This site is larger than the previous two, so batch processing the DEM was necessary. The area is divided into six titles and PyVF is run individually on each tile to obtain potential VF raster cells for the entire mountain area.

The watershed boundaries are generated with an accumulated flow threshold of 10,000 (Figure 12 (c)). The 1.5 IQR of the dh is about 50 m (Figure 14). This value is reasonable since the characteristics of mountain areas are substantially higher than the surrounding terrain and includes large slopes. The minimum dh and r are set 50 m and 25 m (Figure 12 (d)). The VFs with a minimum length of 200 m and 1,000 m are shown in (Figure 12 (e) (f)). There are many short branches using 200 m as the minimum length. PyVF provides users with the option of customizing the minimum length to meet a variety of research objectives.

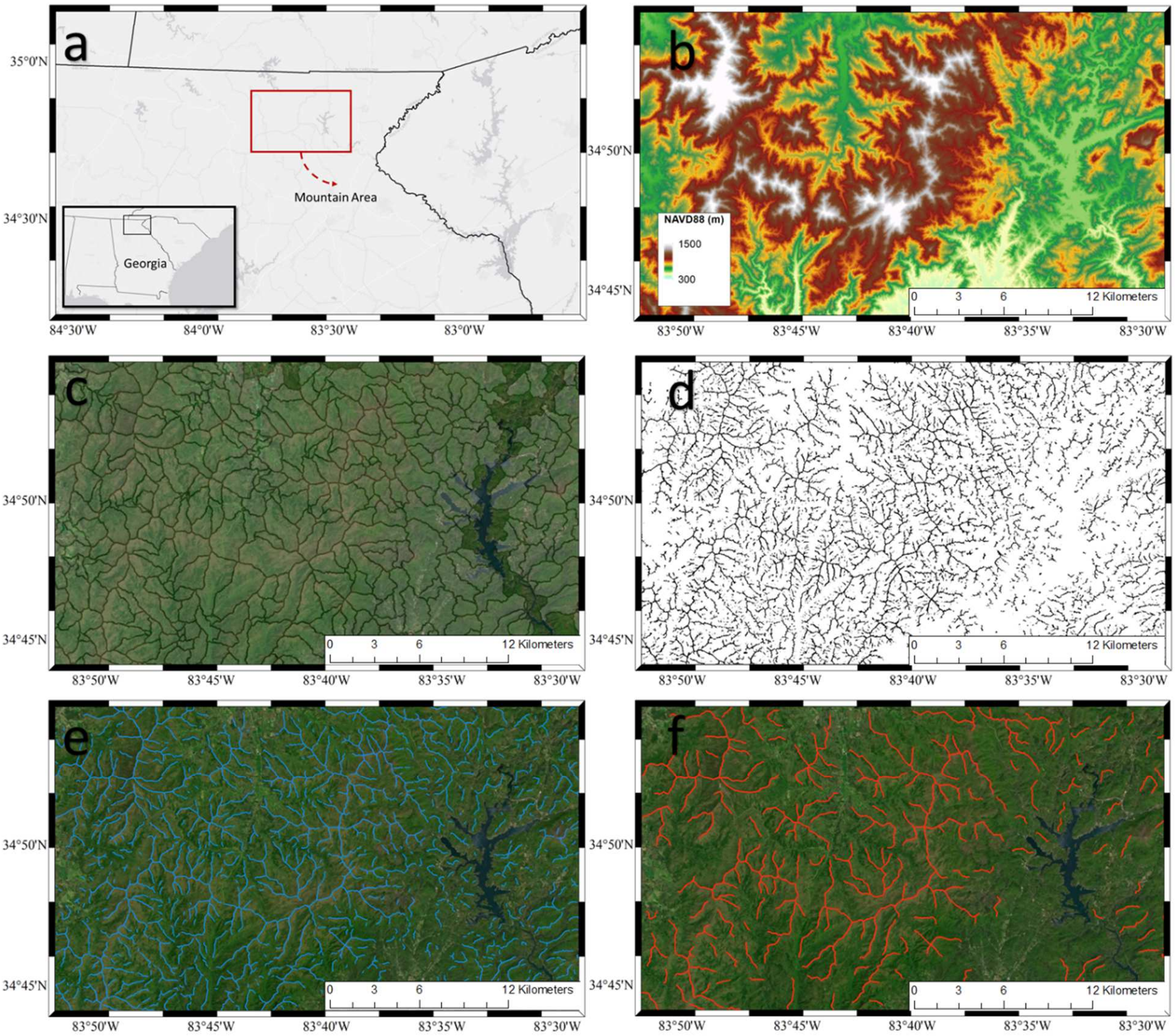

Figure 12. a) Location map of mountain area in Georgia. b) The DEM in mountain area. c) Watershed boundaries with a 10000 accumulated flow threshold. d) The potential VF raster image with the value of $\min_{dh}$ is 50 m and the value of $\min_{r}$ is 25 m in mountain area. e) Vertical features in mountain area longer than 200 m. f) Vertical features in mountain area longer than 1000 m.

3.4 Beach Area

The fourth study area is a coastal area located in Virginia Beach (Figure 13 (a)). According to the 10 m DEM supported by the NOAA (Figure 13 (b)), there is a natural barrier (i.e., sandy beach dune). Beach dunes are important for ecosystems habitats and coastal protection by reducing the impact of extreme coastal hazards such as wave and storm surge (Ranwell and Rosalind, 1986; Roelvink et al., 2009; Van der Meulen and Salman, 1996).

In this area, the 1.5 IQR of the dh is about 1.5 m (Figure 14). Since this area is small, the accumulated flow threshold used for extracting watershed boundaries is 100 (Figure 13 (d)) and the minimum dh and r are set 1.5 m and 25 m (i.e., 2.5 cell size) (Figure 12 (c)). The VFs with a minimum length of 50 m are shown in Figure 12 (e). The longest VF is the beach dune. PyVF is shown as able to extract raised features from the beach area.

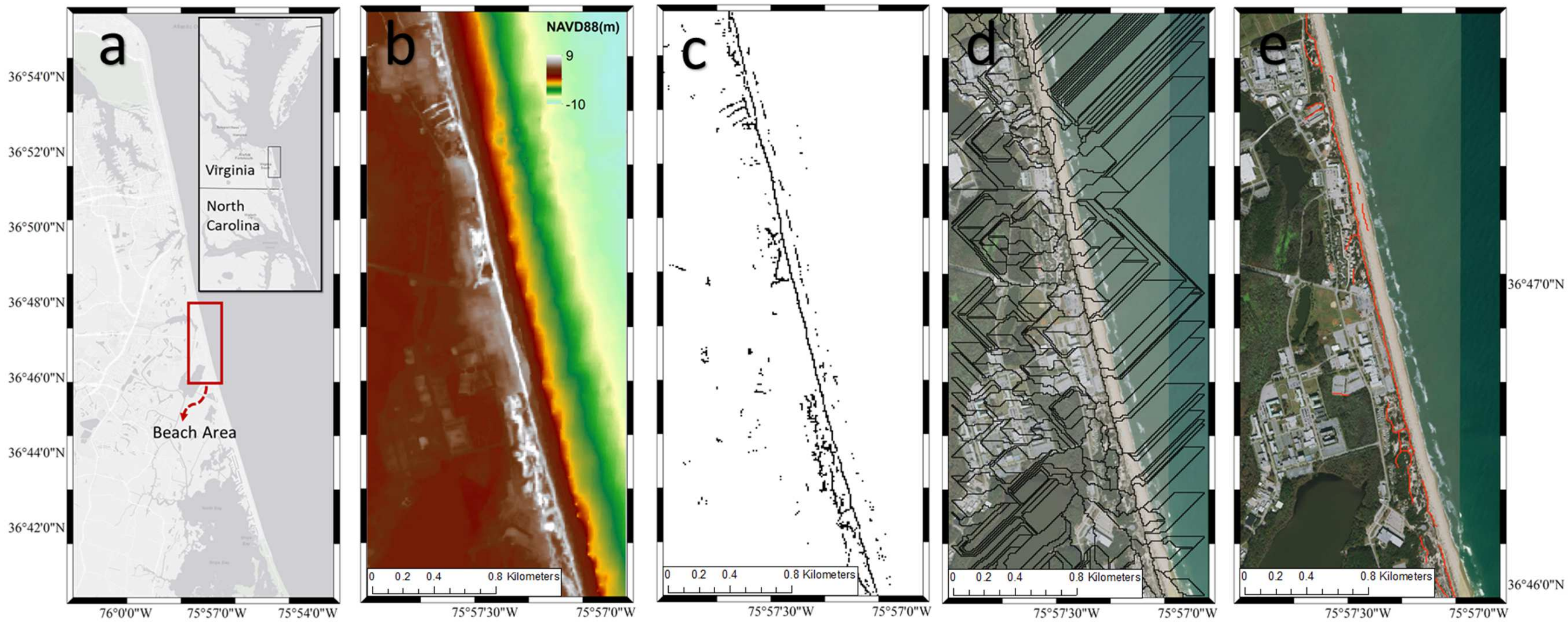

Figure 13. a) Location map of beach area in Virginia. b) The DEM in Beach Area. c) The potential VF raster image with the value of $\min_{dh}$ is 1.5 m and the value of $\min_{dh}$ is 25 m in Beach Area. d) Watershed Boundaries with a 100 accumulated flow threshold. (e) Vertical Features greater than 50 m.

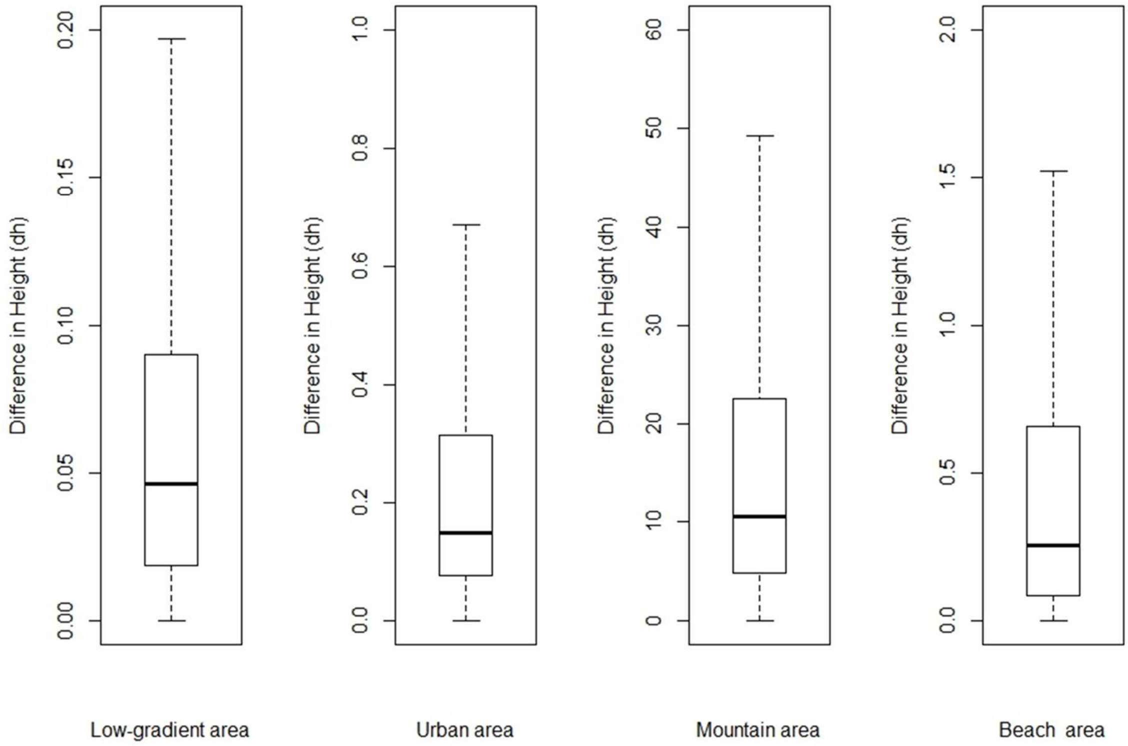

Figure 14. Boxplot of the elevation differences (dh) in low-gradient area, urban area, mountain area and beach area. Note the range of the y-axis varies among the plots.

4. Discussion

Since the extracted vertical features are required to be high enough polylines, the VF extraction method in this research combines two methods to achieve the requirement. They are the target recognition method and the target delineation method. The target recognition methods use the threshold of parameters dh (i.e., $\min_{dh}$) to ensure the detected cells with high elevation difference (Tribe, 1992). After that, with the target delineation method based on the accumulated flow threshold (T) (i.e., target delineation) (Ai, 2007), the potential vertical feature cells can be converted to the potential vertical feature polylines. The values of these thresholds will affect the number of extracted VFs. In this section, the focus is to discuss the considerations when selecting thresholds for these parameters. Finally, the advantages compared to the previous method and the current limitation are elaborated.

The target recognition in this research applies an iterative increasing size moving window to detect each cell within the DEM range. This method is superior to previous methods of fixed size moving windows because the importance of each VF can be automatically assessed by the variable dh and r. Compared with the fixed-size approach, the output VF height (dh) identified by the moving window with the increasing radius (r) is closer to the real height, though it may need additional computing time. Also, the radius (r) could be regarded as the width of a VF. dh and r can be considered as vertical and horizontal increments to calculate the slope of VFs. The four case studies presented highlight that the minimum values of parameter dh and r can be determined based on the IQR and the resolution of DEM. They also prove that the $\min_{dh}$ has a high relationship with the type of terrain for a given study area. For instance, in the area with large elevation variations, such as the mountain region, a larger $\min_{dh}$ is used to filter out less significant VFs. On the contrary, in the low-gradient area, the smaller $\min_{dh}$ is more appropriate due to the minor land elevation variations and small surface slope.

Since the watershed boundary delineation method effectively provides the location of ridges, it naturally serves to use the watershed boundary as VFs. The selection of the threshold (T) also has an impact on the VF delineation. If a large threshold is selected, less output watershed boundaries may cause the loss of significant VFs. If a small threshold is selected, there will be more watershed boundaries, and of course, the intersection of the watershed boundaries will also be significantly increased. Using a small threshold increases computation time, and many weak VFs are extracted and the thinned VFs have numerous spurs. This requires additional post-processing. Hence, the threshold selection should be an iterative process, from large to small, to determine an appropriate threshold.

The work of Coggin (2008) and Bilskie et al. (2015) start from the watershed delineation and considers the watershed boundaries that meet three criteria by special parameters as significant VFs. The method greatly reduces the number of watershed boundaries and can warrant the importance of the extracted VFs (i.e., portions of the watershed boundaries). However, there is no weighting among the extracted VFs. In unstructured mesh design, VFs with close spacing may face trade-offs. The parameter dh can provide the vertically significant order of each VF. Additionally, the watershed boundaries are along the edges of the grid cell instead of the centroid. When the VFs are narrow, the nodes of triangular elements could be positioned on the surrounding lower terrain by accident. The VFs extracted by PyVF are located along the center of raster cells to avoid careless element node placement.

There are, however, some limitations in PyVF. First, VFs rely on the position of watershed boundaries. When potential VF cells and watershed boundaries cannot coincide, there may be gaps in the VFs that should be continuous. This condition requires post-processing to compensate. In addition, the accuracy of VFs extracted with PyVF will be affected by the quality of the DEM. PyVF is more effective if a quality DEM is available. Third, since PyVF applies some ArcGIS functions in the target delineation method, the users must have access to an ArcGIS license. The last limitation that needs to be overcome is to speed up PyVF. The computer system environments used to run PyVF for the four study is listed in Table 2. The running duration for the four areas were list in Table 3. In addition to computer performance, the running duration depends on the number of potential vertical features. In other words, an area of the same size with more VF will take longer.

Table 2. Computer System Environment Parameters

Number	Item	Parameter
1	Operating System	Window 10
2	Memory	16.0 GB
3	CPU	Intel(R) Xeon(R) E5-1620 v3 @ 3.50GHz
4	ArcGIS version	10.8

Table 3. Running Duration for the four study areas.

Study area	Running Duration (h)	Area (Km^2)	DEM Resolution (m)
Low-gradient area	4	18×9	3
Urban area	1	10×8.5	3
Mountain area	8	40×20	10
Beach area	0.4	4×1.5	10

5. Summary and Conclusion

In this paper, the problem of extracting VFs from DEM is presented. PyVF is written in Python to solve this problem using the target recognition and target delineation algorithm. The target recognition aims to extract the potential VF cells applying a circular moving window with an iterative increasing size rather than a fixed size. The objective of the target delineation method is to convert the potential VF cells to VF polylines. The two main algorithms are mainly based on window size r, height parameter dh, and accumulated flow threshold T. Also, post-processing could be required for cleaning up VFs that are insignificant to the research objective.

PyVF is employed to extract VFs in four different landform areas: low-gradient coastal area, urban area, mountain area, and beach area. The VFs such as roadbed, levees, mountain ridges, and beach dunes in these areas are delineated. According to different landforms and research objectives, the appropriate values of parameters are changeable. The results of the four study areas demonstrate automatic VF delineation from disparate DEMs. Our future work will combine the PyVF tool with a local mesh scaling algorithm to extend the delineation of VFs beyond the geometric-based approach to include flow properties.

6. Acknowledgments

The authors would like to thank the feedback provided by Drs. Peter Bacopoulos, Jin Ikeda, and Félix L. Santiago-Collazo during the preparation of this article. This research was funded in part by The Water Institute of the Gulf under project award number CPRA-2015-COE-MB. This research was paid for in part with federal funding from the Department of the Treasury through the Louisiana Coastal Protection and Restoration Authority's Center of Excellence Research Grants Program under the Resources and Ecosystems Sustainability, Tourist Opportunities, and Revived Economies of the Gulf Coast States Act of 2012 (RESTORE Act) and the Louisiana Sea Grant Laborde Chair. The statements, findings, conclusions, and recommendations are those of the author(s) and do not necessarily reflect the views of the Department of the Treasury, CPRA, The Water Institute of the Gulf, or the Louisiana Sea Grant College Program.

Reference

Abdullah, Q.A., 2017. Mapping Matters—Breaklines for Lidar Data, Do We Really Need Them. Photogrammetric engineering & remote SenSing 83(9) 599-602.

Ai, T., 2007. The drainage network extraction from contour lines for contour line generalization. ISPRS Journal of Photogrammetry and Remote Sensing 62(2) 93-103.

Alzahrani, A.S., 2017. Application of Two-Dimensional Hydraulic Modeling in Riverine Systems Using HEC-RAS. University of Dayton.

Bates, P.D., Marks, K.J., Horritt, M.S., 2003. Optimal use of high-resolution topographic data in flood inundation models. Hydrological Processes 17(3) 537-557.

Bern, M., Plassmann, P., 2000. Chapter 6 - Mesh Generation, In: Sack, J.R., Urrutia, J. (Eds.), Handbook of Computational Geometry. North-Holland: Amsterdam, pp. 291-332.

Bevacqua, E., Maraun, D., Vousdoukas, M.I., Voukouvalas, E., Vrac, M., Mentaschi, L., Widmann, M., 2019. Higher probability of compound flooding from precipitation and storm surge in Europe under anthropogenic climate change. Science Advances 5(9) eaaw5531.

Bhaskaran, P.K., Gayathri, R., Murty, P.L.N., Bonthu, S., Sen, D., 2014. A numerical study of coastal inundation and its validation for Thane cyclone in the Bay of Bengal. Coastal Engineering 83 108-118.

Bilskie, M.V., 2012. Influence of Topographic Elevation Error on Modeled Storm Surge.

Bilskie, M.V., Coggin, D., Hagen, S.C., Medeiros, S.C., 2015. Terrain-driven unstructured mesh development through semi-automatic vertical feature extraction. Advances in Water Resources 86 102-118.

Bilskie, M.V., Hagen, S.C., 2018. Defining Flood Zone Transitions in Low-Gradient Coastal Regions. Geophysical Research Letters 45(6) 2761-2770.

Bilskie, M.V., Hagen, S.C., Medeiros, S.C., 2020. Unstructured finite element mesh decimation for real-time Hurricane storm surge forecasting. Coastal Engineering 156 103622.

Bilskie, M.V., Hagen, S.C., Medeiros, S.C., Cox, A.T., Salisbury, M., Coggin, D., 2016. Data and numerical analysis of astronomic tides, wind-waves, and hurricane storm surge along the northern Gulf of Mexico. Journal of Geophysical Research: Oceans 121(5) 3625-3658.

Bilskie, M.V., Hagen, S.C., Medeiros, S.C., Passeri, D.L., 2014. Dynamics of sea level rise and coastal flooding on a changing landscape. Geophysical Research Letters 41(3) 927-934.

Bodoque, J., Guardiola-Albert, C., Aroca-Jiménez, E., Eguibar, M., Martínez-Chenoll, M., 2016. Flood Damage Analysis: First Floor Elevation Uncertainty Resulting from LiDAR-Derived Digital Surface Models. Remote Sensing 8(7) 604.

Briese, C., 2004. Three-Dumensional Modelling of Breaklines from Airborne Laser Scanner Data. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences.

Brugelmann, R., 2000. Automatic Breakline Detection from Airbrone Laser Range Data. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 33(B3) 109-116.

Bunya, S., Dietrich, J.C., Westerink, J.J., Ebersole, B.A., Smith, J.M., Atkinson, J.H., Jensen, R., Resio, D.T., Luettich, R.A., Dawson, C., Cardone, V.J., Cox, A.T., Powell, M.D., Westerink, H.J., Roberts, H.J., 2010. A High-Resolution Coupled Riverine Flow, Tide, Wind, Wind Wave, and Storm Surge Model for Southern Louisiana and Mississippi. Part I: Model Development and Validation. Monthly Weather Review 138(2) 345-377.

Casas, A., Riaño, D., Greenberg, J., Ustin, S., 2012. Assessing levee stability with geometric parameters derived from airborne LiDAR. Remote Sensing of Environment 117 281-288.

Chang, Y.-C., Sinha, G., 2007. A visual basic program for ridge axis picking on DEM data using the profile-recognition and polygon-breaking algorithm. Computers & Geosciences 33(2) 229-237.

Chang, Y.-C., Song, G.-S., Hsu, S.-K., 1998. Automatic extraction of ridge and valley axes using the profile recognition and polygon-breaking algorithm. Computers & Geosciences 24(1) 83-93.

Chen, C., Liu, H., Beardsley, R.C., 2003. An Unstructured Grid, Finite-Volume, Three-Dimensional, Primitive Equations Ocean Model: Application to Coastal Ocean and Estuaries. Journal of Atmospheric and Oceanic Technology 20(1) 159-186.

Coggin, D., 2008. Coggin_LiDAR in coastal storm surge modeling Modeling linear raised features, Department of Civil, Environmental, and Construction Engineering. University of Central Florida: Orlando, Florida.

Colby, J., D., Dobson, J., G., 2010. Flood Modeling in the Coastal Plains and Mountains: Analysis of Terrain Resolution. Natural Hazards Review 11(1) 19-28.

Davies, E.R., Plummer, A., 1981. Thinning algorithms: a critique and a new methodology. Pattern Recognition 14(1-6) 53-63.

Dietrich, J.C., Westerink, J.J., Kennedy, A.B., Smith, J.M., Jensen, R.E., Zijlema, M., Holthuijsen, L.H., Dawson, C., Luettich, R.A., Jr., Powell, M.D., Cardone, V.J., Cox, A.T., Stone, G.W., Pourtaheri, H., Hope, M.E., Tanaka, S., Westerink, L.G., Westerink, H.J., Cobell, Z., 2011. Hurricane Gustav (2008) Waves and Storm Surge: Hindcast, Synoptic Analysis, and Validation in Southern Louisiana. Monthly Weather Review 139(8) 2488-2522.

Dube, S.K., Murty, T.S., Feyen, J.C., Cabrera, R., Harper, B.A., Bales, J.D., Amer, S.A., 2010. Storm surge modeling and applications in coastal areas. World Scientific Series on Asia-Pacific Weather and Climate 4 363-406.

Edwards, P.J., Williard, K.W.J., Schoonover, J.E., 2015. Fundamentals of Watershed Hydrology. Journal of Contemporary Water Research & Education 154(1) 3-20.

Esri, 2016. What is ArcPy? Help documents collection from ArcGIS Online [online].

Gallien, T., Kalligeris, N., Delisle, M.-P., Tang, B.-X., Lucey, J., Winters, M., 2018. Coastal Flood Modeling Challenges in Defended Urban Backshores. Geosciences 8(12) 450.

Gallien, T.W., Sanders, B.F., Flick, R.E., 2014. Urban coastal flood prediction: Integrating wave overtopping, flood defenses and drainage. Coastal Engineering 91 18-28.

Gayathri, R., Murty, P.L.N., Bhaskaran, P.K., Srinivasa Kumar, T., 2016. A numerical study of hypothetical storm surge and coastal inundation for AILA cyclone in the Bay of Bengal. Environmental Fluid Mechanics 16(2) 429-452.

Griffiths, T.R., 2010. An Enhanced Data Model and Tools for Analysis and Visualization of Levee Simulations.

Hagen, S.C., Westerink, J.J., Kolar, R.L., Horstmann, O., 2001. Two-dimensional, unstructured mesh generation for tidal models. International Journal for Numerical Methods in Fluids 35 669-686.

Ham, D.A., Pietrzak, J., Stelling, G.S., 2005. A scalable unstructured grid 3-dimensional finite volume model for the shallow water equations. Ocean Modelling 10(1) 153-169.

Hiller, J.K., Smith, M., 2008. Residual relief separation: digital elevation model enhancement for geomorphological mapping. Earth Surface Processes and Landforms 33(14) 2266-2276.

Horritt, M.S., Bates, P.D., 2002. Evaluation of 1D and 2D numerical models for predicting river flood inundation. Journal of Hydrology 268(1) 87-99.

Ikeuchi, H., Hirabayashi, Y., Yamazaki, D., Muis, S., Ward, P.J., Winsemius, H.C., Verlaan, M., Kanae, S., 2017. Compound simulation of fluvial floods and storm surges in a global coupled river-coast flood model: Model development and its application to 2007 Cyclone Sidr in Bangladesh. Journal of Advances in Modeling Earth Systems 9(4) 1847-1862.

IPCC, 2013. Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.

Kim, B., Sanders, B.F., Schubert, J.E., Famiglietti, J.S., 2014. Mesh type tradeoffs in 2D hydrodynamic modeling of flooding with a Godunov-based flow solver. Advances in Water Resources 68 42-61.

Koike, K., Nagano, S., Ohmi, M., 1995. Lineament analysis of satellite images using a Segment Tracing Algorithm (STA). Computers & Geosciences 21(9) 1091-1104.

Kraemer, C., Panda, S.S., 2009. Automating arc hydro for watershed delineation, Proceedings of the 2009 Georgia water resources conference, held at the University of Georgia.

Lin, C.-W., Tseng, C.-M., Tseng, Y.-H., Fei, L.-Y., Hsieh, Y.-C., Tarolli, P., 2013. Recognition of large scale deep-seated landslides in forest areas of Taiwan using high resolution topography. Journal of Asian Earth Sciences 62 389-400.

Luettich, R.A., Jr. ; Westerink, J. J. ; Scheffner, Norman W., 1992. ADCIRC: An Advanced Three-Dimensional Circulation Model for Shelves, Coasts, and Estuaries. Report 1. Theory and Methodology of ADCIRC-2DDI and ADCIRC-3DL.

Lundh, F., 2001. Python standard library. " O'Reilly Media, Inc.".

Marsh, C.B., Spiteri, R.J., Pomeroy, J.W., Wheater, H.S., 2018. Multi-objective unstructured triangular mesh generation for use in hydrological and land surface models. Computers & Geosciences 119 49-67.

McGuigan, K., Webster, T., Collins, K., 2015. A Flood Risk Assessment of the LaHave River Watershed, Canada Using GIS Techniques and an Unstructured Grid Combined River-Coastal Hydrodynamic Model. Journal of Marine Science and Engineering 3(3) 1093-1116.

Moftakhari, H.R., Salvadori, G., AghaKouchak, A., Sanders, B.F., Matthew, R.A., 2017. Compounding effects of sea level rise and fluvial flooding. Proceedings of the National Academy of Sciences 201620325.

Naccache, N.J., Shinghal, R., 1984. SPTA: A proposed algorithm for thinning binary patterns. IEEE transactions on systems, man, and cybernetics(3) 409-418.

Namin, M., Lin, B., Falconer, R.A., 2004. Modelling estuarine and coastal flows using an unstructured triangular finite volume algorithm. Advances in Water Resources 27(12) 1179-1197.

Nicholls, R.J., Wong, P.P., Burkett, V., Codignotto, J., Hay, J., McLean, R., Ragoonaden, S., Woodroffe, C.D., Abuodha, P., Arblaster, J., 2007. Coastal systems and low-lying areas.

NOAA, 2007. Topographic and Bathynebtruc Data Considerations.

Pain, C.C., Piggott, M.D., Goddard, A.J.H., Fang, F., Gorman, G.J., Marshall, D.P., Eaton, M.D., Power, P.W., de Oliveira, C.R.E., 2005. Three-dimensional unstructured mesh ocean modelling. Ocean Modelling 10(1) 5-33.

Parajuli, P.B., Ouyang, Y., 2013. Watershed-scale hydrological modeling methods and applications, Current Perspectives in Contaminant Hydrology and Water Resources Sustainability. Hydrology, pp. 57-80.

Purvis, M.J., Bates, P.D., Hayes, C.M., 2008. A probabilistic methodology to estimate future coastal flood risk due to sea level rise. Coastal Engineering 55(12) 1062-1073.

Ranwell, D.S., Rosalind, B., 1986. Coastal dune management guide. Institute of terrestrial Ecology.

Roberts, H.J., 2004. Grid Generation Methods for High Resolution Finite Element Models Used for Hurricane Storm Surge Prediction.

Roelvink, D., Reniers, A., Van Dongeren, A., De Vries, J.V.T., McCall, R., Lescinski, J., 2009. Modelling storm impacts on beaches, dunes and barrier islands. Coastal Engineering 56(11-12) 1133-1152.

Santiago-Collazo, F.L., Bilskie, M.V., Hagen, S.C., 2019. A comprehensive review of compound inundation models in low-gradient coastal watersheds. Environmental Modelling & Software 119 166-181.

Shen, J., Zhang, K., Xiao, C., Gong, W., 2006. Improved Prediction of Storm Surge Inundation with a High-Resolution Unstructured Grid Model. Journal of Coastal Research 226 1309-1319.

Sofia, G., Fontana, G.D., Tarolli, P., 2014. High-resolution topography and anthropogenic feature extraction: testing geomorphometric parameters in floodplains. Hydrological Processes 28(4) 2046-2061.

Tarboton, D.G., 2005. Terrain analysis using digital elevation models (TauDEM). Utah State University, Logan.

Tarolli, P., Sofia, G., Dalla Fontana, G., 2010. Geomorphic features extraction from high-resolution topography: landslide crowns and bank erosion. Natural Hazards 61(1) 65-83.

Tribe, A., 1992. Automated recognition of valley lines and drainage networks from grid digital elevation models: a review and a new method. Journal of Hydrology 139(1) 263-293.

Van de Sande, B., Lansen, J., Hoyng, C., 2012. Sensitivity of Coastal Flood Risk Assessments to Digital Elevation Models. Water 4(3) 568-579.

Van der Meulen, F., Salman, A., 1996. Management of Mediterranean coastal dunes. Ocean & Coastal Management 30(2-3) 177-195.

Walt, S.v.d., Colbert, S.C., Varoquaux, G., 2011. The NumPy Array: A Structure for Efficient Numerical Computation. Computing in Science & Engineering 13(2) 22-30.

Wang, X., Liu, T., Yang, D., Qu, Z., Clary Calvin, R., Wunneburger, C., 2011. Simulating Hydrologic Effects of Raised Roads within a Low-Relief Watershed. Journal of Hydrologic Engineering 16(7) 585-597.

Wang, Z., Ma, H., Zheng, C., Chen, R., 2018. High-Quality Contour Line Generation from LiDAR Point Clouds for the Area of Forests.

Westerink, J.J., Luettich, R.A., Feyen, J.C., Atkinson, J.H., Dawson, C., Roberts, H.J., Powell, M.D., Dunion, J.P., Kubatko, E.J., Pourtaheri, H., 2008. A Basin- to Channel-Scale Unstructured Grid Hurricane Storm Surge Model Applied to Southern Louisiana. Monthly Weather Review 136(3) 833-864.

Xie, D., Zou, Q.-P., Mignone, A., MacRae, J.D., 2019. Coastal flooding from wave overtopping and sea level rise adaptation in the northeastern USA. Coastal Engineering 150 39-58.

Yang, B., Huang, R., Dong, Z., Zang, Y., Li, J., 2016. Two-step adaptive extraction method for ground points and breaklines from lidar point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 119 373-389.

Yoon, T.H., Kang, S.-K., 2004. Finite Volume Model for Two-Dimensional Shallow Water Flows on Unstructured Grids. Journal of Hydraulic Engineering 130(7) 678-688.

Zhan, C., 1993. A Hybrid Line Thinning Approach. Proceedings Auto-Carto 11 396-405.

Zhang, H., Qu, D., Hou, Y., Gao, F., Huang, F., 2016. Synthetic Modeling Method for Large Scale Terrain Based on Hydrology. IEEE Access 4 6238-6249.

Zhang, T., Suen, C.Y., 1984. A fast parallel algorithm for thinning digital patterns. Communications of the ACM 27(3) 236-239.

Zheng, F., Westra, S., Leonard, M., Sisson, S.A., 2014. Modeling dependence between extreme rainfall and storm surge to estimate coastal flooding risk. Water Resources Research 50(3) 2050-2071.

Zheng, F., Westra, S., Sisson, S.A., 2013. Quantifying the dependence between extreme rainfall and storm surge in the coastal zone. Journal of Hydrology 505 172-187.